Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 116 (2017) 596–603

Procedia Computer Science

www.elsevier.com/locate/procedia

2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, 13-14 October 2017, Bali, Indonesia

Mango Fruit Sortation System using Neural Network and Computer Vision

Emny Harna Yossy[a], Jhonny Pranata[a], Tommy Wijaya[a], Heri Hermawan[a], Widodo Budiharto[a*]

[a] *Computer Science Department, School of Computer Sciences, Bina Nusantara University, Syahdan Campus, Jakarta 11480, Indonesia*

Abstract

Mango has different colors and sizes that indicate the level of maturity. Mango maturity level often makes farmers confused when choosing a mango that has a good maturity. Sometimes, mango farmers still use manual methods to distinguish mango maturity, while the way that human labor is often inaccurate and different in its determination. The difference is due to the different perceptions of each person. From these problems then the need of machine sorting system on agriculture is felt important. Therefore, researchers will conduct research on mango sortation system. Mango has many types such as “Harum Manis”, “Apple”, “Gincu”, etc. In this study type of mango that will be studied is mango “Gincu” because has a good color distribution. The goal of the research is to create a system that can sort mango that ripe or unripe. The method that used to do this research is separated into few step: problem identification, algorithm development, implementation and evaluation. The system is made using C language, Computer Vision and ANN (Artificial Neural Network) so the system can detect the color of mango that has been ripe or unripe. The output of this research will be compared to related research. The final output of this research is the system can detect the ripe or unripe mango with 94% accuracy.

Peer-review under responsibility of the scientific committee of the 2nd International Conference on Computer Science and Computational Intelligence 2017.

Keywords: Computer Vision; Artifial Intelligence; Neural Network; Black Propagation.

1. Introduction

Fruit is one of the foods needed by the body. Fruit has vitamins and antioxidants that can improve regeneration cell, prevent certain diseases, and various benefits. All the benefits of fruit are obtained from the fruit that has the

*Corresponding author. Tel.: +6221-534-5830; fax: +6221-530-0244.
E-mail address: wbudiharto@binus.edu

1877-0509
Peer-review under responsibility of the scientific committee of the 2nd International Conference on Computer Science and Computational Intelligence 2017.
10.1016/j.procs.2017.10.013

best quality. The best quality fruit is obtained by sorting process as the sort of fruit maturity level. Fruit maturity can be seen from skin color and size. Color becomes one of the easily recognizable traits to determine whether the fruit is ripe, undercooked, or raw. The principle that can be applied to the sorting system is color extraction. There are several previous studies that discuss the measurement of the maturity level of fruit by color.

In determining the maturity of the fruit can be through the color identification using three colors of red, yellow and green[1]. Red and yellow represent the ripe fruit, while the green represents the immature fruit. They are making a system that can check the fruit maturity viewer and star fruit sorter and do 50 times test on color paper and get 100% success, in star fruit get error of 14%, in tomato get error 4% and at Bananas get 8% error. Another case, that determines the maturity of color based on the composition of color[2]. The apple to be made histogram then compared[2]. The resulting information is the percentage of similarity and classification of fruit arrivals which include raw (18% -100%), half-cooked (12% -17%), and mature (0% -11%). Determine fruit maturity by recognizing the cucumber maturity in terms of fruit skin texture and [to] know the value of accuracy after the system is tested and its accuracy rate reaches 75%[3]. Based on the research that has been done can be seen that the sorting done on star fruit, tomatoes, bananas, apples, and cucumbers. In this study the fruit to be studied is Mango. Mango has many types such as “Harum Manis”, Apple, “Gincu”, etc. In this study type of mango that will be studied is mango “Gincu” because has a good color distribution.

Mango is tropical fruit and the tropical fruit ripeness condition we can see it from the fruit color[4], so we can use the colors of the mango fruit to measuring the ripeness level. Mango maturity level often makes farmers confused when choosing a mango that has a good maturity. Sometimes, mango farmers still use manual methods to distinguish mango maturity, while the way that human labor is often inaccurate and different in its determination. The difference is due to the different perceptions of each person. From these problems then the need of machine sorting system on agriculture is felt important. Therefore, we will conduct research on mango sortation system and build a system that can sort mango fruit automatically using neural network and computer vision that can sort mango with four category that is large – size ripe mango fruit, small – size mango fruit, large – size unripe mango fruit, and small – size unripe mango fruit. And why we use computer vision method and neural network method because computer vision can extract the fruit information from the image[5]. And neural network method can make the system to process the information like a human[6]. The objective of this paper was determined how to choose mango fruit with the good ripeness automatically with high accuracy.

