Development of a Pineapple Fruit Recognition and Counting System using Digital Farm Image

Nancy C. Woods[1], Emmanuel O. Abuh[2] and A.B.C. Robert[3]
[1, 3] Department of Computer Science, University of Ibadan, Ibadan, Nigeria.
Email: [1]Chyn.woods@gmail.com, [3]abc.robert@live.com
&
[2]Department of Computer Science, Federal College of Forestry, Jos, Nigeria.
Email: [2]numiabuh@gmail.com

ABSTRACT
Fruit detection is important for farmers to estimate yield and manage their farm. Unfortunately, most farmers carry out fruit detection and especially counting manually. Using computer vision, an automated fruit detection and counting system would enable farmers improve and modernize their harvest process. The automated system could give the farmers ability to predict their yield as they plan the sale. In this research work, an automated system was developed to detect, recognize and count pineapple fruits in a digital still image of a farm. The proposed system includes image acquisition from the orchard using a camera. Image noise removal was carried out using median filters, then the regions of interest were segmented. Furthermore, SURF feature description and extraction were used to extract feature points. The feature points extracted were then classified using a support vector machine for training, testing and detecting fruits points in still images. A dataset consisting of 120 images were used in this study. Approximately 80% and 20% of the images were used for training and testing, respectively. MATLAB was used. The simulation result revealed that the method is reliable, feasible and efficient when compared to the manual detection method, with successful pineapple fruit detection and counting rates of 87.37 %.

Keywords: *Image processing, Fruit recognition, Pineapple fruit, fruit counting*

African Journal of Computing & ICT Reference Format:

Nancy C. Woods, Emmanuel O. Abuh and A. B. C. Robert (2019), Development of a Pineapple Fruit Recognition and Counting System using Digital Farm Image,
Afr. J. Comp. & ICT, Vol.12, No. 2, pp. 131 - 139.

I. INTRODUCTION

Computer vision is a technology for acquiring and analysing an image of a real scene by computers to control machines or to process it [1]. It includes capturing, processing and analysing images to facilitate the objective and non-destructive assessment of visual quality characteristics in agricultural and food products [2]. The application of Computer Vision Systems in agriculture has increased tremendously in recent years, since it provides substantial information about the nature and attributes of the produce, reduces costs, guarantees the maintenance of quality standards and provides useful information in real time. This is usually achieved via image classification, which is used to identify different objects in an image automatically and to extract the required features from an image [3]. One of the major challenges in detecting unripe pineapple fruit using colour techniques lies in the similarity of the colour of the fruit and the leaves. Natural outdoor conditions, non-uniform illumination is another challenging problem. Partial occlusion of the pineapple fruit by its leaves and other fruits can cause further challenges in detection and counting of pineapple fruits using computer vision [4].

Fruit counting and mapping in orchards is important for growers as it facilitates efficient utilization of resources and improves returns per unit area and time. An accurate automated fruit detection and counting algorithm gives agricultural enterprises the ability to optimize and streamline their harvest process, manage processes such as chemigation, fertigation and thinning [5] [6].

The problem of identifying fruits on trees or plants with similar background colour has been of great interest in agricultural crop estimation work [7]. Obviously, it is clear that a complete estimation of fruits on trees or plants is quite tedious and time-consuming. Therefore, in order to minimize the time spent with manual counting methods as well as reduce the cost of personnel, an automated system is needed to obtain estimates of the total number of fruits in a farm image. Thus, the major goal of this research is to investigate the applicability of computer vision techniques in automated recognition and counting of total number of pineapple fruits in an orchard using texture analysis.

II. RELATED WORK

Numerous studies have proposed the use of texture analysis in fruit detection methods. [8], [9], [10], [11] used Gabor texture analysis proposed by [12]. [10] worked on segmenting green citrus fruits from their background. Their result from texture analysis was integrated with blob analysis and they identified 75.3% of the fruits. Their algorithm had a 27.3% false detection of leaves and stems as green citrus. The authors recognized that variable lighting condition, visual complexity of the background and varying fruit size limited the accuracy of the system. Therefore there is need to improve on feature detection and extraction. Feature detection and image matching represent two vital tasks in computer vision applications [13]. A feature is usually defined as the intersection of two edges, or a point for which there are two dominant and different edge directions in a local neighbourhood of pixels. [14] Carried out experiments on different computer vision systems to identify fruits for automated harvesting using a laser range-finder.

