



Determination of Lycopersicon maturity using convolutional autoencoders

I-Hsi Kao, Ya-Wen Hsu, Ya-Zhu Yang, Ya-Li Chen, Yi-Horng Lai, Jau-Woei Perng*

Department of Mechanical and Electro-Mechanical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan



ARTICLE INFO

Keywords:

Agriculture
Artificial neural networks
Computer vision
Image processing
Supervised learning

ABSTRACT

The field of computer science is witnessing the development of new and advanced applications in agricultural science and related technologies. Accurate evaluations of the ripeness of fruits and vegetables are very important in agricultural science as fruit growers can profit from the automatic detection and interpretation of fruit maturity levels. In this study, we propose a method of classifying Lycopersicons based on three maturity levels (immature, semi-mature, and mature). Our method includes two artificial neural networks, a convolutional autoencoder (CAE), and a backpropagation neural network with a Softmax layer. A CAE involves the convergence of convolutional neural networks and an autoencoder and has recently gained considerable attention in the field of Engineering. However, a traditional backpropagation neural network also plays an important role in the proposed method. To adapt the classification system to various complex scenarios, the CAE functions as a background filter, and it determines the region of interest (ROI) in an image. A contribution of this study is the use of a CAE to determine the ROI in the Lycopersicon image instead of tuning handcrafted parameters manually to set the ROI. The machine detects the Lycopersicon through self-learning mechanisms. Using the extracted features, the machine employs self-learning mechanisms to determine Lycopersicon maturity. The experimental results demonstrate that our method can recognize maturity levels with an accuracy rate of 100%. Therefore, the proposed algorithm provides objective and useful information concerning maturity to optimize the harvest time of Lycopersicons.

1. Introduction

Lycopersicons are native to Central and South America. They are widely cultivated as edible vegetables and exhibit a certain medical efficacy. Lycopersicons are some of the model organisms of Solanaceae flowering plants and there are many Lycopersicon varieties. Most fruit varieties are red while some are orange, yellow, green, purple, pink, and white; moreover, Lycopersicons with colored stripes also exist. According to the researcher C. M. Rick, Lycopersicons can be categorized into the following nine varieties: *Lycopersicon esculentum*, *Lycopersicon pimpinellifolium*, *Lycopersicon cheesmanii*, *Lycopersicon parviflorum*, *Lycopersicon chmielewskii*, *Lycopersicon hirsutum*, *Lycopersicon chilense*, *Lycopersicon peruvianum*, and *Lycopersicon pennellii*. The most famous genetic features of Lycopersicons were reported by C. M. Rick in 1956 (Rick and Butler, 1956).

In this study, we consider salad tomatoes and beefsteak tomatoes because of their extensive cultivation. The types of salad tomatoes include Pantano Romanesco, Costoluto Genovese, Sweet Clusters, Carmello, and Early Girl. The types of beefsteak tomatoes considered include Big Beef, Brandywine Pink, Caspian Pink, and Mortgage Lifter. These types of beefsteak tomatoes are very common in Lycopersicon

farms in Taiwan, and most of them belong to the *Lycopersicon esculentum* group. Though our study targets the types of beefsteak tomatoes grown on Lycopersicon farms in Taiwan, our proposed method is equally applicable for other targeted varieties.

Computing technology has been recently extended to the food industry. Analysis of fruit maturity has always been a topic of concern. In (Effendi et al., 2011), an image recognition system for identifying the maturity level of *Jatropha Curcas* fruits was proposed. The image recognition system classifies *Jatropha Curcas* fruits into various categories. The recognition procedure is completed by adopting a backpropagation diagnosis model and by using image pixels as inputs. According to the experimental results, the backpropagation diagnosis model exhibits an accuracy of 93%.

In (Alonso-Salces et al., 2005), a high-performance liquid chromatography-diode array detector was used to analyze the thiolysis and direct solvent extracts of freeze-dried apple pulps and peels. The authors observed that univariate data treatment presents several challenges to achieving the goal mentioned in the experiment. Therefore, a multivariate approach was considered. Supervised pattern recognition procedures such as linear discriminant analyses, K- nearest neighbors, soft independent modeling of class analogies, and multilayer feed-

* Corresponding author.

E-mail address: jwperng@faculty.nsysu.edu.tw (J.-W. Perng).

forward artificial neural networks were used to develop a heuristic technique to classify samples into established categories. The results demonstrate accuracies of 97% and 99% for the predictions of unripe and ripe apples, respectively.

In (Asnor et al., 2013), J. I. Asnor et al. argued that farmers could not properly follow the guidelines provided by the Federal Agriculture Marketing Authority because experts and farmers have different perspectives on the maturity classification of pineapples, owing to their varied experiences. To help farmers in identifying the maturity of pineapples, the authors proposed an automatic maturity recognition system. They extracted red and green pixels as color features and used them as inputs for the artificial neural network. The experimental results exhibit an accuracy of 75%.

In (Rodríguez-Pulido et al., 2012), 21 phenolic compounds were detected using a high-performance liquid chromatography diode array detector, and mass spectrometry was proposed as the standard for diagnosing the phenolic maturity level of grape seeds. Using the DigiEye® system, the CIELAB parameters, including area, aspect, roundness, length, width, and heterogeneity of seeds were analyzed. This study provides a useful and efficient tool for optimizing decisions during harvest time.

Further research on grape maturity estimation was conducted through the recognition of seed images using neural networks (Zuñiga et al., 2014). By using a pattern recognition methodology and supervised neural networks, a maturity system was constructed, which classifies the grape phenolics as overly mature, mature, or immature. The experimental results offer objective and useful information concerning the optimal harvest times for grapes.

In the context of studies on the maturity of Lycopersicon, F. Hahn predicted the maturity stages using backpropagation neural networks (Hahn, 1999). According to the results, the spectral signatures of each maturity stage in Lycopersicons are different. After extracting the respective wavelength ratio of a Lycopersicon, the backpropagation input is confirmed. The experimental results demonstrate that the accuracy of using neural networks to recognize the maturity level of Lycopersicons is higher than that obtained by the use of discriminant analysis.

In (Wang et al., 2011), X. Wang et al. proposed a judgment system for the maturity level of Lycopersicons under natural conditions to determine the optimal harvest time. The work divided Lycopersicon maturity into five different maturity stages: the red, light-red, pink, turning, and breaker stages. A multi-spectral camera and charge-coupled device camera were employed. The near-infrared images at a wavelength of 810 nm were screened by the multi-spectral camera. The judgment procedure was completed after analyzing the pixel values. The experimental results demonstrate that the red-green mean method delivers more satisfactory results than the hue-mean method, with an accuracy rate of above 96%.

In (Xiao et al., 2015), the surface color-changing process of Lycopersicons was predicted using the changing stages based on temperature conditions. The experimental results allow farmers to determine dynamically, the ripening process during every maturity stage, thereby optimizing their income.

In (Wan et al., 2018), backpropagation neural networks and color feature extraction were employed.

The average accuracy rate for detecting the maturity level of Lycopersicons is 99.31%.

Other Lycopersicon maturity determination methods without a color image format are mentioned in (Zhang and McCarthy, 2012; Gómez et al., 2006; Clément et al., 2008; Lien et al., 2009). This study proposes a completely new architecture that has proven effective in experiments.

Two systems, a background removal system and recognition system, are presented. The background removal system is implemented based on a deep convolutional autoencoder (CAE) (Leng et al., 2015; Holden et al., 2015; Chen et al., 2017; Makhzani and Frey, 2015) structure.

In (Leng et al., 2015), the author used a stacked local convolutional

autoencoder to extract features of 3D objects. By training each layer to regularize parameters, the well-trained deep model for detecting 3D object features is obtained. In (Holden et al., 2015), the author used a convolutional autoencoder to learn the motion manifolds of the Kinect motion capture system. Through this technology, corrupt motion data can be corrected. In (Chen et al., 2017), the author used the convolutional autoencoder as an unsupervised learning structure. By using the convolutional autoencoder to extract features from an image, only a small amount of label data is needed for efficient feature learning. In (Makhzani and Frey, 2015), the author reported that autoencoders can be used to learn deep sparse representations from the MNIST, CIFAR-10, ImageNet, Street View House Numbers, and Toronto Face datasets, and achieve competitive classification performance. Numerous studies demonstrate the use of autoencoders to achieve their goals.