2. Material and Method

2.1 Artificial Intelligent (AI)

Based on Russel and Norvig[7], there are 8 definitions of artificial Intelligence that can be categorized based on 4 approaches: (1) Thinking humanly: AI is described as a new attempt to make computers able to think, a machine that has a full mind and feel, or in other words can also be called an activity that adopts the way people think, such as decision making, problem solving, Learning, and so forth; (2) Thinking rationally: AI is described as a study through computational modeling, where this AI study can make everything possible to perceive and have reason to do; (3) Acting humanly: AI is described as an art to make machines able to display functions that require intelligence when used by humans; (4) Acting Rationally: AI is described as a study of the design of intelligent agents and AI focused on intelligent behavior.

2.2 Computer Vision

Computer vision is a branch of the study of the science of artificial intelligence that aims to help make the right decision about the description of objects and scenes that exist in an image[8]. The purpose of computer vision is to extract useful information from images[5]. OpenCV is open source computer vision library that available from *htttp://SourceForge.net/projects/opencvlibrary*[9]. The OpenCV structure consists of basic image processing and higher-level computer vision algorithms. Image processing is a process for extracting visual information needed to perform manipulation, navigation, and recognition tasks[7]. An image is a two-dimensional space that generally represents the projection of several objects in three-dimensional space[10]. An image is composed of each set whose numbers can be specified. Each element is in the corresponding location and has a value of[11]. Image segmentation is an operation to divide an image into several parts representing an object[8]. Detection is a method to identify an image that has a dominating object[12]. Thresholding is a method to convert gray scale image into binary image so that object is separated from its background[13]. Image resizing or scaling aims to analyze an image by minimize or enlarging. In computing, grayscale digital image is an image where the value of each pixel is a single sample that has information intensity. Recognition is a state when the computer gives the name of an object[9]. By using three types of receptors, humans can distinguish thousands of colours. Trichromatic Red Green Blue (RGB) in the graphics system typically uses three bytes enabling (28) 3 and produces approximately 16 million different colour codes. Each 3-byte or 24-bit RGB pixel always has one byte each (range between 0 - 255) for red, green, blue[7]. Feature extraction is the process of defining a set of features or characteristics of an image used for the classification process[14]. The features of the image can be classified under General Features and domain specific features.

2.3 Neural Network

Neural network or better known as neural network. Neural network is a system that processes information that has characteristics that are identical with biological neural networks[6]. Neural network can be used as a pattern recognition (pattern recognition), signal processing, forecasting, and other things lagi.Neural network is determined by three things, among others: (1) Pattern of relationship between neurons (network architecture): Some commonly used network architectures include: (1) Single layer network: in this network, a set of input neurons is connected directly to a set of outputs. The signal runs in the same direction from the input layer to the output layer. No vertices are in the same layer, each node connected to another node is usually above it and below it; (2) Multi layer network: this network is an extension of the single layer network, where in addition to the input and output units, there is another unit called the hidden layer. This network can solve more complex problems than a single screen network, although sometimes the training process is more complex and longer. (3) Network Recurrent: Recurrent network model is similar to single layer network and multi layer network. Only, on this network there is an output node that gives a signal to the input unit, often called feedback loop. In other words, in this model the signal goes both ways, ie forward and backward.