[15] Worked on methods to recognize apples grown on trees, which used the texture and redness colour. It was shown that redness works equally well for green apples as for red ones. [16] Proposed a method to recognise mature tomato fruits and locate fruit cluster positions for tomato harvest applications. [17] Developed a picking robot to recognise and cut sweet peppers in greenhouses, but their image analysis methods were developed only for this specific application under fixed lighting conditions. [18] Implemented an automatic detection and counting system for citrus fruit yield estimation. They noted that green citrus identification in the green background was a very difficult task with the problem of occlusion. In their approach, shape and

texture classification combined with support vector machine (SVM) was used to detect as many citrus fruits as possible. The problem of wrong detection was solved by graph-based connected component algorithm and Hough transform for line recognition. In their work, partially occluded citrus fruit were impossible to detect due to the presence of shadow which caused poor discrimination between the background and citrus fruit.

[6] Developed a method for green fruit counting in digital images of orange trees with a combination of the techniques of colour model conversion, thresholding, histogram equalization, spatial filtering with Laplace and Sobel operators and Gaussian blur. A part of the work focused on counting the orange fruits in a single image with a detection rate of false-positives of 3% in images acquired in good conditions.

[5] Implemented an Image processing framework for detecting fruit and estimating yield in an orchard. Feature learning algorithms were utilized for image segmentation, followed by detection and counting of individual fruits. The image segmentation was performed using ms-MLP and a CNN. Watershed segmentation (WS) and circular Hough transform (CHT) algorithms were used to detect and count individual fruits.

III. METHODOLOGY

3.1. Image dataset

The major image dataset was obtained from National Horticultural Research Institute (NIHORT) pineapple farm. A total of 206 images were taken during the second week of December 2017 and the last week of February 2018, using monocular video cameras. At the end of the image capturing exercise, a dataset containing 120 carefully selected images was created and used for the research work. The images were taken in outdoor lighting condition. Brightness and aperture speed was adjusted for each farm area before taking pictures of the farm area. Higher aperture speeds were required during bright daylight condition and lower shutter speeds were useful during the late afternoon to obtain good images with approximately unvarying brightness. During image acquisition, brightness, contrast, constant aperture speed and aperture of the camera was used most of the time during image capture. For perfect images, the image has to be captured in a user controlled environment. The controlled environment refers to a situation where the end user controls the picture background, the distance between a camera and an object, and a light source. Some images were also obtained from the internet for testing purposes.

3.2: Image pre-processing

Figure 1 summarizes the architecture of the pre-processing steps performed on the images by the system. The first step in this research work was to reduce the noise in images that were acquired during the image acquisition process, because digital images are prone to various types of noise. Median filtering was used to remove the noise in the image because it is able to remove some outliers without reducing the sharpness of the image. Median filtering is a nonlinear method used to remove noise from images. It works by moving through the image pixel by pixel, replacing each value with the median value of neighbouring pixels. The pepper and salt noise (Figure 2) was also removed by the median filter technique and then the images were converted to grayscale (Figure 3).

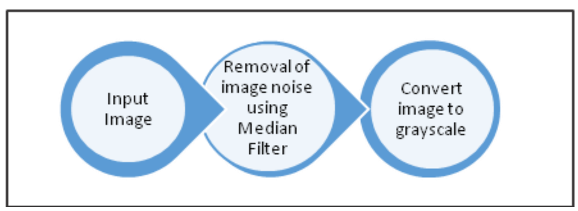

Figure 1: Image pre-processing steps

Figure 2: image with pepper and salt noise

Vol. 12, No. 2, June 2019, pp. 131 - 139 *ISSN 2006-1781*
Nancy C. Woods, Emmanuel O. Abuh and A. B. C. Robert (2019), Development of a Pineapple Fruit Recognition and Counting System using Digital Farm Image

Figure 3: grayscale output image

3.3 Fruit Detection

The pineapple fruit detection process as summarized in figure 4 consists of Feature Location Identification, Feature Descriptor Computation, Descriptor Classification, and Morphological processing. These steps made use of the pre-processed grayscale images.

