

Detection of Cucumber Fruit on Plant Images Using Artificial Neural Network

Davood Mohammadzamani^{1*}, Seyed Mohamad Javidan², Moein Zand¹, Mohammad Rasouli³

¹Department of Biosystems Engineering, Islamic Azad University, Takestan Branch, Takestan, Iran*
E-mail: davoodmohamadzamani@gmail.com

²Department of Biosystems Engineering, Tarbiat Modares University, Tehran, Iran

³Assistant Professor, Department of Plant Breeding, Takestan Branch, Islamic Azad University, Takestan, Iran

Abstract

The main purpose of this study was to provide a method for accurately identifying the position of cucumber fruit in digital images of the greenhouse cucumber plant. After balancing the brightness histogram of the desired image, it multiplies the image with a window containing the image of a cucumber fruit, which causes larger coefficients to be obtained in areas with suspected cucumber. By extracting these local maximums, clusters of initial points are obtained as possible windows of cucumber existence. Then, in order to accurately detect the location of the cucumbers, these points and areas around them are referred to a neural network that has been trained using a number of images including cucumber images, non-cucumber images and their optimal responses. The proposed method was implemented in the Simulink toolbox of MATLAB software. The proposed method was then simulated using this network structure and tested on 120 images obtained from a greenhouse by a digital camera. The areas obtained from this network led to the accurate detection of the location of the cucumbers in the image. The proposed method was then simulated and tested on 120 images. The proposed method had a low error and was able to detect high levels of cucumber fruit in the images. This detection took an average of 5.12 seconds for each image. The accuracy of the network in correctly identifying the position of the cucumber fruit in the images was 95.3%. This method had low error and was able to detect a high rate at a good time of cucumber fruits in discover images.

Keywords: Detection, Greenhouse Cucumber, Image Processing, Artificial Neural Network

Introduction

Greenhouse cucumber is one of the important high yields horticultural crops which are widely cultivated in Iran. Due to the rapid growth of cucumber and its difficult and time-consuming harvestation, it is necessary to mechanize the greenhouse cucumber harvesting operations. One way to improve the quality and health of the production is to use automated machines without the involvement of human agents (Li, 2011). In the past, most farming operations were done manually using manpower. With recent advances in various technologies, and consequently in agricultural engineering, and in particular agricultural machinery, the role of manpower in agricultural activities has diminished (Zhao, 2016). The robot consists of three main parts, including mechanics, electronics, and software. The core of the robot's software is the visual machine, which gives the robot an understanding of the world around it and enables the robot to perform its tasks correctly. For example, in a cucumber harvesting robot, the visual

machine has the task of using image processing and machine learning techniques to determine the location of the cucumber fruit so that the robot can detect the fruit and remove it from the stem (Li, 2019). So the purpose of image processing of greenhouse cucumber bushes by the neural network is to provide a convenient and efficient way, without the need for sophisticated and specific hardware in greenhouse images processing in order to detect cucumber fruit in these images and to separate it from the plant. The results of this detection should be applicable to the mechatronic portion of the greenhouse cucumber harvesting robot.

Yuan *et al.* (2008) proposed a visual machine algorithm for detecting cucumber fruit based on near-infrared spectral imagery. They first eliminated the problem of fragmentation of similar colors in complex environments by shooting near-infrared monochromatic spectra at 85 nm. Then, to distinguish the fruit, they used image fragmentation using threshold processing. Then, by adaptive pattern extraction software, they divided the images into other sub-branches, and finally eliminated possible noises, including parts of the stem and leaf. they took. This method was tested on 40 images including 30 images containing cucumber and 10 images without cucumber taken from a greenhouse environment and the results showed that the detection accuracy was 83.3% and 100%, respectively. Yuan *et al.*, (2010) developed an algorithm for detecting cucumber fruit in a greenhouse environment. They first used a binocular stereo imaging system to capture near-infrared monochrome images to eliminate the problem of color segmentation in complex environments, and then they proposed an algorithm to identify the cucumber fruit to distribute gray matter histograms, threshold processing and noise removal using morphological analysis and finally feature extraction using pattern recognition. The results of the experiment on 120 greenhouse cucumber images showed that the proposed algorithm can detect fruit at a rate of 86%. Li *et al.* (2017) prepared a technique based on texture analysis and color analysis for detecting cucumber in greenhouse. In this research RGB image was converted to gray-scale image and HSI image to perform algorithm, respectively. MSER and HOG were applied to texture analysis in gray-scale image. The support vector machine is the classifier used for the identification task. The accuracy of this method was 83.2%. Ting *et al.* (2009) developed an optical machine algorithm for the detection of cucumber fruit based on near-infrared spectral imaging. They first eliminated the problem of splitting similar colors in complex environments by single-infrared imaging at 85 nm then, using threshold processing, image segmentation to detect fruit and split the images into subcategories by extraction software and eventually