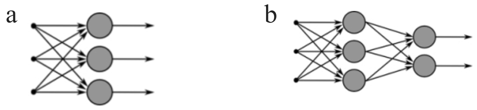

Fig.1. (a) Single Layer Network; (b) *Multi Layer Network*

Based on how to modify the weight, there are two kinds of training, among others: (1) Training with supervision (supervised): in this training, there are a number of data pairs (inputs and output targets) used to train the network to obtain the desired weight. At each training, an input is given to the network. The network will process and output. The difference between output and output target is an error. The network will modify the weights according to the error. Examples of networks that use this training are perceptron, ADALINE, and back propagation; (2) Training without supervision (unsupervised): in this training, network weight changes are based on certain parameters and the network is modified according to the parameters; (2) Activation function: The activation function is used to determine the output of a neuron. The activation function activator is net input (linear combination of input and weight). If net = $\sum x_i\, w_i$ (1), then the activation function is $f(net) = f(\sum x_i\, w_i)$ (2). Some of the frequently used activation functions include: (1) Identity function: $f(x) = x,\ untuk\ semua\ x$ (3); (2) Binary sigmoid function: $f(x) = \frac{1}{1+\mathrm{e}^{-\mathrm{x}}}$ (4).The value of this function lies between 0 and 1. The derivative of this function is: $f'^{(x)} = f(x)[1 - f(x)]$ (5); (3) Bipolar sigmoid function: $f(x) = \frac{1-\mathrm{e}^{-\mathrm{x}}}{1+\mathrm{e}^{-\mathrm{x}}}$, (6).

2.4 Back Propagation

Back propagation is one of the applications of Multi Layered Network. The back propagation training method is a supervised training algorithm for multiple-screen networks. Because the method used is monitored training, both input and output targets have been provided to train the network. The error in the data at the output layer is calculated using the output network and the target output. Then the error is propagated back to the hidden layer, giving weight changes to the synapses leading to the layer[15].

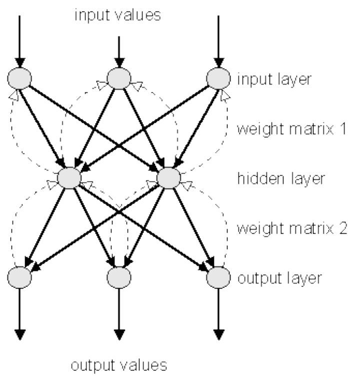

Fig.2. Back Propagation Network

2.4.1 Feed Forward Phase

Neural Network model is the simplest compared to other Neural Network models. Here, the information only goes one way, from the input layer, to the hidden layer (if any), then to the output layer. In this model there is no iteration or cycle. Mathematically, for each neuron in the ANN, get / receive the Xi input which is then modulated by a weight Wi, so the total number of inputs is: $\sum_{i=1}^{n} x_i\, v_{ij}$ (7).
Here is an algorithm from feed forward network[16] : (1) Define a network architecture consisting of input layer (xi; i = 1, 2, 3, ..., n), hidden layer (z_j; j = 1, 2, 3, ..., p), and output Layer (y_k; k = 1, 2, 3, ..., m). Input layer and hidden layer has a bias node which is usually worth; (2) Initialize the weight of the input layer to the hidden layer (v_{ij}; i = 0, 1, 2, 3, ..., n; j = 1, 2, 3, ..., p) and from hidden Layer to output layer (w_{jk}; j = 0, 1, 2, 3, ..., p; k = 1, 2, 3, ..., m) with a fairly small random value, for example from -0.5 Up to 0.5; (3) Each node on the input layer (x_i; i = 1, 2, 3, ..., n) receives the xi signal and forwards the signal to all nodes in the hidden layer; (4) Each node in the hidden layer (z_j; j = 1, 2, 3, ..., p) sums the weight of the input signal with the following equation:$\mathrm{z\_in_j} = \mathrm{v_{0j}} + \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{x_i\, v_{ij}}$ (8), and apply the activation function to calculate its output signal: $z_j = f\left(z_{in_j}\right)$ (9), then sends the signal to all nodes on the output layer; Each node in the output layer (y_k; k = 1, 2, 3, ..., m) sums the weight of the input signal: $y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j\, w_{jk}$ (10), and apply the activation function to calculate its output signal: $y_k = f(y\_in_k)$ (11).