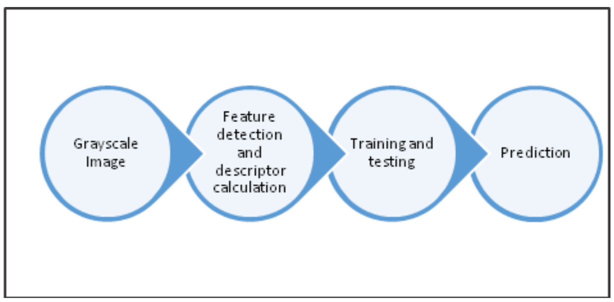

Figure 4: Fruit Classification and Detection Architecture

Feature detection selects regions of an image that have unique content, such as corners or blobs and are used to find points of interest that can be used for further processing. The key to feature detection is to find features that remain locally invariant so that you can detect them even in the presence of rotation or scale change. Feature Descriptors rely on image processing to transform a local pixel neighbourhood into a compact vector representation. This new representation permits comparison between neighbourhoods regardless of changes in scale or orientation. The descriptor calculation for these methods takes a region around the feature point, divides it into sub-regions, then returns a vector of numbers to represent gradient magnitudes and orientations over the region. The SURF algorithm in [19] was adopted for feature extraction because it approximates gradients using block differences.

After computing feature points and descriptors, each descriptor was classified using a support vector machine (SVM) classifier. The feature points and descriptors from a set of training and test images are extracted, then each feature point hand-labelled as positive (the key point is on the fruit's surface) or negative (the key point is on the plant or the background). The image region used to compute the feature descriptor may overlap the boundary between a fruit and the background.

A series of SVMs were then trained to perform a grid search using k-fold cross-validation in the SVM's hyper-parameter space, to find the best set of hyper parameters in terms of accuracy on the cross-validation test set. The k-fold cross-validation method was adopted for classifier validation. First, the data are divided into k partitions (groups), one fold is taken as a validation set, and then the classifier is trained on the remaining k-1 folds and tested on the validation fold. The procedure was executed k times, each time with a different validation set, and the result was the average validation set classification accuracy over k rounds. At runtime, the resulting pre-trained SVM model was simply loaded into memory and used to classify each feature point in the input image as positive or negative.

3.4 Candidate Fruit Region Extraction

For each input image, after feature location identification, feature descriptor computation and feature classification using the pre-trained SVM, the points classified as positives (on a fruit surface) are used to construct a binary image M in which each pixel corresponding to a positive is set to 1 and all other pixels are set to 0. Then morphological operations are used to connect regions with dense positive detections and discard regions too small to be considered candidate fruit regions. The output is shown in Figure 5.

Figure 5 : Fruit Region Extraction

3.5 Mature Pineapple Fruit Detection and Precise Segmentation

To detect mature fruit, a technique that can be compared directly to that used in [4] was used. As earlier written, the dataset used for this research work comprised of a set of 120 selected images. The dataset was partitioned such that 100 images were used for training, while the test data comprised of 20 images. 50 random images were also selected from the training dataset and used along with the same 20 test images. These images were also manually labelled, by selecting only the fruit skin without the crown as a fruit.

Our research was conducted in two stages. In the first stage, image data on mature pineapple plants was processed with the aim of segmenting fruit from the background. In the second stage, the SVM classifier was used, to identify and classify the fruit region as false or true positive. Beyond pineapple fruit recognition, the fruits were also counted.

3.6. Automated Fruit Counting Algorithm

Connected component labeling algorithm [20] was used to count the total number of pineapple fruits in each test image. It works by scanning the image, pixel-by-pixel in order to identify connected pixel regions, which are regions of adjacent pixels that share the same set of intensity values V. For a binary image V={1}; however, in a gray scale image V will take on a range of values, for example: V={51, 52, 53, ..., 77, 78, 79, 80}.

For this work, we used binary input images and 8-connectivity. The connected components labeling operator scans the image by moving along a row until it comes to a point *p* (where *p* denotes the pixel to be labeled at any stage in the scanning process) for which V={1}. When this is true, it examines the four neighbours of *p* which have already been encountered in the scan (i.e. the neighbours (i) to the left of *p*, (ii) above it, and (iii and iv) the two upper diagonal pixels). Based on this information, the labeling of *p* occurs as follows:

If all four neighbours are 0, assign a new label to p,

else

if only one neighbour has V={1}, assign its label to p,

else

if more than one of the neighbours have V={1}, assign one of the labels to p and make a note of the equivalences.

After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalent classes.

The complete algorithm/ for this study is summarized in Listing 1.

