CROP SEGMENTATION AND YIELD COUNT USING GRAPH BASED APPROACH

[1]Vishwajeeth P

Department of Computer Science & Engineering

SJB Institute of Technology

[2]Subheeksha Kishor Sheka

Department of Computer Science & Engineering

SJB Institute of Technology

[3]Siddharth Mishra

Department of Computer Science & Engineering

SJB Institute of Technology

[4]Sumit Prakash

Department of Computer Science & Engineering

SJB Institute of Technology

Abstract ---- 'Arecanut', botanically known as Areca catechu, is tropical plant found all over South-East Asia. To obtain best quality produce and maximize the profit, arecanut must be harvested at specific stages. Hence segmentation of the crop bunches becomes very important. This paper focuses on graph-based segmentation along with superpixels. Generation of superpixels is performed by the help of SLIC (Simple Linear Iterative Clustering) Algorithm. The foreground and background of image is identified by markings done by human intervention. Number of Arecanuts in images is counted by using watershed algorithm.

***Keywords* —Computer vision; SLIC; Graph-cut; WaterShed algorithm; Object detection; image processing.**

I. INTRODUCTION

Traditionally, majority population of India takes agriculture to be their main occupation. It is considered as the backbone of India as the Indian economy depends on it. The majority population mentioned is about 58%, depending on agriculture for all their necessities such as food, shelter and for other living income. Among all the agricultural production of food, feed and fiber, Areca nut production in India is the largest in the world. The Areca palm is a tall-stemmed erect palm that grows to various heights depending on the climate. After 5 years of planting, the bearing begins [1]. It bears in huge clusters making it difficult for the farmer to count the areca nuts produced on the palm. Counting nuts is an important task for farmers for the purpose of yield estimation and to maintain the farm. An accurate nut segmentation and counting algorithm offers farming businesses the potential to optimize and streamline their harvest. More the farmers understand the variety of the yield in their entire farmland, better they can take decisions which can be knowledgeable. The decisions can also be taken keeping cost in regard to need of labour, storage of arecanuts, packages and also transportation of the nuts.

Image Segmentation is dividing an image into meaningful non overlapping regions, according to some objective criterion, homogeneity in some feature space or separability in some other. It eliminates unwanted background and produce the foreground image of the nuts making the counting of clustered nuts easier.

The paper has the following contents. Section 2 compiles the related works referred for the research on methods available for image segmentation and node count. Section 3 elaborates on graph cut segmentation with SLIC and watershed algorithm. It gives an insight on the design and architecture of the model used and explains the complete methodology. Section 4 contains result and analysis. Section 5 gives the conclusion.

II. RELATED WORK

A. Image segmentation:

Segmentation of images have been a very challenging task in the computer vision. There has been a lot of research on the segmentation considering various applications. One of them being yield harvesting and works have been for grapes [2], berry fruit [3], strawberry [4]. Other than the applications there are different ways or approaches for the segmentation like segmentation based on colour [5],[6]. Colour image segmentation is primarily based totally at the colour feature of image pixels assumes that homogeneous shades withinside the image corresponds to split clusters and for this reason significant objects in the image. In different words, every cluster defines a category of pixels that share comparable colour properties. As the segmentation outcomes rely upon the used colour space that can provide acceptable results for all kinds of images. In recent years, neural networks have shown quite interesting results for image segmentation [7],[8],[9].

B. Graph cut segmentation:

Graph cut is a semiautomated segmentation technique for separating foreground and background features in an image.[10][11] Good initialization is not required for graph cut segmentation. To distinguish what you want in the foreground and what you want in the background by creating scribbles on the image. These scribbles are irregular lines drawn on the image. The image Segmentation code automatically segments the image based on the given scribbles and shows the segmented image. Then the segmentation can be fine-tuned by adding more scribbles to the image until a precise outcome is obtained. To improve the speed of segmentation, graph cut segmentation implements graph theory on image processing. It creates an empty graph of the image where pixels of the image will be considered as nodes connected along with weighted edges.