2.4.2 Backward Phase

At step backward, weight changes from synapses are done. The amount of weight changes made is influenced by the magnitude of the error or the difference between the output network and the target output. Here is the algorithm of the backward phase (17): Calculate factor δ output unit based on error in each output unit y_k (k = 1, 2, 3, …, m). $\delta_k = (t_k - y_k) f'\left(y_{in_k}\right) = (t_k - y_k)\, y_k\,(1 - y_k)$,(12), t_k = target output; δ_k = Is the unit of error to be used in the change of weight layer below it. Calculate the weight w_kj change with the rate of understanding α:$\Delta w_{kj} = \alpha \delta_k$ (13), k = 1, 2, …, m; j = 0, 1, …, p. Calculate factor δ unit output based on error in each hidden unit z_j (j = 1, 2, …, p): $\delta_j = \delta\_in_j\, f'\left(z\_in_j\right) = \delta\_in_j\ z_j(1 - z_j)$,(14), Calculate the change rate weight v_{ji}: $\Delta v_{ji} = \alpha \delta_j x_i$(9), j = 1, 2, …, p; i = 1,2, …, n. Calculate all weight changes. Change the weight of the line leading to the output unit, namely: $w_{kj}(new) = w_{kj}(old) + \Delta w_{kj}$,(15), k = 1, 2, …, m; j = 0, 1, …, p. Changes in weight leading to hidden units, namely: $v_{ji}(new) = v_{ji}(old) + \Delta v_{ji}$,(16), j = 1, 2, …, p; i = 1,2, …, n.

The parameter α is the rate of understanding that determines the speed of the iteration. The value of α lies between 0 and 1 ($0 \le \alpha \le 1$). The larger the value of α, the less iteration occurs. But if the value of α is too large, then the correct pattern can become corrupted and slow down the learning process. Early weight selection greatly influences the neural network in the speed of training toward convergence. If the initial weight is too large, then the input to each hidden layer or output layer will fall in the area where the derivative of the sigmoid function will be very small, so if the initial weight is too small. This causes the training process to be very slow. Usually initial weight is randomly initialized with values between -0.5 to 0.5 (or -1 to 1 or other intervals). After the training is completed, the network can be used for pattern recognition. In this case, the only feed forward is used to determine the output network[17].

2.5 Proposed Model System

There are three important stages in the application that will be made, namely pre-processing process, feed forward process or introduction and training process along with optimization. The preprocessing process is used to detect and extract character features on mango fruit, feed forward process used to configure mango that is ripe, immature, rotten, and feasible for export. To use this application, the user put the mango fruit in front of the camera and then if it is appropriate the user can start training by using a certain button. The system will process the mango fruit through the camera and display on the screen whether it is mature or immature. The flow of the system works in this application can be described as follows:

Fig. 3. The flow chart of processing level.

2.6 Preprocessing process

At this phase, image is processing by computer vision method. The flow of each processing level is summarized in Figure 4.

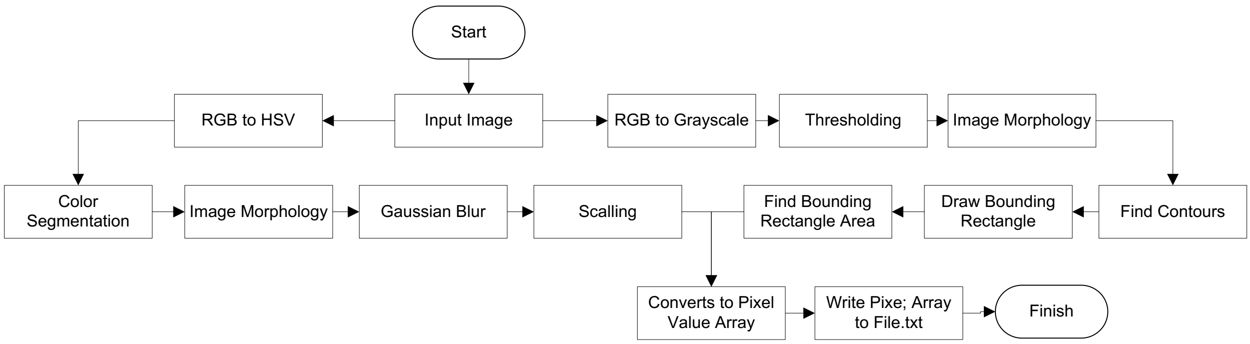

Fig. 4. The Flow of Each Processing Level.