eliminated potential noise including parts of the stem and leaves. The method was tested on 40 images containing 30 images of cucumber and 10 images of no cucumber capture in a greenhouse environment and the results showed that the accuracy of detection was 83.3% and 100%, respectively. Hayashi *et al.* (2002) have designed and fabricated an eggplant harvesting robot using an optical machine algorithm to detect eggplant fruit. The basis of their visual machine algorithm was the color characteristics and morphological characteristics of the eggplant fruit. So that in the images taken from the product using eggplant color characteristics removed it from the field and since some parts of the leaf and stem of the plant could be mistaken for the fruit as the morphological characteristics of eggplant fruit with the help of fuzzy feedback control model. Their system was able to harvest the eggplant correctly at a 62.5% rate. Liu *et al.* (2019) developed Resnet-101 for cucumber fruits detection in greenhouses based on instance segmentation. This method is selected as the backbone of Mask RCNN with feature pyramid network (FPN). The test results are compared with that of original Mask RCNN. The score of improved Mask RCNN in test results reaches 89.47%, which is higher than the other methods. The average elapsed time of improved Mask RCNN is 0.3461 s, which is only lower than the original Mask RCNN.

The main objective of this study is to present a method for real-time detection of cucumber fruit location in greenhouse images. The main difference in the fruit recognition mechanism in images with the same color spectrum is the complexity in distinguishing fruit from leaf and stem. This was also the case with images of cucumber fruit. Therefore, in this study, an innovative method was used to separate the color characteristics of cucumber fruit from background.

Materials and Methods

Proposed Method

This research was conducted in the research greenhouse of Islamic Azad University, Takestan Branch in 2021. This research seeks to provide an efficient way of detecting cucumber fruit in greenhouse images which can be used to design greenhouse cucumber harvesting robots. Therefore, this method should be far from complex so that it can be practically implemented and also does not require sophisticated hardware. Processing time was also important in this study and should be as short as possible for the greenhouse cucumber harvesting robot to have a good efficiency (Ma, 2018). Another challenge with cucumber fruit detection is that the greenhouse cucumber has a similar color to its leaves and stems, thus color and background color differences

cannot be used to detect it. So another feature should be used to reduce the complexity of the issue (Lin, 2019, Bao, 2016). Finally, a method for greenhouse cucumber detection using window correlation coefficient including an arbitrary cucumber on image and neural network classifier was presented. The proposed method consists of two parts. The first part includes finding windows that are more likely to have cucumbers, which will explain how to find them. Each window is identified by its center. The output of the first part is a number of points that are considered to contain cucumber. These places are called inaccurate centers, because these windows do not contain cucumbers and their location is slightly different from the actual cucumbers in the image. Here comes the second part, which is the inaccurate space search algorithm. In this algorithm, space exploration requires a cucumber to accurately detect windows to test the presence and absence of cucumbers. It is a neural network trained with windows with and without cucumber. These parts are presented in more detail in the following.

Inaccurate Determination of Cucumber Location

The purpose of this section is to identify inaccurate windows which may contain cucumbers. A large image contains thousands of windows. If the window is identified by its center which is a dot or a pixel, we can approximate the number of pixels of the window image by approximating the margins. The problem is a large number of windows. With a large number of operations that occur on each window, it is not possible to process all the windows at the desired time. On the other hand, cucumber fruit color information cannot be used. The main idea is to apply correlation coefficients. The correlation coefficient of a window containing an arbitrary cucumber with the cucumber in the image produces a larger value than other parts of the image such as the background. The correlation is equal to moving the window over the image and multiplying it internally. In order to determine the incorrect centers, the histogram balancing process is performed on the input image first. This is done to create the highest possible contrast. The result is then multiplied by a window of arbitrary cucumber image that produces larger coefficients in areas with suspected cucumber presence. By extracting these local maximums, clusters of initial points are obtained as inaccurate centers of potential cucumber windows. Figure 1 shows the general outline of this method.