C. SLIC:

Superpixels are an easy-to-use basic for computing local image characteristics. They are increasingly valuable for application such as depth estimation, image segmentation and bode model estimation, and object localization because they capture redundancy in the image and considerably minimise the complexity of future image processing tasks [12]. To be helpful, superpixels must be quick to utilise and provide high-quality segmentations. The algorithm simple linear iterative clustering (SLIC) is used for generation of the superpixels. It conducts local clustering for pixels in a 5-D space defined by the CIELAB colour space's L, a, and b values, as well as the x, y pixel coordinates. A revolutionary distance measure in super pixel shapes pushes compactness and regularity, and smoothly accommodates grayscale and colour images.

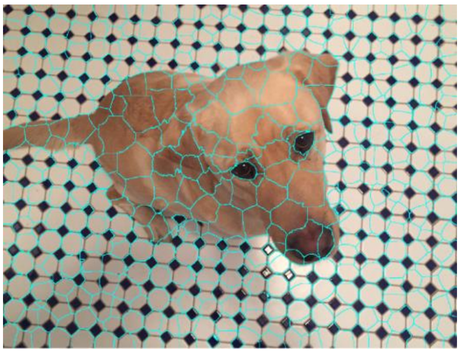
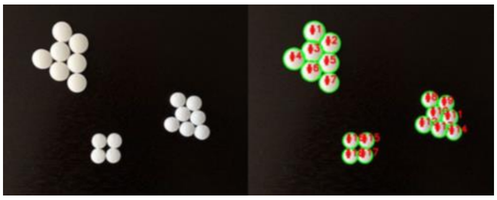


Figure 2.1. Superpixels using SLIC

D. Watershed algorithm:

Nodes in images can be calculated using various algorithms. Blob detection [13] is one such of algorithm. A blob is a group of connected pixels in an image that share some common property. Blob detection is used to locate and mark the dark linked patches, which are known as blobs. These blobs can be filtered by colour, size and shape. Convolution is the most used approach for blob detection. Blob detectors are divided into two categories: (i) differential approaches, which are based on the function's derivatives with respect to position, and (ii) methods based on local extreme, which are focused on identifying the function's local maxima and minima. And another important, the watershed algorithm is a segmentation algorithm for extracting touching or overlapping objects in images. We must start with user-defined markers when using the watershed method. These markers can be manually defined using point-and-click, or they can be defined automatically or heuristically using thresholding and/or morphological processes.

The watershed technique treats pixels in our input image as local elevation based on these markers – the methods 'floods' valleys from the markers outwards until the valleys of various markers meet. The markers must be set accurately in order to produce an accurate watershed segmentation.