Although the architecture used in these research works is similar, the use of the convolutional autoencoder has not been fully explored. In this study, CAE is used for background removal, whereas most of the previous works use CAE for feature extraction. In this work, we discovered a different application for CAE.

In this study, using the CAE, the background of the Lycopersicon image is removed. After the background has been removed, the area of an inscribed circle is used to extract Lycopersicon features from the image. By using the maximal inscribed circular area of a Lycopersicon surface, the feature extraction area can be identified. The inscribed circular area is divided into five concentric color feature areas (CFAs). The features of the Lycopersicon image are used as inputs of the backpropagation neural network to determine the maturity level. Finally, the extracted features are used as inputs for the backpropagation neural network, which determines the maturity of the Lycopersicon by calculating the Softmax layer.

This study is based on (Wan et al., 2018) with significant changes. The main changes include the addition of a background removal system and the removal of handcrafted feature steps. By adding the background removal system, the method can detect the Lycopersicon in the image. Therefore, the system recognizes maturity even with complex image backgrounds.

The background removal system is based on a CAE. The addition of a background removal system is the primary difference between the procedure in (Wan et al., 2018) and the procedure proposed in this study. The additional step enables the system to detect Lycopersicons with more complex image backgrounds.

The remainder of this paper is organized as follows. In Section 2, the three Lycopersicon maturity types are presented. The three maturity levels depend on the interpretation of Lycopersicon cultivation experts. Lycopersicon are classified as immature, semi-mature, or mature. The system structure and algorithms are explained, and in Section 3, the experimental results and the analysis of the results are presented. Finally, Section 4 provides a conclusion and suggests a topic for future research.

2. Materials and methods

In this section, we present the Lycopersicon samples and explain the system structures and their importance. The method includes two systems: a background removal system and a recognition system.

2.1. Lycopersicon samples

The following types of Lycopersicon are chosen: salad tomatoes and beefsteak tomatoes, which originate from Guanxi Township, Hsinchu County 306, Taiwan (R.O.C.).

Farmers in Hsinchu County categorize Lycopersicon maturity into four levels: pre-immature, immature, semi-mature, and mature; “pre-immature” implies that the Lycopersicon can still grow; “immature” means that the Lycopersicon cannot grow anymore. However, at this maturity level, the peel is completely white and green, the flesh is hard,

and the flavor is poor. The semi-mature level implies that the Lycopersicon begins to change color from green to red; “mature” means that over 3/4 of the Lycopersicon turns red, pink, or yellow. At this maturity level, the highest nutritional value can be derived. Therefore, it is the optimal maturity level for fresh Lycopersicons.

Lycopersicons eaten as vegetables are generally harvested 40–50 days after flowering. When the fruit reaches maturity, it must be harvested in a timely fashion. During summer and autumn, Lycopersicons change color faster than during spring, and they mature faster. The fruits picked for sales to nearby locations should be harvested during the semi-immature stage; the fruits picked for long-distance transportation should be harvested during the immature stage.

The harvest time of Lycopersicons can be erratic owing to environmental changes such as temperature, humidity, and changes in the sunlight. In the past, harvest time was determined based on the farmer's expertise in determining the optimal harvest time for Lycopersicons. Farmers determined the maturity stage by considering the color, weight, hardness, and odor. However, for the fully automated farms of the future, non-human identification technology for Lycopersicon maturity is of critical importance.

In this study, three maturity levels are considered: immature, semi-immature, and mature. There are 16,200 experimental images, with 5400 images for each stage of maturity. The image samples of three Lycopersicon maturity levels are shown in Fig. 1.

2.2. Background removal system

The background removal system is based on a CAE. This system was proposed in (Masci et al., 2011) and it is architected as an unsupervised feature-learning structure. The CAE is trained by using a conventional online gradient descent and no additional regularization terms are required. The CAE combines the autoencoder (Deng et al., 2010) and convolutional neural networks (LeCun et al., 1990).

Convolutional neural networks are widely used in image processing and video recognition systems (Makhzani and Frey, 2015; Masci et al., 2011). A convolutional neural network includes a convolutional layer and a pooling layer.

A convolutional filter is applied to the entire input image. The mathematical equation of the output is:

$$c_{i,j} = f_{act} \left(\sum_{m=1, n=1}^h \omega_{m,n} \rho_{i,j} + \varrho \right) \quad (1)$$

where ω is the convolutional filter of size $m \times n$; h is the max size of the convolutional filter; $\rho_{i,j}$ the pixel value of the image position $x = i, y = j$; ϱ the bias of the convolutional filter; f_{act} the activation function; $c_{i,j}$ the pixel value of the output images.

Next, the feature maps generated by the convolutional layer are passed through a pooling layer. In this study, the max-pooling function was applied (Scherer et al., 2010) in the pooling layer.

The autoencoder is a fully connected neural network that conducts feature extraction and data compression. It includes two structures, i.e., an encoder and a decoder.

The encoder compresses the data whereas the decoder reverts it.

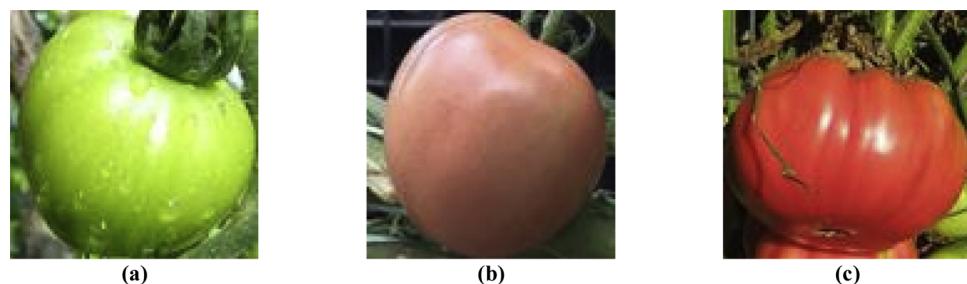


Fig. 1. Lycopersicon maturity levels: (a) immature, (b) semi-immature, and (c) mature.

However, by reverting the data, the original input data cannot be obtained. An advantage of the autoencoder is that it learns encoding and decoding on its own and reduces handcrafted parameters. The encoder and decoder can be described as follows:

$$\phi, \psi = \arg \min_{\phi, \psi} \|x - (\psi \circ \phi)x\|^2 \quad (2)$$

where ϕ is the encoder, ψ the decoder, and x the input. The following is used as the encoder input:

$$x \in \mathbb{R}^d = X \quad (3)$$

and is mapped onto:

$$z \in \mathbb{R}^p = F \quad (4)$$

where F is the output of the encoder. The compressed features z can be obtained using:

$$z = f_{act}(\omega x + \varrho) \quad (5)$$

where ω are the weights. After the encoder completes the encoding process, z is reconstructed by the decoder to equalize \hat{x} and x to revert the input; i.e.,

$$\hat{x} = f'_{act}(\hat{\omega}z + \hat{\varrho}) \quad (6)$$

where the parameters $\hat{\varrho}$, $\hat{\omega}$, and f'_{act} might not be equal to those of the encoder.

The implementation process of the CAE is consistent with that of the autoencoder. Both run the encoding process followed by the decoding process. The CAE is widely used to reconstruct images. It can learn to execute the removal of disturbances or reconstruct lost portions of images. To revert the input image, a structure of transposed convolutional neural networks is added. Transposed convolution was proposed in (Radford et al., 2015). The transposed convolutional generative adversarial networks in (Radford et al., 2015) are part of a larger class of generative adversarial networks used to generate an image. In this study, the transposed convolution step performs a role similar to that of image generation in the larger class of networks; however, it does not generate images but reverts them. The common padding mode includes a valid mode and a same mode (Dumoulin and Visin, 2016). In addition, the padding method in the convolutional and transposed convolutional layer uses the same mode to set the size as a constant value.

In order to tune a convolutional autoencoder for optimal performance, experience is very important. When designing a CAE, the following concepts must be satisfied: the filter size decreases from large to small and the filter quantity decreases from more to less. A pooling layer must be connected to follow the convolutional layer, and batch normalization must be completed after processing by the pooling layer. Therefore, we propose to adjust the parameters based on the above principles, relying on experience and past experiments to tune the parameters of the neural network. When the training error is more than the setting rate, the number of layers or the size of the convolutional kernel must be increased. When the testing error is greater than the setting rate, the number of layers or the size of the convolutional kernel must be reduced. Through this method, we can insure that the error can be reduced without overfitting.