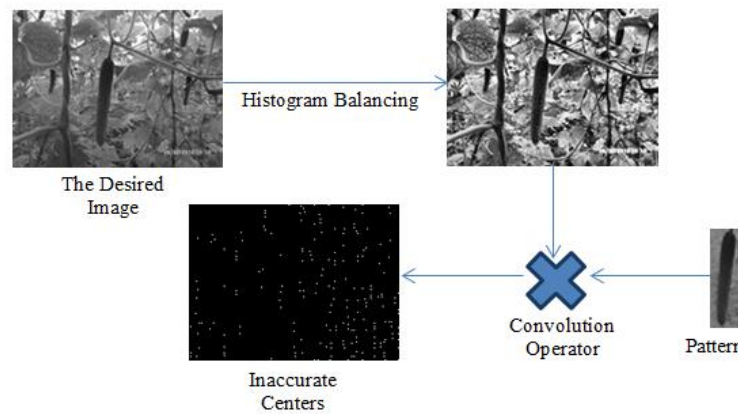


Fig. 1. Total structure for determining the inaccurate center of the windows containing cucumber

In practice, to increase the accuracy of this method and to reduce the dependence on the lighting conditions, two identical cucumber images with two different contrasts were used to determine the incorrect centers, each individually multiplied by the desired image and the sum of the points obtained.

Precise Detection of Cucumber Location

According to the test the incorrect centers in the previous section, which are centered on rectangular windows, either do not contain cucumbers in general or that the center of the windows on the cucumber center does not match the image. In this situation, if necessary, an algorithm is needed to investigate the space around these inaccurate centers to find the best answer for the presence of cucumbers. The following algorithm provides such a possibility.

- A. Inaccurate centers are first marked to refer to the network.
- B. Until a test center is not found:
 1. Return the marked centers to the neural network.
 2. If the neural network returns a value greater than the threshold for each center, the lateral pixels are marked. Finally, only those necessary areas have been investigated, and the algorithm provides a great saving in detecting cucumbers.

Extracting features

Referencing the window to the neural network is not done directly. Rather, there is a class for extracting window features, and these are the properties vectors that go to the neural network for the presence or absence of cucumber. These extracted properties make the cucumbers detection resistant to changing ambient light. The feature extraction class used the Gabor filter (Gabor wavelet transform). Gabor wavelet transform is used like other wavelet transforms due to its unique properties in the areas of image processing and visual machine (Fernandes, 2015). These wavelets provide the background for understanding the frequency of images and analyzing them in the field of location and their greatest advantage is the slow changes in the frequency domain. Gabor's 2D functions enhance the edges as well as bumps of the image. If the input image is with light intensity values of gray surfaces, the image wavelet transform is calculated according to Eq 1 from the convolution of one member of the wavelet family with the input image (Paramadiahanto *et al.*, 1996; Zhu *et al.*, 2004).

$$O_{\mu, \nu}(z) = I(z) * \psi_{\mu, \nu}(z) \quad (1)$$

Where * denotes the convolution operator, O is the result of the conversion, ψ is the wavelet used to convert with μ and magnification ν . The Gabor wavelet equation is, in fact, a simple plane wave with a specific frequency and direction trapped under a Gaussian function. This equation can be defined in different shapes and forms depending on the coordinate system, whether polar or Cartesian, and the form Eq 2 is the most common form of representation in various articles (Paramadiahanto *et al.*, 1996) (Zhu *et al.*, 2004).

$$\psi_{u,v}(z) = \frac{\|k_{u,v}\|^2}{\delta^2} \exp\left(\frac{-\|k_{u,v}\|^2 \|z\|^2}{2\delta^2}\right) \left(e^{ik_{u,v}z} - e^{-\frac{\delta^2}{2}}\right) \quad (2)$$

In Eq (2), K denotes the wave length and direction and is calculated from Eq (3):

$$\phi_{\nu} = \mu \frac{\pi}{8} k_{\nu} = \frac{k_{ma}}{f\nu} k_{\mu, \nu} = k_{\nu} e^{i\phi_{\mu}} \quad (3)$$

As it is known from Eq 3, μ is multiplied by $\pi / 8$ and has a phase number k, so it will have an integer value of zero to seven and ϕ is known as the phase offset of the function. Larger quantities produce a wave with duplicate directions. U can also have a value between zero and four and in total forty wavelets can be obtained with different directions and sizes (3 and 2). For the better analytical understanding of this wavelet, one of the 128×128 dimensions and the

parameters $\delta = 2\pi$, $k_{ma} = \pi$, $f = f = \sqrt{2}$, $\mu = 4$, $\nu = 7$ is shown in Figure 2 (Paramadihanto *et al.*, 1996) (Zhu *et al.*, 2004).