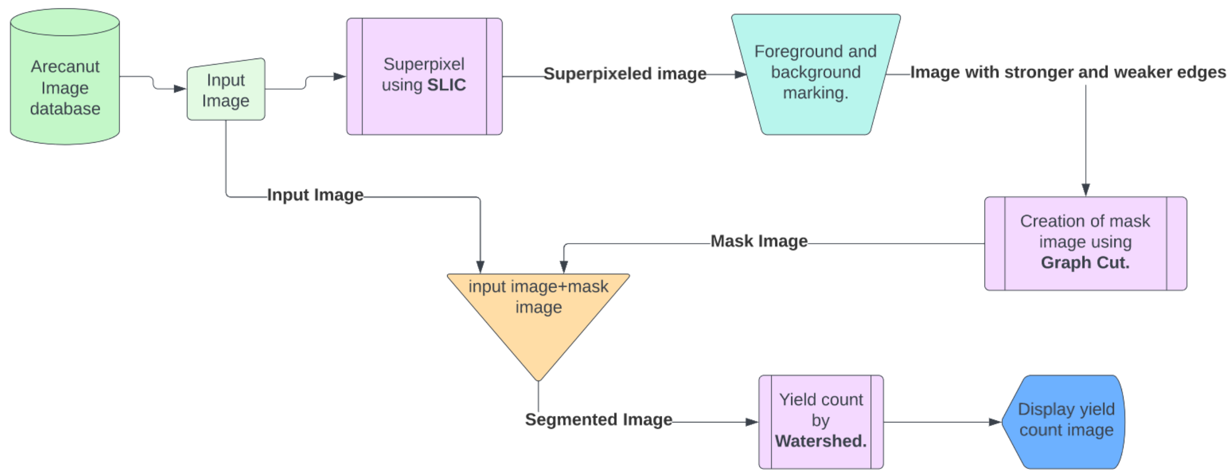


(a) (b)

Figure 2.2. Watershed algorithm. (a) Input image (b) Counting using Watershed algorithm

III. METHODOLOGY

1. System Architecture.



3.1.1 System Architecture

System architecture is a conceptual model that defines the structure and behaviour of the system. It comprises of the system components and the relationship describing how they work together to implement the overall system.

The image will be entered by the user into program. There is no need of image pre-processing before entering the image into the program. After the image is inserted by the user the algorithm starts working, after the image is the form of superpixel the user makes certain foreground and background marking, after the marking is done by the user the image is segmented by the graph segmentation code and we get a mask image, the mask and the real image is used to make the proper segmented image, and then that segmented image is used by the watershed algo to calculate the to yield count of arecanut in the image, after getting the yield count we calculate the performance of the segmentation. In the fig 3.1.1 you can see the detail architecture.

2. Methodology.

Graph cuts are used to divide an image into background and foreground segments. The graph-cut method is divided into two steps. The first involves integrating marker background regions with adjacent regions. Some non-marker background regions will be combined with the relevant background markers after this merging. The second step concentrates on non-marker regions that remain after the first. We would be unable to extract each individual areca nut from the segmented image using traditional image processing methods such as thresholding and contour detection, but by leveraging the watershed algorithm, we are able to detect and extract almost every areca nut, even if they are overlapped or touching. The watershed method produces a series of labels, each of which corresponds to a single object in the image. All we have to do now is loop through each of the labels one by one and extract each item.

A. Process the Image into Superpixels by SLIC.

SLIC is simple to use, and only one parameter can be controlled to adjust the number of images superpixels. This algorithm begins by sampling K cluster centres that are evenly distributed and transferring them to seed positions that correspond to the lowest gradient point in a 3x3 neighbourhood. Image gradients are computed as:

$$G(x,y)=\|I(x+1,y)-I(x-1,y)\|^2+\|I(x,y+1)-I(x,y-1)\|^2$$

$\|.\|$ is the L2 norm, and $I(x,y)$ is the lab vector corresponding to the pixel at location (x,y). This takes both colour and intensity information into account. Each pixel in the image is linked to the cluster centre closest to it, whose search area overlaps it. After all of the pixels have been assigned to the nearest cluster centre, a new centre is calculated using the average labxy vector of all of the cluster's pixels.
Until convergence, the procedure of associating pixels with the closest cluster center and recomputing the cluster centre is repeated. A few stray labels may exist at the end of this operation, that is, pixels in the neighbourhood of

a bigger segment with the same label but not related to it. The penultimate phase of the method can ensure connectivity by relabelling disjunct segments with the labels of the largest neighbouring cluster.

Fig 3.2.1: Normal Image

Fig 3.2.2: Superpixel image using SLIC

B. Build a Graph using superpixels as nodes.

Now each superpixel in the image will treated as a node and a graph will be formed by connecting those nodes. As the color model [10], an input image is represented as a color vector of the form:

$$Z = (z_1, \cdots, z_n, \cdots, z_N)^T$$

In this work, we modify this representation of an image from pixel-based to superpixel-based. And n z is the color vectors of the superpixel i. Segmentation of the image is represented as a binary vector of form:

$$\boldsymbol{a} = \{\boldsymbol{a_1}, \dots, \boldsymbol{a_n}, \dots, \boldsymbol{a_N}\}, \quad \boldsymbol{a_n} \in \{\mathbf{0}, \mathbf{1}\}$$