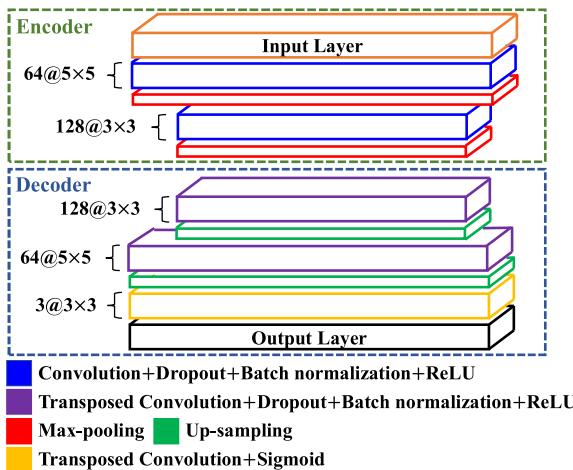


Fig. 2. CAE architecture of the proposed method.

After tuning the CAE architecture by considering the experimental results, it generally includes two convolutional layers, two max-pooling layers, three transposed convolutional layers, two up-sampling layers, and four batch normalization layers, as shown in Fig. 2. Convolutional layers 1 and 2 have 64 filters with a size of 5×5 and 128 filters with a size of 3×3 . The transposed convolutional layers 1 and 2 have 128 filters with a size of 3×3 and 64 filters with a size of 5×5 . The size of the max-pooling layers and up-sampling layers is 2×2 . The size of the input layer and output layer is the image size.

After passage of an image through each convolutional layer or transposed convolutional layer, there follows a batch normalization process (Ioffe and Szegedy, 2015). Using this step, vanishing-gradient problems are reduced. The output of the neural network is normalized via batch normalization, and weight and bias are added after the output layer. The output layer is normalized according to:

$$B_i = \dot{x}_i - \mu_B / \sqrt{\sigma_B^2 - \varepsilon} \quad (7)$$

where \dot{x} is the output set of the neural networks, μ_B the average of the output, σ_B^2 is the variance, $\sqrt{\sigma_B^2 - \varepsilon}$ is the standard deviation of the output, ε is a negligible positive number to avoid a division by 0, and B is the output set of batch normalization; μ_B and σ_B^2 can be described as follows:

$$\mu_B = 1/n \sum_{i=1}^n (\omega_i x_i + \varrho) \quad (8)$$

$$\sigma_B^2 = 1/n \sum_{i=1}^n (\omega_i x_i + \varrho - \mu_B)^2 \quad (9)$$

where n is the number of output neurons. Under this condition, B is limited to a normal distribution. The characterization ability of the neural networks may be reduced after the normalization step. To solve this problem, new weights and biases are added after the normalization. Using this step, higher output values will be reduced by the normalization step. However, if the output value is too high for the neural networks, the added weight and bias will increase the output value.

A rectified linear unit (ReLU) (Nair and Hinton, 1995) is chosen as the activation function of the convolutional layers, and a sigmoid function (Han and Moraga, 2005) is chosen as the activation function of the last transposed convolutional layer. The following equation describes the ReLU and sigmoid:

$$f_{ReLU}(x) = x^+ = \max(0, x) \quad (10)$$

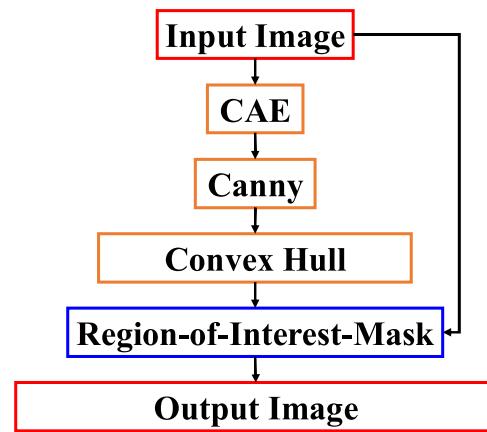


Fig. 3. Lycopersicon background removal system.

$$f_{Sigmoid}(x) = e^x / (e^x + 1) \quad (11)$$

A dropout process is also added to the CAE. It was first mentioned in (Srivastava et al., 2014). By forcing some neurons to stop working in each training epoch, the overfitting phenomenon can be reduced. The dropout rate in this work is 80%.

The CAE produces images similar to the label images by using the input images. However, differences between the produced images and the label images still exist. The signal-to-noise ratio (SNR) and the peak signal-to-noise ratio (PSNR) between the produced images and the label images are calculated; their values prove that the effect of the CAE is limited. However, the Lycopersicon edges in the produced and label images are similar. A Canny edge detector (Canny, 1986) is used to detect the edges of the produced image. A convex hull (Barber et al., 1996) is then used to detect the edges of the Lycopersicon. A region-of-interest (ROI) mask is constructed using the edges of the Lycopersicon and is superimposed on the input image. By following the steps above, the background is removed and the Lycopersicon is detected. The background removal procedure in this study is shown in Fig. 3.

2.3. Recognition system

The recognition system in this study is a simple machine-learning structure that includes feature extraction and classification. The feature extraction divides the Lycopersicon image area into five CFAs by using the maximal inscribed circle. The ROI mask is a simple signal used to detect the maximal inscribed image area and five CFAs. The inscribed circle is:

$$\begin{cases} x_c = 1/m \sum_{i=1}^m x_i \\ y_c = 1/m \sum_{i=1}^m y_i \end{cases} \quad (12)$$

$$r = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (13)$$

where (x_c, y_c) is the center point of the inscribed circle, (x_i, y_i) is the pixel position of the edge, r is the distance from the Lycopersicon center to the edge, and m is the total number of pixel points in the ROI mask image. The circular shape with radius r can be determined with r_{min} in the ROI mask image. Therefore, the distances between the CFAs are equal, as shown in Fig. 4.

After determining the five CFAs, the mean values of the RGB color values are extracted from the five CFAs by using the following equations:

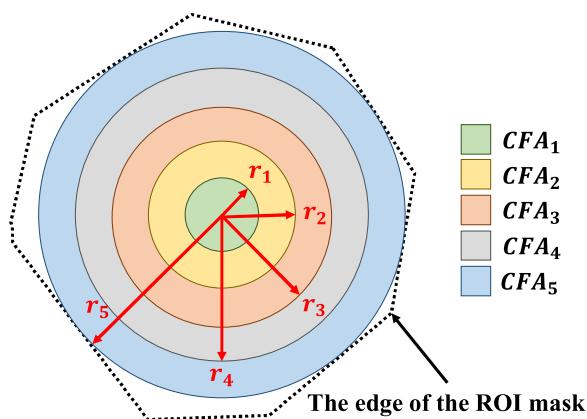


Fig. 4. Five color feature areas (CFAs) and maximal inscribed circle of Lycopersicon sample.

$$\begin{cases} \bar{R}_z = 1/m_z \sum_{j=1}^{m_z} R_{zj} \\ \bar{G}_z = 1/m_z \sum_{j=1}^{m_z} G_{zj} \\ \bar{B}_z = 1/m_z \sum_{j=1}^{m_z} B_{zj} \end{cases} \quad (14)$$

where \bar{R}_z , \bar{G}_z , and \bar{B}_z are the average values of the RGB color values for all pixel points in the CFAs, z is the number of the five CFAs, j is the number of pixel points in each CFA, and m is the total number of pixel points in the five CFAs.