According to Figure 2, the Gabor wavelet is a plate wave with reduced amplitude. Figure 3 shows all the 45 wavelets used in this study, which five of these wavelets repeated twice. After introducing the Gabor filter here, the feature is extracted from the image of cucumber. Figure 3 shows the general structure of the feature extraction steps of a cucumber image. The image window that is to be decided on its cucumber is first of all balanced by the histogram of the gray surfaces. In the next step, the Fourier transform is multiplied by the Fourier transform of the Gabor wavelets (time-domain convolution). Then the inverse Fourier transform of the 45 images obtained is aligned side by side, making the matrix properties of the desired window. Here, if the rectangular entry window is 60×20 , then the properties matrix will be obtained from the Eq 3 and 4.

$$(60 \times 20) \quad \times \quad 45 \quad = \quad 54000$$

Eq (4)

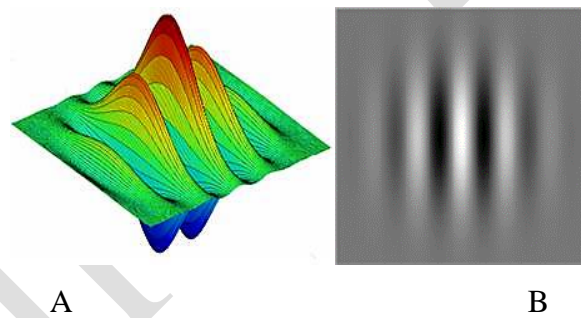


Fig. 2. Gabor wavelet representation as: A) two-dimensional B) three-dimensional

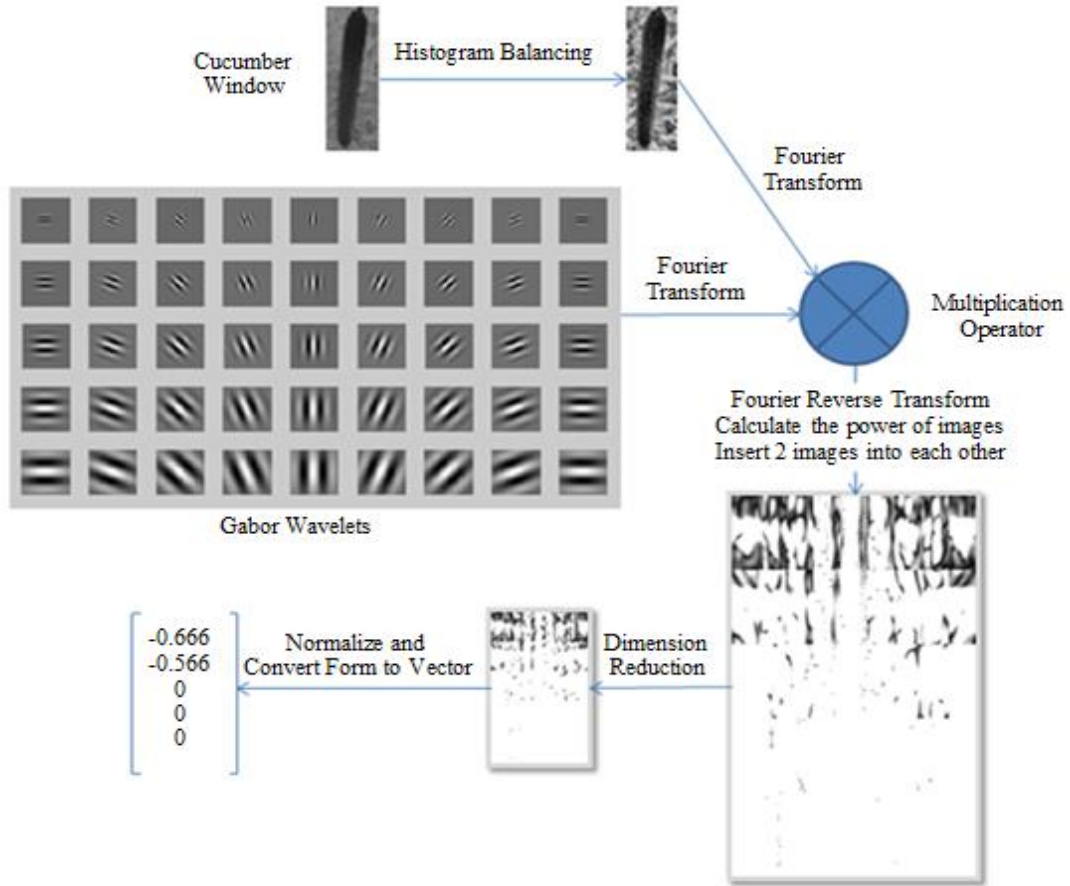


Fig. 3. Feature extraction stages by Gabor filter for a cucumber image

Obviously, this number of features (54000) is too high for processing in any classifier and will slow down the detection process. Feature reduction algorithms such as Principal Component Analysis (PCA) can be applied. However, applying this algorithm in a space of its own magnitude is time-consuming. Therefore, considering the processing time, in this study, the dimension matrix dimensionality was reduced by averaging equal blocks and averaging its 3×3 blocks. This reduced the attribute matrix to 6000 (60×100) layers. In the next step, the matrix is vectored, and then its densities are normalized to the largest dot in absolute magnitude so that all the dots are in the range $[-1, +1]$. This was done because the maximum and minimum values of the output of the Gabor filter are not specified and in neural network design the constraints between the two values are needed. At this stage, the vector properties of the image window are ready to be referred to the artificial neural network.

Artificial Neural Network Structure

Classifying the properties of the vector into two classes of cucumber and non-cucumber can be done in a number of methods. One of these methods is the artificial neural network which has been developed in recent years.

To select and design the artificial neural network structure many factors should be considered. Three prominent examples of these factors are computational volume, response time, and generalizability. Generalizability means that the network has been trained with a limited number of training data, including cucumber and non-cucumber vectors, and it is expected that the network will respond favorably to vectors that have not been previously observed. To improve network response, network structure, training data and extracted features play an important role. Neural networks are subdivided into different types from topological and structural to learning approaches, and each performs well in specific applications. Because no specific method has been devised so far to select the best network to solve a particular problem, this is done experimentally based on experimentation of different networks as well as different network structure and finally the best response.