Gaussian Mixture Model (GMM) is used in the color model; the combination of mean and variance vectors of each Gaussian model is stored in a parameter . The energy function in the form of Gibbs energy [10] is formulated as

$$E(a, \theta, z, k) = U(a, \theta, z, k) + V(a, z)$$

In this energy function, describes a measure of the estimated color models to the segmentation mask a.

$$U(a, \theta, z, k) = \sum_n D(a_n, \theta, z_n, k_n)$$

$$D(a_n, \theta, z_n, k_n) = -\log p(z|a_n, k_n, \theta) - \log \pi(a_n, k_n)$$

Where and p(.) is a Gaussian probability distribution, and (·) are mixture weighting coefficients. Model parameters are computed by using Expectation Maximization (EM) algorithm iteratively [10].

$$V(a, z) = \gamma \sum_{(m,n) \in C} B_{(m,n)} [a_n \neq a_m]$$

is the boundary characteristics of superpixel an, the coefficient

$$B_{(m,n)} = dis(m,n)^{-1} \exp(-\beta \| z_m - z_n \|^2)$$

is the penalty coefficient of the superpixels m and n, if the superpixels m and n are similar, the coefficient will be relatively large, and vice versa; γ is the weight coefficient of them.

So, the graph on the superpixel image will look like this.

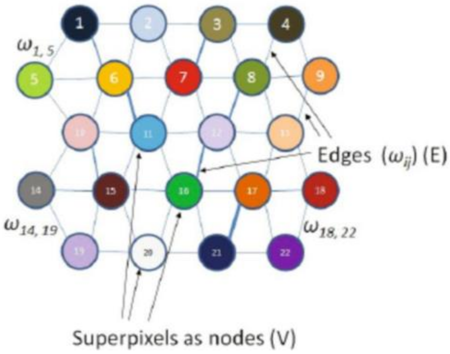


Figure 3.2.3. Graph representation of superpixels

C. Mark the seeds of regions manually.

We specify the seed points dynamically in the picture using dashes, indicating which parts are arecanut and which areas are non-apparel, based on the results of image pre-processing segmentation.

Fig 3.2.4: Markings are done on the image

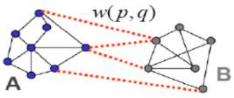
D. Mask image created using Graph Cut algorithm.

Graph Cuts is an image segmentation algorithm based on Gaussian Mixture Model (GMM). The energy function is optimized by min-cut(max-flow) algorithm. The main process of segmentation algorithm is as follows:

1. Superpixels with red seeds are initially placed in the foreground class, while superpixels with

blue seeds are initially placed in background class.

2. The colour GMMs, are initially created for the foreground and background classes.
3. Each superpixel in foreground class is assigned to the GMM component in the foreground GMM, and each region in the background class is assigned to GMM component in the background class, as mentioned in part b.
4. New GMM component in the background class, as mentioned in part b.
5. Using min cut(max flow) to estimate the segmentation and find the new foreground and background classification of superpixels.
6. Repeat 3) until convergence



$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

Criteria: By minimizing this cut value, one can optimally bi-partition the graph and achieve good segmentation.

Normalized cut [1,2] computes the cut cost as a fraction of the total edge connections to all the nodes in the graph.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$\text{where} \quad assoc(A,V) = \textstyle\sum_{u \in A, t \in V} w(u,t)$$

In the below figure (Fig 6.) we can see how graph cut separated the weaker edges from the strong edges. So, a line will be drawn by the algo over the weaker edges of the graph and using that link the graph will be cut and weaker neighbours will turned black and stronger neighbours will be turned white, hence after cutting we get a black and white masked image.