The sensitivity of human vision is represented by the HSI (hue, saturation, intensity) color format (Helson, 1938). When extracting color features, it is unwise to extract only the RGB. Therefore, the proposed system also extracts the HSI, hue, saturation, value (HSV) (Smith, 1978), and RGB color formats. The conversion between RGB and HSV is governed by the equations:

$$H_z = \begin{cases} \text{undefined, if } \max = \min \\ 60^\circ \times \frac{\bar{G}_z - \bar{B}_z}{\max - \min}, \text{ if } \max = \bar{R}_z \text{ and } \bar{G}_z \geq \bar{B}_z \\ 60^\circ \times \frac{\bar{G}_z - \bar{B}_z}{\max - \min} + 360^\circ, \text{ if } \max = \bar{R}_z \text{ and } \bar{G}_z < \bar{B}_z, \\ 60^\circ \times \frac{\bar{B}_z - \bar{R}_z}{\max - \min} + 120^\circ, \text{ if } \max = \bar{G}_z \\ 60^\circ \times \frac{\bar{R}_z - \bar{G}_z}{\max - \min} + 240^\circ, \text{ if } \max = \bar{B}_z \end{cases}$$

$$S_z = \begin{cases} 0, \text{ if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, \text{ otherwise,} \end{cases}$$

$$V_z = \max, \quad (15)$$

where max is the maximum value among the RGB values and min the minimum value among the RGB values. The equations for the conversion of RGB and HSI are:

$$H_z = \begin{cases} \text{undefined, if } \max = \min \\ 60^\circ \times \frac{\bar{G}_z - \bar{B}_z}{\max - \min}, \text{ if } \max = \bar{R}_z \text{ and } \bar{G}_z \geq \bar{B}_z \\ 60^\circ \times \frac{\bar{G}_z - \bar{B}_z}{\max - \min} + 360^\circ, \text{ if } \max = \bar{R}_z \text{ and } \bar{G}_z < \bar{B}_z, \\ 60^\circ \times \frac{\bar{B}_z - \bar{R}_z}{\max - \min} + 120^\circ, \text{ if } \max = \bar{G}_z \\ 60^\circ \times \frac{\bar{R}_z - \bar{G}_z}{\max - \min} + 240^\circ, \text{ if } \max = \bar{B}_z \end{cases}$$

$$\dot{S}_z = 1 - \frac{3 \times \min(\bar{R}_z, \bar{G}_z, \bar{B}_z)}{\bar{R}_z + \bar{G}_z + \bar{B}_z},$$

$$I_z = \frac{\bar{R}_z + \bar{G}_z + \bar{B}_z}{3}, \quad (16)$$

Therefore, the color features CFz in the five CFAs of the Lycopersicon samples can be described as:

$$CF_z = (\bar{R}_z, \bar{G}_z, \bar{B}_z, H_z, S_z, V_z, \dot{H}_z, \dot{S}_z, I_z) \quad (17)$$

Finally, the nine color features are extracted and determined. The total number of color features is 45 because each CFA has nine extracted features. However, to recognize the Lycopersicon maturity stage, a supervised learning process is still necessary. The supervised learning structure proposed here comprises backpropagation neural networks. The input layer has 45 neurons; the hidden layer has 64 neurons; the output layer has 3 neurons. The ReLU is used as an activation function in the backpropagation neural networks except for the output layer. The activation function of the output layer is a Softmax function (Hinton and Salakhutdinov, 2009) that can be described as follows:

$$f_{\text{SoftMax}}(z)_j = \frac{e^{z_j}}{\sum_{k=1}^T e^{z_k}}, \text{ for } j = 1, \dots, k \quad (18)$$

where z is the output of a neural network without an activation function, T is the output size of the neural networks, and j is an integer number of the output set. The Softmax function calculates the probability rate of the determined maturity stage. Finally, the highest probability rate of the determined maturity stage is chosen as the final result of the neural networks.

3. Experimental results

3.1. Training

To train the two neural networks, the experimental dataset must be divided into a training dataset and a testing dataset. The maturity data is collected and averaged to prevent the neural network from overfitting the characteristics of a single maturity. An overview of the dataset is presented in Table 1. The data is collected from Lycopersicon farms. We collected a total of 450 pictures for each maturity level. The training data will consist of 400 pictures and 50 pictures will be the testing data. The 400 pictures of each level and 50 pictures of test data are increased to 2400 and 300, respectively, by flipping and mirroring. 2400 pictures of training data and 300 pictures of testing data are increased to 4800 and 600 by adding Gaussian noise. Therefore, there are 4800 training images for each maturity level, and 600 testing images for each maturity level.

To train the neural network, a loss function and an optimization must be chosen. The loss function in this study is the mean squared error (MSE) function. It can be described as follows:

$$L_{\text{mse}} = \frac{1}{n \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (19)$$

where \hat{y}_i is the predicted output, y_i the target value, and n is the sample number. The optimization in this study is performed by employing Adam (Kingma and Ba, 2015). The formula demonstrates a good convergence speed so that the neural network can learn faster. The respective equations are:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t,$$

$$n_t = \beta_2 n_{t-1} + (1 - \beta_2) g_t^2,$$

$$\theta_{\text{adam}, t+1} = \theta_{\text{adam}, t} - \frac{\eta}{\sqrt{n_t} + \varepsilon} \hat{m}_t, \quad (20)$$

where m_t is the mean of the gradient, n_t is the variance of the gradient, β_1 is a weight of m_t , β_2 is a weight of n_t , g_t is the gradient of $\theta_{\text{adam}, t}$, ε is a small positive number to avoid a division by 0, and η is the learning rate. Here, β_1 is 0.9 and β_2 is 0.9999.

Table 1
Data Overview.

Total experimental data	16,200
Training data	14,400
Immature training data	4,800
Semi-immature training data	4,800
Mature training data	4,800
Testing data	1,800
Immature testing data	600
Semi-immature testing data	600
Mature testing data	600

3.2. Background removal results

The proposed background removal system includes a CAE structure and a simple image processing procedure. The CAE tries to restore the labeled image with the input image. The MSE loss between the labeled image and the reverted image is reduced through optimization. The MSE loss score of the experimental dataset is presented in Table 2. The average MSE loss of the training dataset is smaller than the average MSE loss of the testing dataset. Although overfitting occurs, the effects of this phenomenon are negligible.

The CAE reverts the image but differences between the labeled image and the reverted image still exist. The SNR and PSNR are standards that express the degree of difference between images. Their values are presented in Table 3.

The SNRs and PSNRs demonstrate that the reverted image is not equal to the input image. However, the Lycopersicon edge can still be useful. After reverting the background-removing image, the edge of the Lycopersicon is clear. Using the algorithm mentioned in Section 2 B, the edge can be detected.

The test results of the simple background removal process are shown in Fig. 5. The proposed process in this study improves the process in (Wan et al., 2018). For a complex background, the algorithm can use a CAE to effectively filter out the target and provide clearer target images, thereby enhancing recognition accuracy.

The background removal system is suitable for special cases with multiple overlapping objects, a dark image, leaf blocking, and disturbances within the image. The special-case results are presented in Fig. 6. Evidently, the recognition system can process these special cases. The CAE can learn and mark the Lycopersicon in the image. Although it cannot learn or mark the Lycopersicon perfectly, the Lycopersicon edge in the reverted image resembles the Lycopersicon edge in the input image.

The comparison between the results from (Wan et al., 2018) and from this study is shown in Fig. 7. In Fig. 7, the result demonstrates that the method in (Wan et al., 2018) cannot remove the background. However, using the method in this study, the background can be removed. By changing the background removal process in (Wan et al., 2018), we can increase the generalization of the method. The method in

Table 3
Average SNRs and PSNRs between label image and reverted image.

Dataset category	SNR	PSNR
Training dataset	20.46	23.26
Immature training dataset	18.88	21.86
Semi-immature training dataset	22.95	25.24
Mature training dataset	19.51	22.69
Testing dataset	18.52	21.35
Immature testing dataset	16.11	19.12
Semi-immature testing dataset	21.61	23.91
Mature testing dataset	17.85	21.02

(Wan et al., 2018) can only be applied in the laboratory or to a straightforward image with a simple background. However, using our method, the application can be deployed even on Lycopersicon farms for monitoring the maturity of the Lycopersicon.

After proving that the CAE can remove complex background images, simple background images are also tested in this experiment. The result is shown as Fig. 8. In this simple case, the background is clear and there is only one target in the image. The conditions are similar to those in (Wan et al., 2018). However, even in the case in which the experimental conditions are similar, using the method in (Wan et al., 2018) does not produce satisfactory results. It can be observed that a small amount of light reflection will affect the experimental results of (Wan et al., 2018). However, our algorithm can still cope with the reflection effects.