To select the neural network suitable for the proposed method, four networks including perceptron, feedback propagation, cascade forward propagation and Elman network were simulated. In order to determine the best network for the proposed method, these networks were simulated, trained and tested. All four networks were attempted to simulate as closely as possible the same conditions. They all have two layers. The perceptron, the feedback propagation and the back-propagation cascade network have 500 neurons in the first layer and 1 neuron in the output layer and the Elman network has 25 neurons in the first layer and 1 neuron in the output layer. The training algorithm for each of the four networks is SCG and the mean squared error with regularization (msereg) rate is determined 10^{-3} and was trained for up to 200 epochs. All four networks were trained using 24 images of cucumber, 27 non-cucumber images and their optimal response. After training, the four networks were tested with 14 images in identical conditions and their performance was compared.

Perceptron multilayer neural network with error propagation learning method is one of the most commonly used networks, which in this study, after investigating and testing several networks, was considered as the best possible neural network for the proposed method to detect cucumber fruit in greenhouse images. The neural network used is a three-layer perceptron neural network with 320 neurons in the first layer and 160 neurons in the second layer and one neuron

in the output layer. The transfer function of each neuron is a tan-sigmoid whose output varies between -1 and +1 as shown in Figure 4.

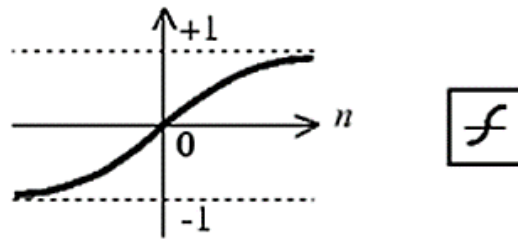


Fig. 4. Tan-sigmoid function

Figure 5 shows the structure of the perceptron network used in this study. It is notable that so far no general rule of thumb about the number of layers and neurons in multilayer neural networks has been recommended and different results can be obtained by changing the structure of the network. In this research, many networks were simulated and tested in terms of combination of number of layers and number of neurons of each layer, among which three combinations were presented that had the best results. These three combinations are:

- 1- Two layer network with 500 neurons in the first layer and 1 neuron in the output layer.
- 2- Three layer network with 320 neurons in the first layer, 160 neurons in the second layer and 1 neuron in the third layer.
- 3- Four-layer network with 300 neurons in the first layer, 100 neurons in the second layer, 20 neurons in the third layer and 1 neuron in the last layer.

The neural network identifies each cucumber with a point that corresponds to the geometric center of the cucumber fruit. This point is the center of the rectangle drawn around the cucumber fruit, and the drawing of this rectangle around the coordinates revealed by the neural network in the processed images is merely to show the result of the processing. Therefore, the output vector of the neural network(Y) includes the coordinates of the geometric center of the rectangles that the neural network has identified as the fruit of the cucumber.