1. Acquire input image *I*.
2. Perform image preprocessing
3. Detect fruit regions in a sub-image of *I*. Let *N* be the number of detected fruit regions and let $R_i, i \in \{1, \dots N\}$, be the set of pixels in region *i*.
4. For each subsequent sub-image *Ij* of *I* ;
   a. Detect fruit regions $\{R_i\}\ i,\ i \in 1, \dots N_j\ in\ I_j$
   b. Classify the fruit regions $\{R_i\}\ i,\ i \in 1, \dots N_j$ into new fruit regions not associated with existing trajectories and existing fruit regions associated with existing trajectories based on

Vol. 12, No. 2, June 2019, pp. 131 - 139 *ISSN 2006-1781*
Nancy C. Woods, Emmanuel O. Abuh and A. B. C. Robert (2019), Development of a Pineapple Fruit Recognition and Counting System using Digital Farm Image

whether or not they overlap with a fruit detection in the previous sub-image.

c. For each fruit region R_i overlapping with one or more fruit regions in sub-image *j-1*, add region R_i to the closest largest set of fruit regions.

5. Perform morphological processing of the identified fruit regions.
6. Count identified fruits in the binary image using connected component labelling algorithm.

Listing 1. Pseudo code for automated system

IV. RESULTS

Figure 6 shows the output of the feature point extraction and labelling for a sample farm Image showing the 300 strongest point in the image, while figure 7 shows a region that has been marked as a pineapple fruit, given the threshold that was set.

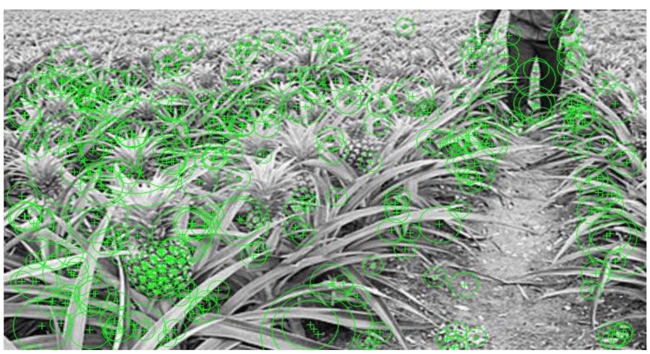

Figure 6: 300 strongest Feature Points from a farm image

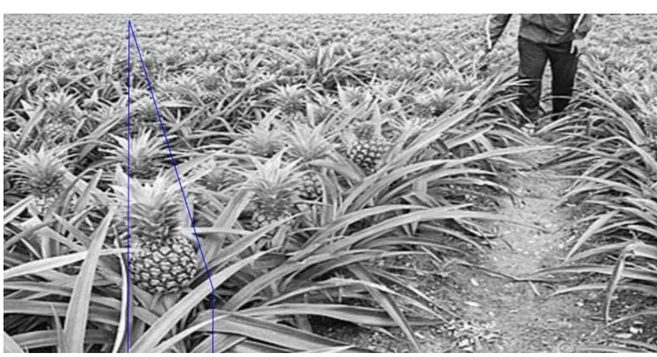

Figure 7: Identified Pineapple Fruit based on Corresponding features from figure 7

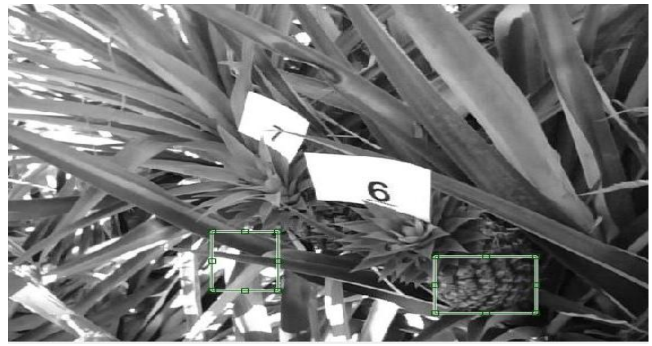

Figure 8. Fruit Detection with Hits, Misses and False Hits

There were occasions when a region in an image was classified as a pineapple fruit, because the feature points in that region were within the threshold, despite the fact that no fruit was in that region. This is shown in the first 'box' in figure 8. At other times in the study, it was observed that the algorithm detected a single pineapple fruit as two fruits (figure 9).

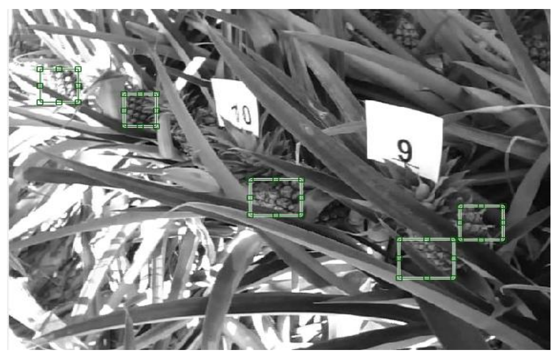

Figure 9: Fruit Detection with Fragmentations

In summary, Table 1 shows the result of pineapple fruit recognition using the algorithm specified in the methodology.