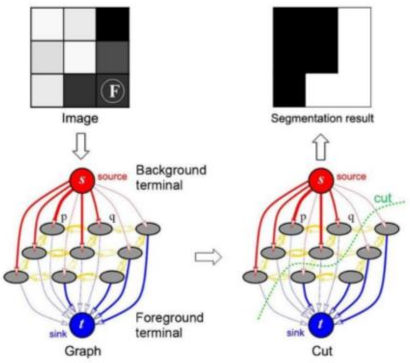


Figure 3.2.5. Segmentation process

So below is the arecanut masked image created after graph cut.

Fig 3.2.6. Masked Segmented image of arecanut.

E. Segmented Image (output image).

Output image is created by combining masked segmented image with the input image, so below is the output image Fig 8. This image will be used for yield count.

Fig 3.2.7. Segmented image of arecanut.

F. Getting the Yield Count of arecanut.

To get the yield count we use watershed algo to do it. The watershed algorithm is a classic algorithm used for segmentation and is especially useful when extracting touching or overlapping objects in images.

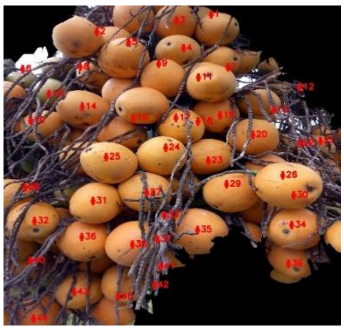


Fig 3.2.8: Yield count of Arecanut

IV. RESULTS AND ANALYSIS

It is tough to make a clear comparison between different image segmentation methods. So, we can't give a straight decision on the best model. So, we use performance metrics for analysing which model is the most efficient. For performance measure we use two types of performance metrices:

1. Intersection-Over-Union (IoU, Jaccard Index).
2. Dice similarity coefficient.

1.Intersection-Over-Union (IoU, Jaccard Index).

The Intersection-Over-Union (IoU), also known as the Jaccard Index, is one of the most used metrics in semantic segmentation and for good reason. The IoU is a very straightforward metric that's extremely effective. IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth, as shown on the image to the left. This metric ranges from 0–1 (0–perfectly overlapping segmentation.

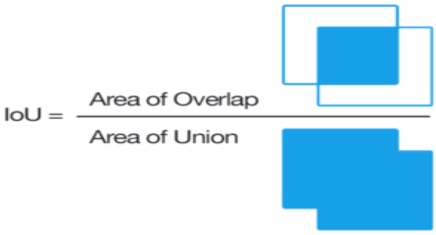


Fig 4.1 IoU

For binary (two classes) or multi-class segmentation, the mean IoU of the image is calculated by taking the IoU of each class and averaging them. (It's implemented slightly differently in code).

2.Dice similarity coefficient.

Simply put, the Dice Coefficient is 2 * the Area of Overlap divided by the total number of pixels in both images. (See explanation of area of union in Fig 4.2).

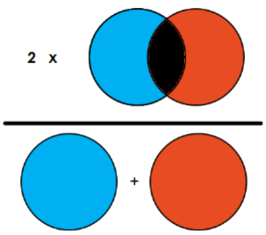


Fig 4.2. Illustration of Dice Coefficient. 2xOverlap/Total number of pixels

Experiments were carried out on 1000 ripe and unripe natural images of arecanut using the above process. The corresponding images are used and value of IoU and DSC of 5 ripe and unripe images is computed by comparing them with the ground truth as shown in Table 1 and Table 2. Their False Positive (FP) and error percentage was also carried out by using yield count of machine and actual count for comparison. Image ripe_3, ripe_4 and unripe_4 have little low score because they have lot of inflorescences which could hinder the accuracy of the model by few margins. Rest all images in the dataset show promising results with the model we have used. Yield Count of images are accurate in best case scenarios where there is less content of inflorescence. In other cases, there is an average loss percentage of 4.56% in yield count of arecanuts. Average IOU and DSC of 1000 ripe arecanut images are 0.713115 and 0.85166 and for 1000 unripe arecanut images are 0.698675 and 0.84655 as shown in Table 3.