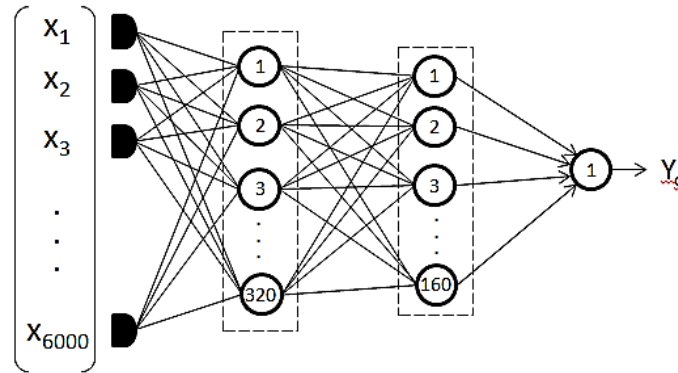


Fig.5. Perceptron network used in this research (X_i : feature vector, Y : vector containing fruit)

Neural Network Training

The neural network is trained to minimize the function of the network. Neural networks are divided into supervised and unsupervised education groups. There are several training algorithms for training neural networks in which 5 training methods have been investigated in this study to select one of them for training the network used in the proposed method. These five algorithms are Gradient Descent with Adaptive learning rate (GDA), Gradient Descent back-propagation with momentum (GDX), Scaled Conjugate Gradient (SCG), Conjugate Gradient Fletcher (CGF) and Conjugate Gradient Polak (CGP) algorithm. The five training algorithms were applied to train an identical perceptron network.

Finally, the network is weighted by the Scaled Conjugate Gradient (SCG) algorithm. The training of the network was that the images of cucumbers with the quality of $20 \times 60 \times 60$ as well as the dimensions of the windows were sent individually to the feature extraction unit and then converted to 6000 elements after extraction of the feature. Then a large matrix with each column a feature vector of one of the training images was used to train the network with the optimal response of each column equal to 0.9 for cucumber vectors and -0.9 for non-cucumber vectors. Images of cucumber are easily accessible and produced. The main problem is the selection (not the preparation) of non-cucumber images. It is obvious that these images can be customized, but the network efficiency depends on the selected images. For this purpose, the windows diagnosed as inaccurate in the network test process can be added to the non-educational image set and the training process can be repeated. In this study, 21 cucumbers and 45 non-cucumber images were used in the training phase. Also, for each image of cucumber, its mirror image as well as at angles of 5, 10 and 15 degrees in the positive and negative direction and images with one pixel

shift in all four directions were included in the training set to reduce the sensitivity of the network. For non-cucumber images, their mirrors and their 180-degree conversion were included in the training data. The proposed method was simulated by MATLAB software and run on a computer with an Intel Core i3 processor with 2 GB of memory. After simulation of the proposed method, the three-layer perceptron neural network was trained.

Results and Discussion

The three-layer perceptron network is well trained and has a minimal performance function. The results of the training of the four networks are presented in table 1.

Table 1- The results of the training of the four networks

Network	Epoch	Time	Performance
perceptron	61	2' 38"	0.000999
feedback propagation	79	3' 22"	0.000939
cascade forward propagation	132	5' 37"	0.000981
Elman	200	7' 58"	0.0021

The perceptron, the feedback propagation and the cascade forward propagation networks have reached the set performance of 10^{-3} , but the perceptron network has done so with less epochs and a shorter period of time. However since only achieving performance rates is not the determinant of the best network, these four networks need to be tested to determine which network works best. Therefore, in order to select the best network, these four networks have been tested using the proposed method for cucumber fruit detection in 14 identical images, which are presented in table 2.

Table 2- The results of four selected networks test on 14 identical images containing 20 cucumber fruits

network	average time(s)	undetected cucumbers error	incorrect detection error
perceptron	12.5	2	1
feedback propagation	15.46	3	5
cascade forward propagation	15.84	6	11
Elman	16.11	3	22

In Table 2, the undetected cucumber errors are cucumbers not detected by the network and incorrect detection errors are the points incorrectly identified as cucumbers. As can be seen from

the results, the perceptron network performs better than other networks and has less processing time. Therefore, the perceptron network is more suitable than the other networks and it is considered as the base network for the proposed method.

The network trained by the SCG algorithm has reached the minimum performance function (msereg) with less epoch than the other algorithms. This demonstrates the appropriate ability of the SCG algorithm to train the perceptron network to detect cucumber fruit in greenhouse images. After selecting the perceptron network as the most appropriate neural network and the SCG algorithm as the best training algorithm possible for this particular problem, one can now optimize the neural network structure in terms of number of layers and number of neurons per layer.