We also test our algorithm with multiple targets as shown in Fig. 9. In the first image, a simple case of multiple objects is tested. The two targets in the image have the same maturity level. The edge between the Lycopersicon and the background is clearly visible. In the second image, there are two different maturity levels of Lycopersicon, and the result is stable. In the third image, the two targets overlap, and are connected. The result is still acceptable. In the fourth image, an image with targets that are far from each other is tested, and the result is still acceptable. In the fifth image, an image with three targets is tested, and the model appears to be capable of removing the background, but certain parts of the objects seem to be accidentally filtered out. Such experiments demonstrate that having more objects can be a burden to the model. However, the experimental solution also shows the CAE filtering the background, even when there are multiple objects in the picture.

3.3. Recognition results

The recognition system, i.e., the backpropagation system, decides the maturity stage of the Lycopersicon. By learning 14,400 Lycopersicon features, the accuracy rate achieved for the testing data is 100%.

To balance the competing needs of reduced computing time and higher rate of network training, the number of neurons in the classifier was set to 64 after multiple adjustments. Dividing the Lycopersicon into five feature regions is the most effective strategy to achieve this trade-off.

According to the results shown in Fig. 10 for different segmentation segments, too many or too few neurons lead to a low recognition system accuracy or long calculation time. The same result is reflected in the number of feature areas. In the training phase, even if only a few feature areas or neurons are available, an accuracy rate of 100% can be easily achieved. However, overfitting may occur during the testing phase. Too few neurons or features prevent the neural network from effectively identifying Lycopersicon maturity. Conversely, too many

Table 2
MSE loss of experimental dataset.

Training dataset	0.00602
Immature training dataset	0.00743
Semi-immature training dataset	0.00324
Mature training dataset	0.00638
Testing dataset	0.01042
Immature testing dataset	0.01580
Semi-immature testing dataset	0.00473
Mature testing dataset	0.01072

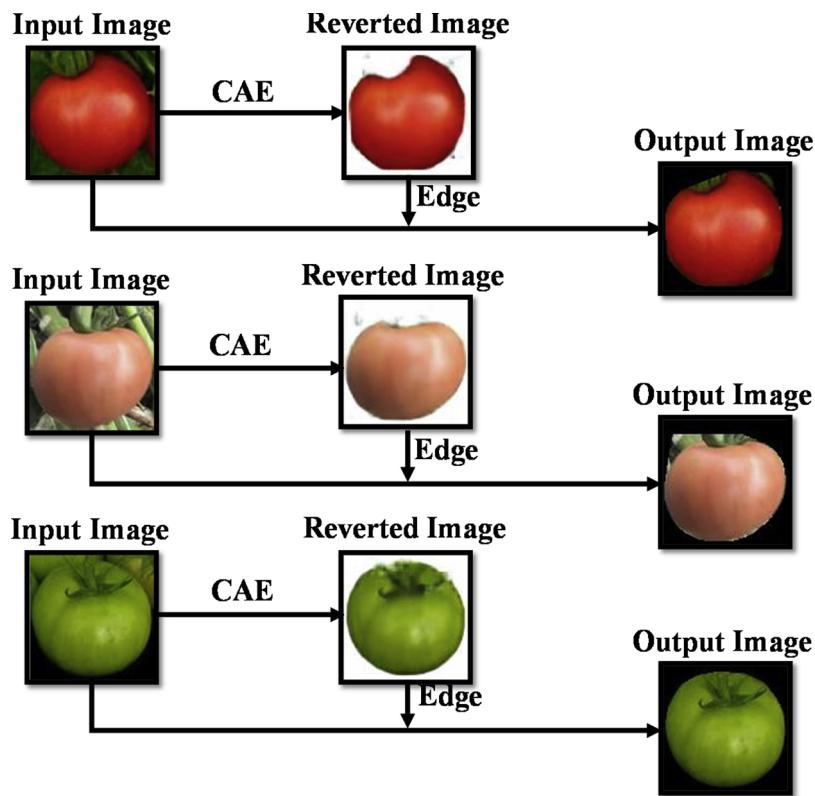


Fig. 5. Testing results of simple background removal process.

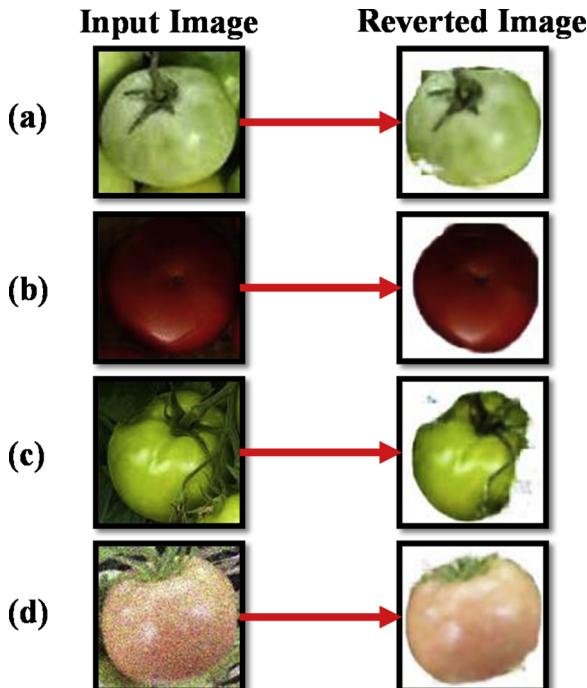


Fig. 6. Results of special cases: (a) multiple overlapping objects, (b) dark image, (c) leaf blocking, (d) disturbances within the image.

neurons or features cause the neural network to overfit. Therefore, moderate features and neuron quantities are essential for the success of this experiment.

The difference in feature extraction between the method in (Wan et al., 2018) and the method proposed in this study is that the method in (Wan et al., 2018) only uses the RGB and HSI color format. In this study, we employ the RGB, HSI, and HSV color formats. The increasing number of color features increases the accuracy rate. In (Wan et al., 2018), only two features (\bar{B}_i , \bar{H}_i) were used one at a time as inputs of the backpropagation neural networks. This study considers all nine color features as simultaneous inputs. In (Wan et al., 2018), only a single feature is used in each experiment. In (Wan et al., 2018), the author attempted to use all six features to find the best result. However, even the best feature, considered alone, could not score an accuracy rate of 100%. To ensure an accuracy rate of 100%, multiple features must be considered.

The experimental dataset collected from this study (16,200 images) is greater in size than that in (Wan et al., 2018). The effectiveness of the method can therefore be thoroughly evaluated.

To determine the most effective feature for the classifier, we introduce the random forest classifier (RFC) (Pal, 2005). After classifying 45 features with the RFC, the weight of each feature can be calculated. The results are presented in Table 4. It can be observed from Table 4 that the color characteristics such as hue saturation value (HSV), and RGB Green are important elements in Lycopersicon maturity feature recognition. Using the RFC, the accuracy rate of classification is 100%. Bagging (Liaw and Wiener, 2002) and Adaboost (Chan and Paelinckx, 2008) are both tested resampling methods for the RFC. The final results

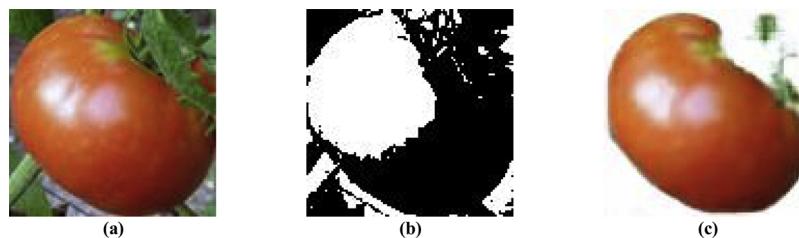


Fig. 7. Results of the complex case for background removal: (a) original complex case image, (b) result of using the method in (Wan et al., 2018), (c) result of using our method.

do not differ much in terms of successful recognition.

Further, a support vector machine (SVM) (Suykens and Vandewalle, 1999), a K nearest neighbor, and a linear regression are tested to ensure that feature extraction has the ability to extract useful features. The accuracy rates of the classifications are 100%.

However, redundant information still exists in the 45 features. To confirm this, a principal component analysis (PCA) (Price et al., 2006; Zhou et al., 2017; Shlens, 2014) is employed and the 45 features are reduced to eight features. According to the results, the accuracy rates of the classifications are still 100%. Therefore, redundant information exists among the 45 features and the PCA can effectively extract effective features. A comparison of the accuracy with and without PCA is shown in Fig. 11. The accuracy curve for the added PCA grows faster than that without PCA. Because the features are reduced, the parameters that the neural network needs to train are also correspondingly reduced. Further, the computational load on the system drops such that the classification accuracy is increased under equal neuron quantities.