In the first stage, the image to be recognized is determined first by the user. The image used as the input image is the image obtained via webcam with JPEG type. After the input image is obtained, a preprocessing of the input image is performed. The preprocessing process is described in the following sections. (1) RGB to HSV: At this stage the colour conversion process where the original colour taken by webcam is the RGB colour model. At this stage the RGB colour model will be converted to HSV colour model. The conversion is done using an existing library in openCV by using the cvtColour () function; (2) Color Segmentation: Once converted to HSV then the next step is to do color segmentation. Here the color in the segmentation based on 3 colors are green, orange and red. The process of color segmentation is done by determining the range of colors in the scalar HSV value. The purpose of this colour segmentation process is that when converting from pixels to arrays we can specify values for each color to be inserted into the array; (3) Image morphology: At this stage, the structuring process of the initial operating element used is an opening operation that aims to eliminate small objects by estimating the bright object with a dark background obtained from the erotion first and then dilation. The next operation used is the closing operation Aims to eliminate small holes in the image obtained from dilation results first then erotion; (4) Scaling: At this stage, the scaling process is done with the input image that has been morphed. The scaling process is done so that the image has 16x16pixel dimensions. The scaling process is performed to reduce computational time and to adjust the filter parameters used in the next preprocessing process; (5) RGB to Grayscale: At this stage the colour conversion process where the original colour taken by webcam is the RGB colour model. At this stage the RGB colour model will be converted into a Grayscale colour model. The conversion is done using an existing library in openCV by using the cvtColour () function; (6) Thresholding: At this stage the thresholding will be performed on the image that has been converted to the grayscale colour model in the previous stage. The purpose of this threshold is to make objects and backgrounds can be separated so that the system can get contours from the image in the next stage. Thresholding is done by using the functionithreshold() which is already available in OpenCV library with threshold value 100, maximum value 255 and inverted threshold binary type; (7) Find Contours: At this stage the image that has been in the threshold of the previous process will be taken contour or border and then stored in the vector. The value of this vector will be used for the next process. The process at this stage is done using the findContours () function that is already available in the OpenCV library; (8) Draw Bounding Rectangle: In this preprocessing stage the system will draw a bounding rectangle or box that will indicate the object to be measured. This Bounding rectangle will indicate the area of all contours already found in the previous find contours stage. The size of the bounding rectangle is what will be measured as the size of the object; (9) Find Bounding Rectangle Area: The bounding rectangle already drawn will be measured in this stage. To get the length or width of this bounding rectangle we use the available functions of Rect.width () and Rect.height (). The result of the function is the length and width of the bounding rectangle is in pixel units, therefore it is necessary to convert the pixel values into cm units. Conversion of pixel values into cm units using pixel * 2.54 / 96 formulas; (10) Convert pixel to array: At this stage, a process for converting a pixel image has been scaled to 16 x 16 pixels into an array which will then be used for the training process using neural network. Array 0-255 will be filled with a value array that represents the colour, the 256 array will contain Value that represents the size. The pixel array will be filled with values: 1 for green, 2 for red, 3 for orange colour, 0 for other colours, 6 for large size and 7 for small size; (11) Write Pixel Arrays Into .Txt Files: At this stage the value that has been written into pixel array will be written to file.txt which will then be used to conduct training using neural network. Value will be written to the txt file.

2.7 Design of Neural Network

In this study, neural network was built after the preprocessing image process was completed. Neural network to be built is a multicultural network with symmetrical sigmoid activation function and with back propagation training. Neural network is used to predict by colour and size. There are several aspects of neural network that will affect the ability or accuracy of training and the speed of training that we will examine in this study, namely: (1) configuration layer; (2) learning rate. Each aspect will be explained in the following sections.

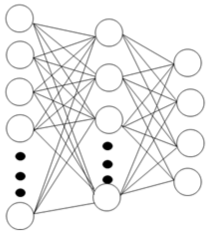

Fig.5. Neural network topology mango sortation system.

2.7.1 Learning Rate

Learning rate or Epsilon is a multiplier factor in the neural network. Learning rate affects the speed and accuracy of the training of the system we will create. In this study, to determine the value of learning rate, we will experiment against samples with varying learning rates. The value of learning rate we will use is <0.1. Where the smaller the learning rate, the training process will be longer, but will produce a higher accuracy. Whereas if the greater learning rate, the training process will be faster but the accuracy will be smaller.