Table 1 Experimental Results of Fruit Detection

Misses	Hits	Fragmentations	Merges	False Hits
4.63%	87.33 %	2.33%	5.32%	0.39 %

The pineapple fruit recognition model performance was evaluated under five major headings specified in table 1. The percentage ratio for hits, misses, fragmentations and merges are calculated based on the actual fruit available from the ground truth. The percentage ratio for false hits is calculated based on the number images tested per time.

The algorithm presented in Listing 1, was applied to all images from the test input set. A 15-pixel radius polygon-shaped structuring element was used for the morphological closing operation, and 700 pixels was used as the threshold for pineapple region size, below which a region is considered unsuitable as a pineapple. Fruit detection evaluation results were classified into hits, misses, false alarms, merges, and fragmentations. The hit rate is the number of fruit correctly predicted, whereas the miss rate is the number of pineapples for which the detection algorithm cannot identify the presence of the fruit. False alarms occur when the algorithm predicts the presence of a fruit but there is no pineapple in that area of the image. Merges occur when an image area containing more than one fruit is reported as a single fruit. Finally, fragmentation occurs when only one fruit is present in an image area but the algorithm reports more than one fruit in that area.

The percentage analysis was calculated as follows:

$$\% \, Hits = \frac{All \; hits}{Sum \; of \; Friuts} \times 100 \quad \text{.....................} \; \textbf{(1)}$$

$$\% \, Misses = \frac{All \; misses}{Sum \; of \; Friuts} \times 100 \text{.....................} \; \textbf{(2)}$$

$$\% \, False = \frac{All \; false}{Total \; Frame} \times 100 \quad \text{.....................} \; \textbf{(3)}$$

$$\% \, Merges = \frac{All \; merges}{Sum \; of \; Friuts} \times 100 \quad \text{.................} \; \textbf{(4)}$$

$$\% \, Fragments = \frac{All \; frag}{Sum \; of \; Friuts} \times 100 \quad \text{............} \; \textbf{(5)}$$

4.1. Automated Fruit Counting Results

Using the connected component labelling algorithm, it was possible to detect and count multiple fruits in an image. Figure 10 shows two of the results of pineapple fruit counting from two different images. It can be noted that all the positive regions identified by the automated system, which ultimately translated to fruits, are labelled. In Table 2, three of the results for automated fruit counting is presented, alongside the manual count.

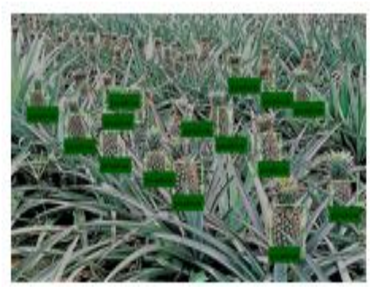

Figure 10. Fruit Counting Using Connected Component Labelling Algorithm.

Table 2: Result of the total pineapple count for manual and connected Component Labelling Algorithm.

	Counting	
Input Image	**Manual**	**Automated**
	**4**	**3**
	**3**	**3**
	**4**	**3**

V DISCUSSION

In order to evaluate the performance of the automated system, images of the farm were counted by the connected component labelling algorithm. We selected eleven images from different sections of the farm, segmented and counted all the identified regions of interest. The same set of images from the farm were counted manually by three different persons. The average of the human count was taken and recorded as the manual count value. At the end of the testing, the connected component labelling algorithm gave 23 as the number of pineapples counted while the manual count gave 27 as the number of pineapples counted. The simulation result revealed that the method is reliable, feasible and efficient when compared to manual counting method, with successful pineapple fruit detection rates of 87.37 % per frame and fruit counting success rate of 85.25%.

VI. CONCLUSION

Nowadays the automated counting systems are in high demand in the agricultural field. This work uses image processing techniques to segment out the region of interest and identify the number of pineapple fruits in digital still images. The proposed system is able to segment fruit and also detect the presence of pineapple fruits in an image. We discovered that SURF feature point detection and SURF key point descriptors, along with an SVM based classifier, gave accurate identification of pineapple fruit despite the similarity between the leaf colour and fruits at maturity.