The results confirm the effectiveness of our background removal algorithm and feature extraction algorithm. Therefore, these algorithms can be applied to other classification algorithms. In addition, by using PCA to reduce the dimension of the features, the computing resources can be reduced without reducing the accuracy rate.

To visualize the distribution of the extracted features, the highest and the second-highest variance ratio feature vector are shown in Fig. 12. In Fig. 12, the red points show the feature points of the mature Lycopersicon; the blue points show the feature points of the semi-immature Lycopersicon; the green points show the feature points of the immature Lycopersicon.

Fig. 12(a) shows the experimental result of using the training dataset. The features of the semi-immature Lycopersicon are distributed in the left half; the features of the immature Lycopersicon are distributed in the middle. The features of the mature Lycopersicon are distributed in the right half. It can be seen that the feature extraction plays a big role. The Euclidean distance between the features of different categories can be increased before classification. Fig. 12(b) shows the experimental results of the testing dataset. The result is similar to Fig. 12(a). This result shows that the training results are consistent with the test results. It also confirms the effectiveness of this method.

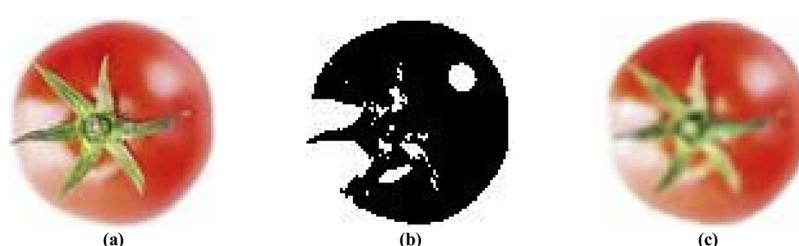


Fig. 8. Results of the simple case for background removal: (a) original simple case image, (b) the result of using the method in (Wan et al., 2018), (c) the result of using our method.

Input Image Reverted Image

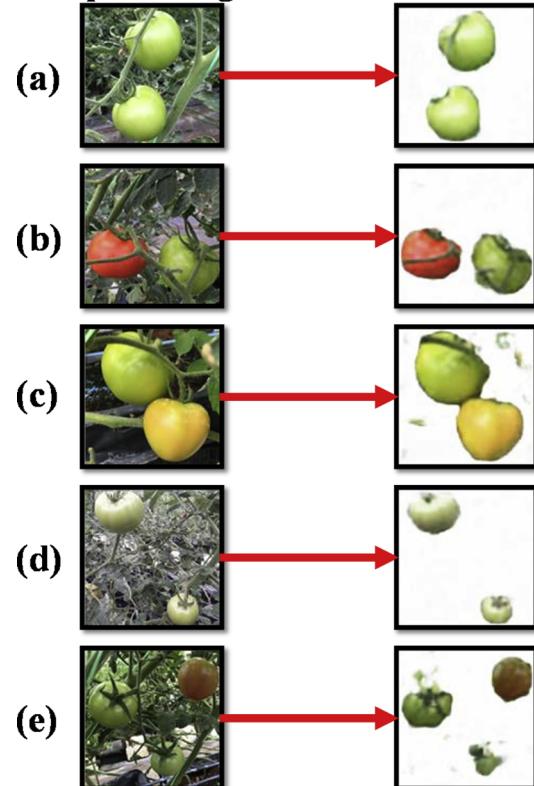


Fig. 9. Experimental results of multiple targets: (a) simple case, (b) different level targets, (c) overlapping targets, (d) far distance targets, (e) three targets.

3.4. Experiment with ImageNet

In this part, the Lycopersicon images on ImageNet (ZZZZZ, 2019) are used as the experimental dataset. Unfortunately, there are only a

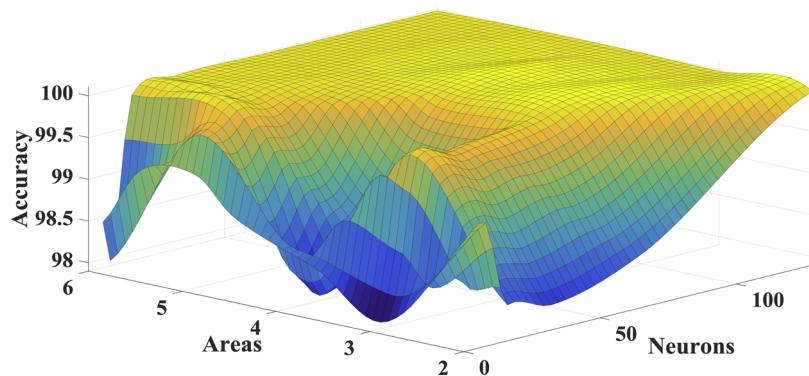


Fig. 10. 3D accuracy curve of the recognition system.

Table 4
Percentage of feature importance among 45 features.

Training 1		Training 2	
Feature	Importance	Feature	Importance
Area4_HSV(H)	19.01%	Area4_HSV(H)	14.09%
Area5_RGB(G)	13.89%	Area3_HSV(S)	9.92%
Area4_RGB(G)	13.57%	Area5_HSI(H)	9.91%
Area3_HSV(H)	10.10%	Area5_HSV(H)	9.77%
Area5_HSV(H)	9.34%	Area3_HSV(H)	9.39%
Area5_HSI(H)	5.32%	Area5_RGB(G)	6.25%
Area5_HSI(S)	5.14%	Area4_HSI(S)	5.20%
Area3_HSI(S)	4.70%	Area5_HSV(S)	4.97%
Area2_HSV(H)	4.49%	Area4_RGB(B)	4.58%
Area5_HSI(S)	4.40%	Area1_HSI(H)	4.35%
Area2_RGB(G)	4.04%	Area3_HSV(H)	4.32%
~Below 1%: ignored~		Area4_HSV(V)	3.99%
...		Area1_HSI(S)	3.84%
...		Area5_HSV(V)	3.29%
...		Area3_HSI(S)	1.21%
...		~Below 1%: ignored~	

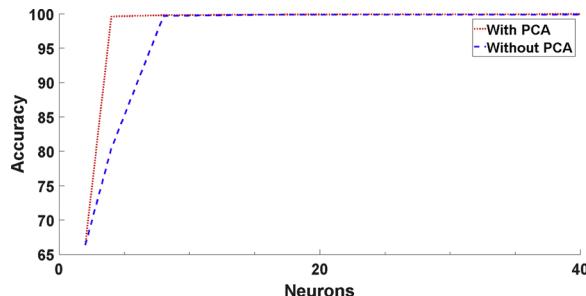


Fig. 11. Accuracy with and without PCA.

few semi-immature Lycopersicon images in ImageNet. Therefore, we use only two types of maturity—immature and mature, in this experiment.

300 Lycopersicon images are collected from ImageNet and labeled. There are 150 image data for each maturity level. At each maturity level, the data from 100 images will be increased to 600 images and added into the training dataset; the data from 50 images will be increased to 300 and added to the testing dataset. The testing result is shown as Fig. 13.

Fig. 13(a) shows the immature level Lycopersicon with a clean background; Fig. 13(b) shows the immature level Lycopersicon with a black background; Fig. 13(c) shows the mature level with a clean

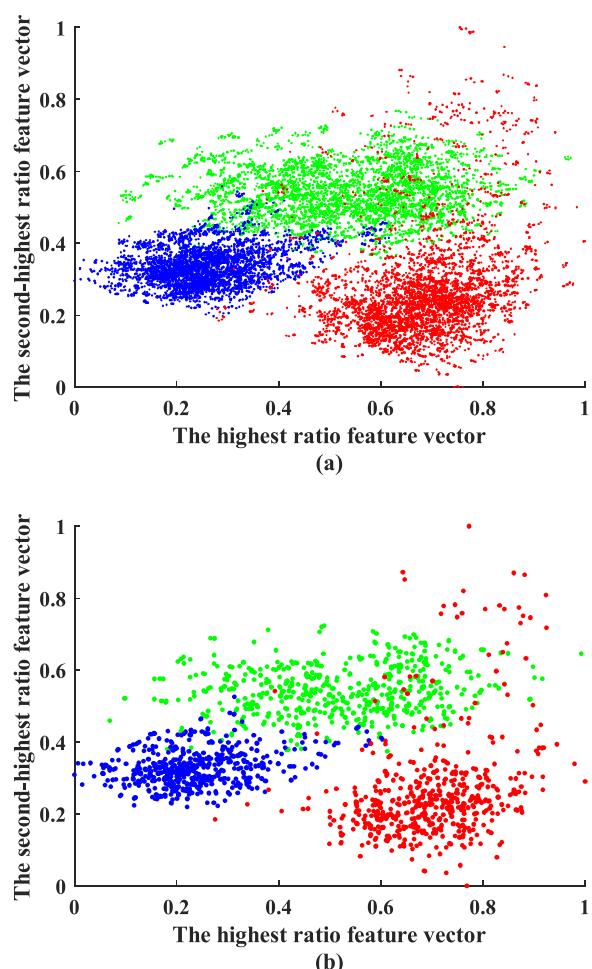


Fig. 12. The distribution of the extracted feature: (a) training results, (b) testing results.

background; Fig. 13(d) shows the mature level with multiple overlapping objects.