These three networks were trained after simulation by SCG algorithm with 21 cucumber images and 45 non-cucumber images and up to 100 epochs to reach 10^{-4} as msereg and then tested on 14 similar images to choose the best combination of them.

Table 3- Training results of the three proposed Perceptron network

Perceptron network	epoch	time	performance function rate
Two layers	100	3' 56"	0.000443
Three layers	37	1' 38"	0.000804
Four layers	100	2' 38"	0.000230

As shown in Table (3), the proposed three-layer perceptron network performs better than the other two networks and has achieved a set performance function value over a fewer epoch and shorter time. These three networks were tested on 14 identical images after the simulation (Table 4).

Table 4- The results of three proposed perceptron networks on 14 images including 20 cucumber fruits

Perceptron network	average time(s)	Number of undetected cucumbers	incorrect detection error
Two layers	12.5	2	1
Three layers	4.57	0	0
Four layers	14.64	2	3

The best results were obtained with a three-layer network with 320 neurons in the first layer, 160 neurons in the second layer, and 1 neuron in the output layer, which showed superiority both

in processing time and in accuracy of detection. This detection took an average of 4.57 seconds for each. The shorter processing time is important to increase the efficiency of the cucumber harvesting robot.

After simulating the proposed method and training the three-layer perceptron neural network for the detection of cucumber fruit in the images, the proposed method was tested on 120 images obtained from a greenhouse. These images were taken daily without special lighting systems and under different lighting conditions using a Samsung S1050 camera that has the optical properties suitable to use in greenhouse cucumber harvesting robots. The relevant results are presented in Table 5.

Table 5- Results of the simulated method test

Number of images	Number of cucumbers on images	Number of undetected cucumbers	incorrect detection error	Average processing time(s)
120	255	12	8	5.12

Incorrect detection error includes locations of the image that were mistaken for cucumber, which was equal to 3.13% (8 cucumbers) in these images. Obviously, these errors are due to the inability of the neural network to correctly classify which can be corrected for network configuration or training process. The undetected error was 4.7% (12 cucumbers) and included locations on the image where the cucumber was present, but was not detected as a cucumber. In other words, the accuracy of the network in correctly identifying the position of the cucumber fruit in the images was 95.3%. This result is consistent with the research of Yuan *et al.* (2008) which corresponds to 40 images including 30 images containing cucumber and 10 images without cucumber taken from a greenhouse environment and reached 83.3% and 100% detection accuracy, respectively. This error can be attributed to the neural network or may not be located in the wrong spots from the beginning. The results showed that most errors were related to inaccurate point's estimation. In other words, some of the cucumbers in the images were not identified as the center of a probable cucumber in the original estimates, in which case the presence or absence of cucumbers was not investigated. Figures 6 and 7 show the three sample images tested by the proposed method, along with the detection results and processing times of each network.



Fig. 6. Three sampled images for processing by proposed method (A: 250*188 pixels, B: 175*131 pixels and C: 210*124 pixels)

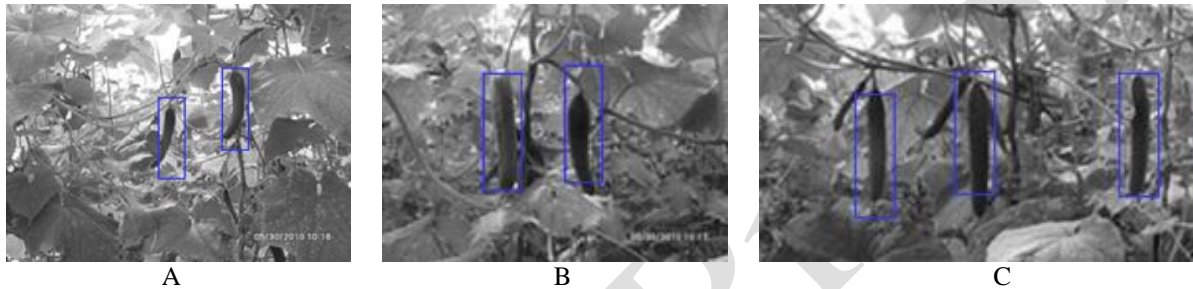


Fig. 7. Results of three processed images by proposed method (Processing time for image A: 13.16 s, image B: 3.65 s and image C: 4.49 s)

Conclusion

In this study, a method was proposed to determine the location of cucumber fruit on greenhouse images for use in greenhouse cucumber harvesting robots. Greenhouse-grown cucumber detection imposes several challenges due to the similar coloration of the fruits and the leaves, stems and flowers; a large number of occlusions, which creates high variability in the apparent shape of the cucumbers; and the random positions and orientations in which fruits can be found. The proposed method was then simulated and tested on 120 images. The proposed method had a low error and was able to detect high levels of cucumber fruit in the images. This detection took an average of 5.12 seconds for each image. Therefore, considering the results of testing this method and also considering the need of the proposed method for specific hardware and the ease of application of this method, it can be concluded that the proposed method can be an efficient method for detecting greenhouse cucumbers on greenhouse pictures. However, preliminary experimental results demonstrate the satisfactory performance of the proposed algorithm and highlight its potential benefits. The results of this study were used to detect fruit in a greenhouse cucumber harvesting robot developed by the authors.