REFERENCES

[1] A. Syal, D. Garg and S. Sharma, “A Survey of Computer Vision Methods for Counting Fruits and Yield Prediction,” *International Journal of Computer Science Engineering,* vol. 2, no. 6, pp. 346-350, 2013.

[2] J. Gill, P. S. Sandhu and T. Singh, “A Review of Automatic Fruit Classification using Soft Computing Techniques,” *International Journal of Computer Science and Electronics Engineering,* vol. 2, no. 2, pp. 99-105, 2014.

[3] S. D. Prabha and S. J. Kumar, “A Study on Image Processing Methods for Fruit Classification,” *Proc. Int. Conf. on Computational Intelligence and Information Technology, CIIT,* pp. 403-406, 2012.

[4] J. Moonrinta, S. Chaivivatrakul, M. N. Dailey and M. Ekpanyapong, "Fruit Detection, Tracking, and 3D Reconstruction for Crop Mapping and Yield Estimation," in *11th International Conference on Control, Automation, Robotics and Vision*, 2010.

[5] S. Bargoti and J. P. Underwood, "Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards," *Journal of Field Robotics,* vol. 34, no. 6, pp. 1039-1060, 2017.

[6] W. Maldonado Jr and J. C. Barbosa, "Automatic Green Fruit Counting in Orange Trees Using Digital Images," *Computers and Electronics in Agriculture,* vol. 127, pp. 572-581, 2016.

[7] U. O. Dorj, M. Lee and Diyan-ul-Imaan, "A New Method for Tangerine Tree Flower Recognition," *Computer Applications for Bio-technology, Multimedia, and Ubiquitous City. MulGraB 2012. Communications in Computer and Information Science,* vol. 353, pp. 49-56, 2012.

[8] D. Stajnko, J. Rakun and M. Blanke, "Modelling Apple Fruit Yield Using Image Analysis for Fruit Colour, Shape and Texture," *European Journal of Horticultural Science,* vol. 74(6), p. 260–267, 2009.

[9] P. K. Sethy, S. Panda, S. K. Behera and A. K. Rath, "On Tree Detection , Counting & Post-Harvest grading of fruits Based on Image Processing and Machine Learning Approach-A Review," *International Journal of Engineering and Technology (IJET),* vol. 9, no. 2, pp. 649-663, 2017.

[10] A. Gongal, S. Amatya, M. Karkee, Q. Zhang and K. Lewis, "Sensors and systems for fruit detection and localization: A review," *Computers and Electronics in Agriculture,* vol. 116, pp. 8-19, 2015.

[11] S. Chaivivatrakul and M. N. Dailey, "Texture-based fruit detection," *Precision Agriculture,* vol. 15, no. 6, p. 662–683, 2014.

[12] J. Zhang, T. Tan and L. Ma, "Invariant texture segmentation via circular Gabor filters," *Object recognition supported by user interaction for service robots, IEEE,* vol. 2, pp. 901-904, 2002.

[13] M. M. El-Gayar and H. Soliman, "A Comparative Study of Image Low Level Feature Extraction Algorithms," *Egyptian Informatics Journal,* vol. 14, no. 2, pp. 175-181, 2013.

[14] A. Jiménez, A. Jain, R. Ceres and J. Pons, "Automatic fruit recognition: a survey and new results using Range/Attenuation images," *Pattern Recognition,* vol. 32, no. 10, pp. 1719-1736, 1999.

[15] J. J. Wang, D. A. Zhao, W. Ji, J. J. Tu and Y. Zhang, "Application of support vector machine to apple recognition using in apple harvesting robot," in *In 2009 International Conference on Information and Automation, IEEE*, 2009.

[16] L. Yang, J. Dickinson, Q. J. Wu and S. Lang, "A Fruit Recognition Method for Automatic Harvesting," in *14th International Conference on Mechatronics and Machine Vision in Practice. IEEE.*, 2007.

[17] S. Kitamura and K. Oka, "Recognition and cutting system of sweet pepper for picking robot in greenhouse horticulture," in *IEEE International Conference on Mechatronics and Automation*, 2005.

[18] S. Sengupta and W. S. Lee, " Identification and Determination of the Number of Green Citrus Fruit under Different Ambient Light Conditions," *Journal of Agricultural Engineering Research,* vol. 117, pp. 51-61, 2014.

[19] R. Strogonovs, "Morf Coding and Engineering," 2017. [Online]. Available: https://morf.lv/files//SURF/. [Accessed 25 March 2018].

[20] "Connected Component Labelling Algorithm," [Online]. Available: www.github.com/ccla. [Accessed 31 March 2018].