Table 1. Performance metrics of Ripe Dataset

Image	Actual	Detected	FP	IOU	DSC	Error(%)
ripe_1	46	44	3	0.6632	0.8132	4.34782609
ripe_2	26	26	1	0.7629	0.9129	0
ripe_3	35	38	1	0.6214	0.7423	8.57142857
ripe_4	42	39	3	0.6288	0.7788	7.14285714
ripe_5	21	25	2	0.8232	0.9732	19.047619

Table 2. Performance metrics of Unripe Dataset

Image	Actual	Detected	FP	IOU	DSC	Error(%)
unripe_1	36	36	0	0.8199	0.9699	0
unripe_2	42	46	5	0.8447	0.9947	9.5238095
unripe_3	49	47	2	0.7045	0.8545	4.0816327
unripe_4	35	34	3	0.5977	0.7477	2.8571429
unripe_5	35	35	3	0.7287	0.8787	0

Table 3. Performance metrics of Dataset

Dataset of 1000 images	IOU	DSC
Ripe	0.713115	0.85166
Unripe	0.698675	0.84655

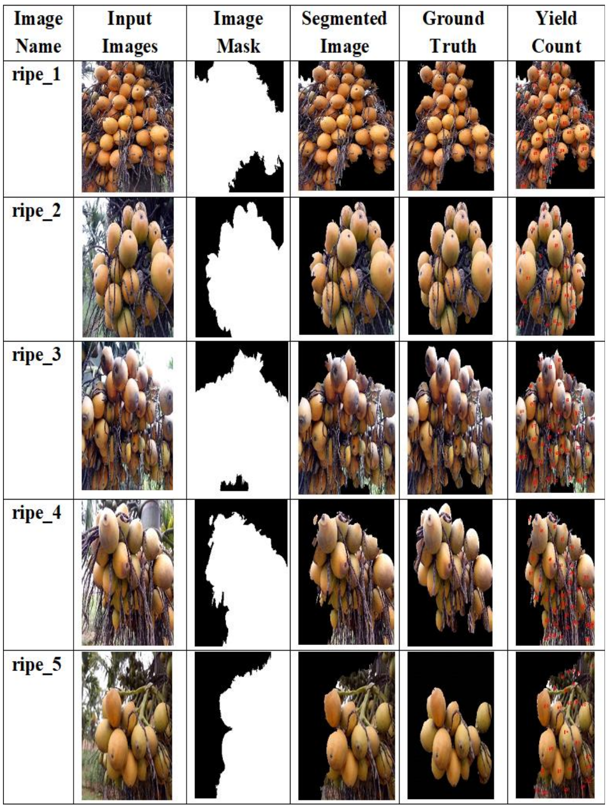


Fig 4.3. Output images of ripe dataset comprising of input image, image mask, segmented image, ground truth and yield count

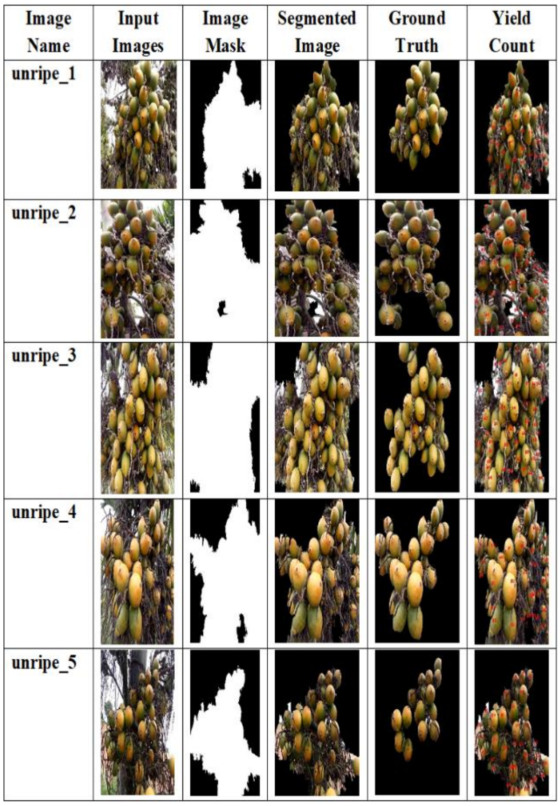


Fig 4.4. Output images of unripe dataset comprising of input image, image mask, segmented image, ground truth and yield count