References

Bao, G., Cai, S., Qi, L., Xun, Y., Zhang, L., & Yang, Q. 2016. Multi-template matching algorithm for cucumber recognition in natural environment. *Computers and Electronics in Agriculture*, 127, 754–762. <https://doi.org/10.1016/j.compag.2016.08.001>

Fernandes, S., & Bala, J. 2015. Study on MACE Gabor Filters, Gabor Wavelets, DCT-Neural Network, Hybrid Spatial Feature Interdependence Matrix, Fusion Techniques for Face Recognition. *Recent Patents on Engineering*, 9(1), 29–36. <https://doi.org/10.2174/2210686303666131118220632>

Hayashi, S., Ganno, K., Ishii, Y., & Tanaka, I. 2002. Robotic Harvesting System for Eggplants. *Japan Agricultural Research Quarterly: JARQ*, 36(3), 163–168. <https://doi.org/10.6090/jarq.36.163>

Jianke Zhu, Mang I Vai, & Peng Un Mak. n.d. Gabor Wavelets Transform and Extended Nearest Feature Space Classifier for Face Recognition. *Third International Conference on Image and Graphics (ICIG'04)*. <https://doi.org/10.1109/icig.2004.71>

Li, D., Zhao, H., Zhao, X., Gao, Q., & Xu, L. 2017. Cucumber detection based on texture and color in greenhouse. *International Journal of Pattern Recognition and Artificial Intelligence*, 31(08), 1754016. <https://doi.org/10.1142/S0218001417540167>

Li, P., Lee, S., & Hsu, H.-Y. 2011. Review on fruit harvesting method for potential use of automatic fruit harvesting systems. *Procedia Engineering*, 23, 351–366. <https://doi.org/10.1016/j.proeng.2011.11.2514>

Li, Z., Miao, F., Yang, Z., & Wang, H. 2019. An anthropometric study for the anthropomorphic design of tomato-harvesting robots. *Computers and Electronics in Agriculture*, 163, 104881. <https://doi.org/10.1016/j.compag.2019.104881>

Lin, G., Tang, Y., Zou, X., Xiong, J., & Fang, Y. 2019. Color-, depth-, and shape-based 3D fruit detection. *Precision Agriculture*. <https://doi.org/10.1007/s11119-019-09654-w>

Liu, X., Zhao, D., Jia, W., Ji, W., Ruan, C., & Sun, Y. 2019. Cucumber Fruits Detection in Greenhouses Based on Instance Segmentation. *IEEE Access*, 7, 139635–139642. <https://doi.org/10.1109/access.2019.2942144>

Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., & Sun, Z. 2018. A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, 154, 18–24. <https://doi.org/10.1016/j.compag.2018.08.048>

Pramadihanto, D., Wu, H., & Yachida, M. 1996. Invariant face recognition by Gabor wavelets and neural network matching. 1996 IEEE International Conference on Systems, Man and Cybernetics. *Information Intelligence and Systems (Cat. No.96CH35929)*. <https://doi.org/10.1109/icsmc.1996.569740>

Yuan, T., Li, W., Feng, Q., & Zhang, J. 2010. Spectral Imaging for Greenhouse Cucumber Fruit Detection Based on Binocular Stereovision. 2010 Pittsburgh, Pennsylvania. <https://doi.org/10.13031/2013.29858>

Yuan, T., Chen-guang, X., Yong-xin, R., Qing-chun, F., Yu-zhi, T., & Wei, L. 2008. Detecting the Information of Cucumber in Greenhouse for Picking Based on NIR Image. College of Engineering China Agricultural University Beijing 100083. [https://doi.org/10.3964/j.issn.1000-0593\(2009\)08-2054-05](https://doi.org/10.3964/j.issn.1000-0593(2009)08-2054-05)

Zhao, Y., Gong, L., Huang, Y., & Liu, C. 2016. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127, 311–323. <https://doi.org/10.1016/j.compag.2016.06.022>

Galley Proof