The experimental results show that CAE can also deal with the datasets of ImageNet. The backgrounds are removed from the images. Most of the Lycopersicon images on ImageNet are with a white background. With a white background, the performance of CAE is better than its performance with a complex background.

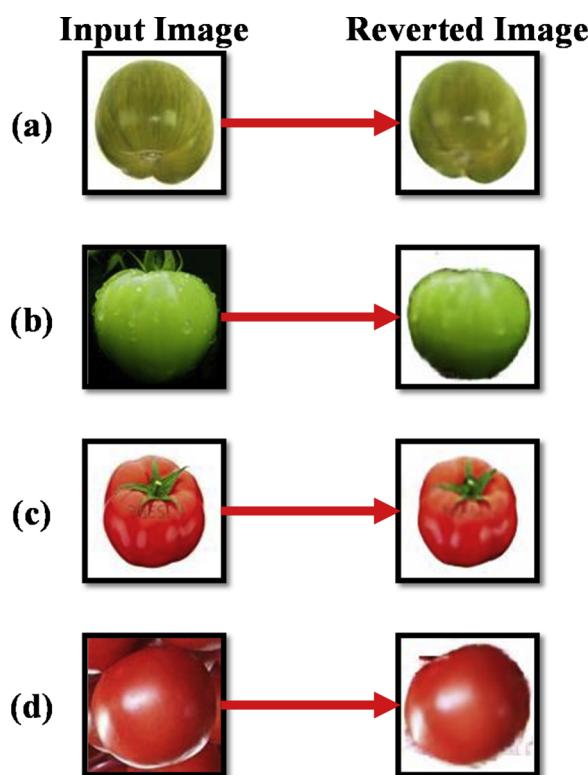


Fig. 13. Experimental results with ImageNet datasets: (a) immature level with a clean background, (b) immature level with a black background, (c) mature level with a clean background, (d) mature level with multiple overlapping objects.

After the removal of the background, the feature extraction and classification method are also tested with the data of ImageNet. The accuracy rate is as good as the result obtained with the dataset from our Lycopersicon farm, i.e., 100%.

Both image datasets, one without a background, and the original image with a background, are tested with a convolutional neural network (CNN) and SoftMax layer. The CNN consists of a convolutional kernel with a size of 2×2 , a fully connected layer with a neuron number of 2, and an output SoftMax layer with a neuron number of 2. We compared two kinds of data to show that background removal can help CNN identify objects more effectively. The accuracy results are shown in Table 5.

In Table 5, the score is the mean of the accuracy rates. The result shows that, with a CAE structure to remove the background, there is an increase in the identification capability of the CNN. Although this phenomenon is not very obvious, the results show that the complex background does reduce the recognition results of a CNN, and that the CAE can deal with this problem. Because the maturity level of the ImageNet dataset is of only two types (immature and mature), another score system can be considered, such as the receiver operating characteristic curve (Hanley and McNeil, 1982) (ROC curve). However, the ROC curves of the two experiments and the area under the curve (AUC) scores of the two experiments are the same. Both the AUC scores are 100%. Both the AUC scores are 100%.

Table 5

Comparison of the accuracy rate between the maturity level detection schemes with CAE and without CAE.

Algorithms	Training			Testing		
	Immature	Mature	Score	Immature	Mature	Score
CNN + SoftMax	99.67%	100%	99.84%	99.26%	100%	99.63%
CAE + CNN + SoftMax	100%	99.84%	99.92%	100%	100%	100%
CNN + SVM	100%	100%	100%	100%	99.81%	99.91%
CAE + CNN + SVM	100%	100%	100%	100%	100%	100%

In this portion of our study, we tried a simple CNN to confirm that the background removal by CAE is useful with other algorithms. Of course, with a more complex structure of CNN, the accuracy rates can all become 100%. But such a scheme makes the comparison meaningless. Under the simple identification algorithm, we wish to identify the groups of data that can show better features.

The experiment also tried to use SVM as a classification method. The result also shows the same phenomenon whereby background removal can help object recognition to be more useful. Unlike SoftMax, SVM can easily get the best solution through the features extracted by CNN.

3.5. Computer system

Experiments for background removal of the required features are conducted at a Lycopersicon farm. The camera used is a Logitech C920 r. As far as possible, additional light sources are not used to complement the natural light environment on the farm.

The learning epoch is 100,000 iterations. However, the experiment never runs 100,000 times. When training neural networks, there is a mechanism to stop early. In our case, the loss value of the training data is monitored. If the loss value stops updating for 100 epochs, the process will be interrupted and the network with the lowest loss value will be saved. In our case, the batch size is 100. If the batch size is too large, the computer system may crash.

The image-processing program was developed on a desktop PC with Intel i7, 32 GB RAM and NVIDIA GeForce GTX TITAN X. The CAE training requires 30.28 s per epoch and the training of the back-propagation neural networks requires 0.1 s per epoch. The forward CAE takes 0.12 s per image and the forward backpropagation neural networks take 0.011 s per image. Overall, the calculation speed of the forward step is still acceptable, and the system can be operated in real time.

4. Conclusion

In this study, a background removal system and a recognition system were proposed to recognize the maturity of Lycopersicons. The Lycopersicon in an image is filtered out via a CAE. After filtering, the Lycopersicon is detected in the image with an edge mask. Nine features of each CFA are then extracted with concentric circles and the RGB, HSI, and HSV color formats. The 45 features obtained are used as inputs to the backpropagation neural networks. After the Softmax layer has calculated the recognition possibility of the three maturity levels, the recognition procedure is completed.

Through the transpose convolutional neural network, CAE has the ability to revert an image. Previously, the reverted image was often an input image. In this work, the reverted image is a background-removed image. The label in this work is not the input image, but an image whose background has been artificially removed. We discovered a new way to use CAE and found that CAE has the ability to remove the background of a Lycopersicon image.

A CAE structure removes the background to pre-process the image. This improvement increases the versatility of this method. The experiments in (Wan et al., 2018) can only be used in the laboratory or in certain simple applications. However, after the change proposed in this

study, the experiment can be executed on a Lycopersicon farm. In addition, the consideration of all extracted features reduces the hand-crafted and manual inputs to the system. In (Wan et al., 2018), the author considered the features one at a time and chose the feature with the largest variance. In this study, all features are considered simultaneously. With such changes in the algorithm, the feature with the largest variance will not be required.

The designed system can be applied on an automated Lycopersicon farm. Determining maturity is of key importance for the harvest. Lycopersicons meant for sales to nearby locations should be harvested during their semi-immature stage; Lycopersicons for long-distance transportation should be harvested during their immature stage. For automatic harvests, the machine must detect Lycopersicons and recognize their maturity stages.

In the future, we plan to improve the complete system to realize an end-to-end neural network. We will focus on minimizing manual labor as much as possible. A harvesting robot and recognition system can improve the automatic harvesting program for Lycopersicons.

Funding

This study was funded by a grant from the Ministry of Science and Technology, Taiwan, R.O.C. under Grant no. MOST 108-2218-E-110-005.