V. CONCLUSION

This paper focuses on image segmentation and yield count of arecanuts. The segmentation method should be robust to handle these variations such as the effect of scale, illumination, and inflorescence in the database. Experimental results show that, the Graph-cut segmentation model segments Arecanut bunches from associate input image with better accuracy providing an average IoU score of 0.7131 and average DSC of 0.8516 for ripe image dataset and an average IoU score of 06986 and average DSC of 0.8465 for unripe image dataset. The algorithm used for the yield count gives an error of 4.56% approximately for both ripe and unripe dataset. These methods can be applied to get accurate yield count even in the rural areas as the algorithms used does not require high computational power like machine learning algorithms.
Performance of the proposed technique could be improved in the future by automating the marking process in the segmentation. And effective removal of inflorescence can improve the accuracy significantly.

VI. REFERENCES

[1] V Raghavan and H. K. Barauh, "Arecanut: India's most popular masticatory- History, Chemistry, and utilization",1958.

[2] R. Marani, A. Milella, A. Petitti, G. Reina "Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera",2020.

[3] Xueping Ni, Changying Li, Huanyu Jiang, Fumiomi Takeda., "Three-dimensional photogrammetry with deep learning instance segmentation to extract berry fruit harvestability traits", ISPRS Journal of Photogrammetry and Remote Sensing 171 297–309,2021.

[4] Zongmei Gao, Yuanyuan Shao, Guantao Xuan, Yongxian Wang, Yi Liu, Xiang Han, "Real-time hyperspectral imaging for the in-field estimation of strawberry ripeness with deep learning", Artificial Intelligence in Agriculture 4 31–38, 2021

[5] Siddesha S, S K Niranjan, V N Manjunath Aradhya, "A Study of Different Color Segmentation Techniques for Crop Bunch in Arecanut" Computer Science Review 38 (2020) 100301.

[6] Dhanesha R, Shrinivasa Naika C.L, Kantharaj Y, "Segmentation of Arecanut Bunches using YCgCr Color Model", 10.1109/ICAIT47043.2019.8987431,2019

[7] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Deep neural network concepts for background subtraction:A systematic review and comparative evaluation,” Neural Netw., vol. 117, pp. 8–66, Sep. 2019.

[8] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, “Background subtraction in real applications: Challenges, current models and future directions,” Comput. Sci. Rev., vol. 35, Feb. 2020, Art. no. 100204.

[9] M. Mandal and S. K. Vipparthi, “An empirical review of deep learning frameworks for change detection: Model design, experimental frameworks, challenges and research needs,” IEEE Trans. Intell. Transp. Syst., early access, May 19, 2021, doi: 10.1109/TITS.2021.3077883.

[10] Pedro F. Felzenszwalb, Daniel P. Huttenlocher, “Efficient Graph Based Image Segmentation”, 2017.

[11] Jianbo Shi, Jitendra Malik “Normalized Cuts and Image Segmentation”, IEEE Transactions on pattern analysis and machine intelligence, vol 22, no. 8, aug 2000.

[12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Susstrunk “SLIC Superpixels Compared to State-of-the-art Superpixel Methods”, 2011.

.

[13] Vladimir Petrović and Jelena Popović-Božović “Towards Real-Time Blob Detection in Large Images with Reduced Memory Cost” , 3rd international conference on electrical and computing engineering, jun 2016.

[14] Md. Sharifur Rahman; Md. Rafiqul Islam “Counting objects in an image by marker-controlled watershed segmentation and thresholding”, 3rd IEEE international advanced computing conference, 2013.