2.7.2 Back Propagation Training

In this study, neural tissues will recognize the colour pattern of the mango fruit. In order for the neural network to recognize the mango fruit, the neural network needs to be trained first using back propagation training. The input data used in the training process is the image of the mango fruit which is then scaled into a 16x16 pixel size. Each pixel of the image will be converted into a pixel array containing the value that will be the input of the neuron. The input of the neuron contains 256 arrays representing colour with an input value of the range 0-3. Where value 1 represents the green colour, value 2 represents the red colour, and value 3 represents the colour of orange and value 0 for the other colours. Then in the 257 array it will be filled with the value representing the size where value 6 is for the fruit with the large size and value 7 For small fruits. The data from the training process will be recorded in the .txt file with namatrainingset.txt. Then when will do the recognition process, image pre-processing stage will convert image file into pixel array value which will be written also into file.txt with testset.txt name as data to be tested. Neural network will make an introduction by checking the input pattern written in testset.txt to the input pattern that has been trained on the system, from the check, the neural network can get the label or output value of the image you want to recognize.

2.8 Results

2.8.1 Implementation

The initial appearance of this mango fruit sortation system modeling is the console to run the system and display the image being captured by the camera. After the system is executed the next stage we have to do is put the mango fruit that is to sort into the container that has been prepared and there is a camera with a height of 30 cm from the bottom that is ready to capture the image of mango. Mango fruit will appear on the system screen the next stage we have to do is press the spacebar then the system will automatically read the value of the mango fruit and training in the neural network. The following results of prediction modeling of Mango fruit sortation system will be displayed by modeling the mango sorting system:

a b c d e

Fig. 6. (a) The final look of the system; (b) Mature small size; (c) Mature Big size; (d) Raw Small size; (e) Raw Big Size.

2.8.2 Test Result

The developed mango sortation system is able to sort the mango into four category. In order to get the optimal and high accuracy system. We do some training with the 52 sample of mango with different hidden layer and learning rate/epsilon.

Table 1. An Epsilon 0.000001

Hidden layer	True Prediction	Wrong Prediction	Error Percentage
5	48	4	7.69231%
20	47	5	9.61538%
25	47	5	9.61538%
40	49	3	5.7692%
50	45	7	13.4615%
60	48	4	7.69231%
75	43	9	17.3077%
80	45	7	13.4615%
100	43	9	17.3077%

Table 2. An Epsilon 0.0001

Hidden layer	True Prediction	Wrong Prediction	Error Percentage
5	48	4	7.69231%
20	48	4	7.69231%
25	46	6	11.5385%
40	48	4	7.69231%
50	43	9	17.3077%
60	44	8	15.3846%
75	43	9	17.3077%
80	43	9	17.3077%
100	43	9	17.3077%

With learning rate 0.000001 and use 40 hidden layer the mango sortation system can sort mango accurately with 94% success percentage. The following graphs obtained from the epsilon comparison of the three tables above are:

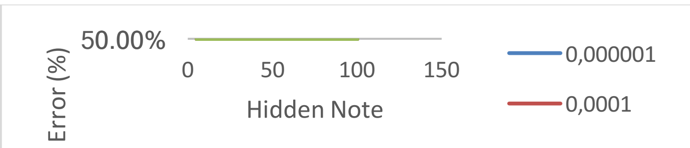

Fig. 7. Epsilon Comparison Chart

From the above graph can be known by using epsilon 0.000001, epsilon 0.0001 with hidden node is 40 then obtained a low error percentage and with the speed of training process 4 seconds.

2.9 Evaluation

Evaluation is used for objective valuation. Objective evaluation is done by comparing the benchmarks used in this study, namely recognition accuracy, training speed, and comparison of similar studies that use the color as an interpreter of the fruits. In a study conducted by Sunu and Dwi[2] is measuring the maturity of apples using image processing based on color composition or color histogram, 100% accuracy and training speeds were not mentioned but the drawback of this study was not to take imagery in real time. While research conducted by Permadi and Murinto[3] which measure the maturity level of fruit with extraction cirri statistic skin color, level of drain imencapai 75% and process of training speed reach 1 minute. Another research conducted by Radityo, Fadillah, Igwahyudi, Dewanto[1], made the sorter and fruit maturity based on the color sensor, the rate reached 100%. But if the surface of the fruit is not flat, then the higher the error rate.Here's an objective evaluation table:

Table 3. Objective Evaluation Table.