References

- Rick, C.M., Butler, L., 1956. Cytogenetics of the tomato. In: In: Demerec, M. (Ed.), Advances in Genetics, vol. 8. Academic Press, pp. 267–382. [https://doi.org/10.1016/S0065-2660\(08\)60504-0](https://doi.org/10.1016/S0065-2660(08)60504-0).
- Effendi, Z., Ramli, R., Ghani, J.A., 2011. Application of back propagation diagnostic model for fruit maturity classification: case *Jatropha curcas*. *Aust. J. Basic Appl. Sci.* 5 (March (3)), 134–140.
- Alonso-Salces, R.M., Herrero, C., Barranco, A., Berrueta, L.A., Gallo, B., Vicente, F., 2005. Classification of apple fruits according to their maturity state by the pattern recognition analysis of their polyphenolic compositions. *Food Chem.* 1, 113–123. <https://doi.org/10.1016/j.foodchem.2004.10.013>. 93: (November).
- Asnor, J.I., Rosmah, S., Wan, Z.W.H., Badrul, H.A.B., 2013. Pineapple maturity recognition using RGB extraction. *Int. J. Electr. Comput. Eng.* 7 (June (6)), 597–600. <https://doi.org/10.5281/zenodo.1077855>.
- Rodríguez-Pulido, F.J., Ferrer-Gallego, R., Lourdes González-Miret, M., Rivas-Gonzalo, J.C., Escrivano-Bailón, M.T., Heredia, F.J., 2012. Preliminary study to determine the phenolic maturity stage of grape seeds by computer vision. *Anal. Chim. Acta* 732 (June), 78–82. <https://doi.org/10.1016/j.acaca.2012.01.005>.
- Zufiiga, A., Mora, M., Oyarce, M., Fredes, C., 2014. Grape maturity estimation based on seed images and neural networks. *Eng. Appl. Artif. Intell.* 35 (October), 95–104. <https://doi.org/10.1016/j.engappai.2014.06.007>.
- Hahn, F., 1999. Neural networks predict tomato maturity stage.". *AeroSense '99* 3722 (March (6)). <https://doi.org/10.1111/12.342913>.
- Wang, X., Mao, H., Han, X., Yin, J., 2011. Vision-based judgment of tomato maturity under growth conditions. *Afr. J. Biotechnol.* 10 (May (18)), 3616–3623. <https://doi.org/10.5897/AJB11.015>.
- Xiao, Q., Niu, W., Zhang, H., 2015. Predicting fruit maturity stage dynamically based on fuzzy recognition and color feature. 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS) 944–948. <https://doi.org/10.1109/ICSESS.2015.7339210>.
- Wan, P., Toudeshki, A., Tan, H., Ehsani, R., 2018. A methodology for fresh tomato maturity detection using computer vision. *Comput. Electron. Agric.* 146 (March), 43–50. <https://doi.org/10.1016/j.compag.2018.01.011>.
- Zhang, L., McCarthy, M.J., 2012. Measurement and evaluation of tomato maturity using magnetic resonance imaging. *Postharvest Biol. Technol.* 67 (May), 37–43. <https://doi.org/10.1016/j.postharvbio.2011.12.004>.
- Gómez, A.H., Hu, G., Wang, J., Pereira, A.G., 2006. Evaluation of tomato maturity by electronic nose. *Comput. Electron. Agric.* 54 (October (1)), 44–52. <https://doi.org/10.1016/j.compag.2006.07.002>.
- Clément, A., Dorais, M., Vernon, M., 2008. Multivariate approach to the measurement of tomato maturity and gustatory attributes and their rapid assessment by vis–nir spectroscopy. *J. Agric. Food. Chem.* 56 (March (5)), 1538–1544. <https://doi.org/10.1021/jf072182n>.
- Lien, C.-C., Ay, C., Ting, C.-H., 2009. Non-destructive impact test for assessment of tomato maturity. *J. Food Eng.* 91 (April (3)), 402–407. <https://doi.org/10.1016/j.jfoodeng.2008.09.036>.
- Leng, B., Guo, S., Zhang, X., Xiong, Z., 2015. 3D object retrieval with stacked local convolutional autoencoder. *Signal Process.* 112 (July), 119–128. <https://doi.org/10.1016/j.sigpro.2014.09.005>.
- Holden, D., Saito, J., Komura, T., Joyce, T., 2015. Learning motion manifolds with convolutional autoencoders. Proceeding SA' 15 SIGGRAPH Asia 2015 Technical Briefs, Kobe, Japan, No. 18. <https://doi.org/10.1145/2820903.2820918>.
- Chen, M., Shi, X., Zhang, Y., Wu, D., Guizani, M., 2017. Deep features learning for medical image analysis with convolutional autoencoder neural network. *IEEE Transactions on Big Data*(June). <https://doi.org/10.1109/TBDA.2017.2717439>.
- Makhzani, A., Frey, B., 2015. Winner-take-all autoencoders. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*. Palais des Congrès de Montréal, Montréal CANADA December.
- Masci, J., Meier, U., Cireşan, D., Schmidhuber, J., 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. *International Conference on Artificial Neural Networks* 52–59.
- Deng, L., Seltzer, M.L., Yu, D., Acero, A., Mohamed, A.-r., Hinton, G., 2010. Binary coding of speech spectrograms using a deep auto-encoder. *INTERSPEECH-2010* (September), 1692–1695.
- LeCun, Y., et al., 1990. Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* 2, 396–404.
- Scherer, D., Müller, A., Behnke, S., 2010. Evaluation of pooling operations in convolutional architectures for object recognition.". *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III* 92–101 September.
- Radford, A., Metz, L., Chintala, S., 2015. Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434*, November .
- Dumoulin, V., Visin, F., 2016. A Guide to Convolution Arithmetic for Deep Learning. *arXiv:1603.07285*, March .
- Ioffe, S., Szegedy, C., 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167*, February .
- Nair, V., Hinton, G.E., 1995. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on Machine Learning* 807–814.
- Han, J., Moraga, C., 2005. The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning. Berlin, Heidelberg, Torremolinos, Spain, June. pp. 195–201.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* (June), 1929–1958.
- Canny, J., 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8* (November (6)), 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>.
- Barber, C.B., Dobkin, D.P., Huhdanpaa, H., 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.* 22 (December (4)), 469–483. <https://doi.org/10.1145/235815.235821>.
- Helson, H., 1938. Fundamental problems in color vision. I. The principle governing changes in hue, saturation, and lightness of non-selective samples in chromatic illumination. *J. Exp. Psychol.* 23 (Nov ember (5)), 439–476. <https://doi.org/10.1037/h0060971>.
- Smith, A.R., 1978. Color gamut transform pairs. *SIGGRAPH Comput. Graph.* 12 (August (3)), 12–19. <https://doi.org/10.1145/965139.807361>.
- Hinton, G.E., Salakhutdinov, R.R., 2009. Replicated softmax: an undirected topic model. *Adv. Neural Inf. Process. Syst.* 22 (December).
- Kingma, D.P., Ba, J., 2015. Adam: a method for stochastic optimization. *International Conference on Learning Representations*.
- Pal, M., 2005. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* 26 (January (1)), 217–222. <https://doi.org/10.1080/01431160412331269698>.
- Liaw, A., Wiener, M., 2002. Classification and regression by randomforest. *R News* 2 (December (3)), 18–22.
- Chan, J.C.-W., Paelinckx, D., 2008. Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sens. Environ.* 112 (June (6)), 2999–3011. <https://doi.org/10.1016/j.rse.2008.02.011>.
- Suykens, J.A.K., Vandewalle, J., 1999. Least squares support vector machine classifiers. *Neural Process. Lett.* 9 (June (3)), 293–300. <https://doi.org/10.1023/a:1018628609742>.
- Price, A.L., Patterson, N.J., Plenge, R.M., Weinblatt, M.E., Shadick, N.A., Reich, D., 2006. Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet.* 38 (July (8)), 904–909. <https://doi.org/10.1038/ng1847>.
- Zhou, W., Yeh, C., Jin, R., Li, Z., Song, S., Yang, J., 2017. ISAR imaging of targets with rotating parts based on robust principal component analysis. *IET Radar, Sonar Navig.* 11 (April (4)), 563–569. <https://doi.org/10.1049/iet-rsn.2015.0647>.
- Shlens, J., 2014. A Tutorial on Principal Component Analysis. *arXiv:1404.1100*, April .
- Image-net.org, 2018, ImageNet. [online] Available at: <http://www.image-net.org/>.
- Hanley, J.A., McNeil, B.J., 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143 (April (1)), 29–36. <https://doi.org/10.1148/radiology.143.1.7063747>.