Research	Accuracy	Speed of Training Process	Deficiency
Sunu Jatmika and Dwi Purnamasari[2]	100%	-	System Cannot be implemented in real time.
Yuda Permandi and Murinto[3]	75%	1 minute	Training process long enough.
Dimas Rizki Radityo, Muhammad Riyan Fadillah, Quincy Igwahyudi, Satrio Dewanto[1]	96%	-	Uneven fruit surfaces increase the percentage of errors.

Based on the above data can be concluded that mango sorting system modeling has faster training process speed from similar research that only takes 3 seconds and also the percentage of accuracy is high enough that reaches 94% and has the advantage can be implemented in real time and can calculate the size of mango.

3. Conclusion

Based on the previous discussion, the conclusion can be drawn as follows: modeling this mango fruit sortation system can utilize neural network and computer vision method to classify mangoes appropriately with high accuracy reach 94%. Two aspects that determine the speed and accuracy of the mango clipping are finding the optimal number of hidden layers while for the learning rate the smaller value will make the learning process slower but the accuracy will increase otherwise the greater the learning rate will make the training process faster but the value small accuracy.

References

1. Radityo DR, Fadillah MR, Igwahyudi Q, Dewanto S. Alat Penyortir dan Pengecekan Kematangan Buah Menggunakan Sensor Warna. Tek Komput [Internet]. 2012;20(2):88–92. Available from: http://research-dashboard.binus.ac.id/uploads/paper/document/publication/Journal/Teknik Komputer/Vol 20 No 2 Agustus 2012/02_Satrio-OK.pdf.
2. Jatmika S, Purnamasari D, Malang A. Rancang Bangun Alat Pendeteksi Kematangan Buah Apel dengan Menggunakan Metode Image Processing Berdasarkan Komposisi Warna. J Ilm Teknol dan Inf ASIA [Internet]. 2014;8(1):8. Available from: lp3m.asia.ac.id.
3. Permadi Y, Murinto. Aplikasi Pengolahan Citra untuk Identifikasi Kematangan Mentimun Berdasarkan Tekstur Kulit Buah Menggunakan Metode Ekstraksi Ciri Statistik. J Inform [Internet]. 2015;9(1):1028–38. Available from: http://journal.uad.ac.id/index.php/JIFO/article/view/2044.
4. Ary Noviyanto. Klasifikasi Tingkat Kematangan Varietas Tomat Merah dengan Metode Perbandingan Kadar Warna. 2009;1(1):12.
5. Simon J.D.Prince. Computer Vision: Models, Learning, and Inference [Internet]. Cambridge University Press; 2015. Available from: http://www.computervisionmodels.com/.
6. L F. Fundamentals of Neural Networks Architectures, Algorithms and Applications. 11994.
7. Stuart Russel, Norvig P. Artficial Intelligence: A Modern Approach. Prentice Hall; 2010.
8. Linda G. Shapiro, Stockman GC. Computer Vision. 1st ed. Pearson; 2001.
9. Bradski G, Kaehler A. Learning OpenCV. O'Reilly Media; 2008. 580 p.
10. Trussell J, Vrhel M. Fundamentals of Digital Imaging. London: Cambridge University Press;
11. Gonzalez RC, Woods RE. Digital Image Processing. 2008.
12. Forsyth D, Ponce J. Computer Vision: A Modern Approach. Pearson; 2012. 549 p.
13. Jain R, Kasturi R, Schunck BG. Machines Vision. McGraw-Hill, Inc.; 1995.
14. Shin H, Riaz I. Detection of Vehicles and Pedestrians. In: Kim J, Shin H, editors. Algorithm & SoC Design for Automotive Vision Systems For Smart Safe Driving System. Springer London; 2014. p. 107.
15. L.Pinjare S, Kumar M A. Implementation of Neural Network Back Propagation Training Algorithm on FPGA. Int J Comput Appl. 2012;52(6):1–7.
16. Sutojo T, Mulyanto E, Vincent S. Kecerdasaan Buatan. Andi; 2011.
17. Rahajaan J. Pengembangan Wahana Pencitraan Bawah Air Guna Identifikasi dan Kuantifikasi Terumbu Karang dengan Metode Jaringan Syaraf Tiruan. MSc Thesis, Bogor Agriculture University. 2011.