Hybrid Neural Network and Linear Model for Natural Produce Recognition Using Computer Vision

Joko Siswantoro[1,*], Anton Satria Prabuwono[2,3], Azizi Abdullah[2] & Bahari Indrus[2]

[1]Departement of Informatics Engineering, Faculty of Engineering,
Universitas Surabaya, Jalan Raya Kali Rungkut, Surabaya, 60293, Indonesia
[2]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia,
Bangi, 43600 UKM, Selangor D.E., Malaysia
[3]Faculty of Computing and Information Technology, King Abdulaziz University,
Rabigh 21911, Saudi Arabia
*E-mail: joko_siswantoro@staff.ubaya.ac.id

Abstract. Natural produce recognition is a classification problem with various applications in the food industry. This paper proposes a natural produce recognition method using computer vision. The proposed method uses simple features consisting of statistical color features and the derivative of radius function. A hybrid neural network and linear model based on a Kalman filter (NN-LMKF) was employed as classifier. One thousand images from ten categories of natural produce were used to validate the proposed method by using 5-fold cross validation. The experimental result showed that the proposed method achieved classification accuracy of 98.40%. This means it performed better than the original neural network and *k*-nearest neighborhood.

Keywords: *Kalman filter; linear model; natural produce; neural network; recognition.*

1 Introduction

Natural produce recognition using computer vision has many applications in the food industry, including for pricing [1], sorting [2-4], grading [5], eating guidance [6], and measurement [7]. To recognize produce, the computer vision system first acquires an image of the produce. The image is then processed to increase its quality. Features of produce such as color, shape and texture are then extracted from the processed image and used to recognize the produce [8].

Bolle, *et al.*[1] have proposed a computer vision system to recognize natural produce automatically called VeggieVision. The system was designed to replace barcodes for pricing produce at supermarkets and grocery stores. However, the system has low accuracy. Roomi, *et al.* [2] have proposed a method for intra classification of mangos using a computer vision system. Shape features, including object contour, region-based descriptors and boundary-based descriptors, were extracted from the images of mangos. Although the proposed method achieved classification accuracy of 90.91% using a Bayesian classifier, it was only validated using three categories of mango. Waltner, *et al.* [6] have

Received August 8th, 2016, Revised January 30th, 2017, Accepted for publication May 23rd, 2017.
 ISSN: 2337-5779, DOI: 10.5614/ itbj.ict.res.appl.2017.11.2.5

proposed a mobile augmented reality application for fruit and vegetable recognition with eating guidance and food awareness functionality. They extracted the discriminative color descriptor and color histogram in RGB, HSV, and LAB color space from acquired images. Random forest was used for classification and achieved classification accuracy of 80.30%.

Several other researchers have proposed methods to recognize natural produce using computer vision by employing a combination of long features and complex classifiers in order to achieve high recognition performance [9-11]. However, more time is needed to extract the long features and to train the complex classifiers in these proposed methods. Other researchers have applied principle component analysis (PCA) to reduce the dimensions of the extracted features [4, 12-14]. The reduced features were then used to classify natural produce using a feed forward neural network trained by several methods. Although PCA can reduce feature dimensions by more than 50%, the use of PCA also increases training time. Prabuwono, *et al.* [8] have proposed a natural produce classification method using computer vision. They used 16 features consisting of 12 statistical color features in HSV color space and 4 shape features from the derivative of radius function and a back propagation neural network classifier. Although the proposed method achieved good performance, it was only validated using three classes of produce.

Neural network is a promising classifier that has been applied in many fields. The structure of the neural network is an important factor for classification performance. The complexity of this structure depends on the number of input features and the number of output classes. If a neural network classifier uses an inappropriate structure then it tends to be a weak classifier [15]. Several methods have been proposed to increase classification performance of neural networks. One of them is by using a linear model based on a Kalman filter [15]. The model was used to adjust the predicated output of a neural network such that the classification performance was increased.

This paper proposes a method for natural produce classification by employing a combination of neural network and linear model based on a Kalman filter as classifier. Statistical color features and the derivative of radius function were used as features for classification. The rest of this paper is arranged as follows. Section 2 describes the proposed method and materials used in the experiment. Section 3 explains the validation of the proposed method. Section 4 presents the result of the experiment and its discussion. The conclusion is provided in Section 5.

2 Material and Method

The proposed method was implemented on a computer vision system consisting of hardware and software. The hardware comprised a camera, a light source, and a personal computer. The system used a Logitech HD Webcam C270h to capture the image of the produce. Room lighting, consisting of fluorescent lamps located on the ceiling of the room, was used as light source. The camera was connected to a 3.00 GHz Pentium (R) Dual-Core portable computer with 2 GB RAM and 32-bit Windows 7 operating system using a USB cable. The computer was equipped with software for controlling the camera, image processing, feature extraction, and classification. The software was developed using Visual C++ 2010 with open-source computer vision library OpenCV 231 [16]. The following subsections describe the steps of the proposed method.

2.1 Image Acquisition

During image acquisition, the produce was located about 40 cm below the camera. The images of produce were acquired one by one against a black background and saved in RGB (Red Green Blue) format. The images had dimensions of 640 × 480 pixels and a resolution 96 dpi both in the vertical and the horizontal direction. Figures 1 and 2 show the position of the produce during image acquisition and samples of the acquired images, respectively.

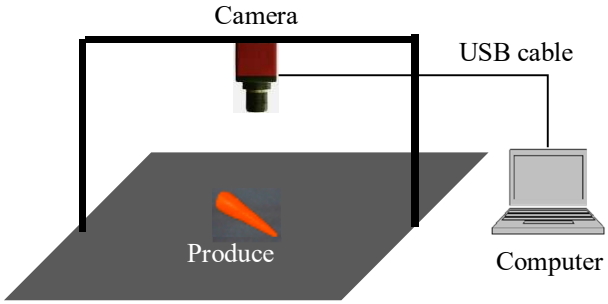

Figure 1 Position of produce during image acquisition.

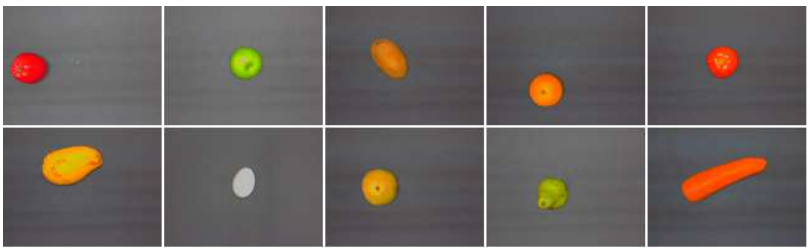

Figure 2 Samples of acquired images.

2.2 Pre-processing

Pre-processing is a step to improve the quality of the acquired image in order to facilitate the next steps. The result of this step is a gray scale image that will be used in the segmentation step. In this step, the image was first transformed from RGB color space to HSV (Hue Saturation Value) color space. Figure 3 (a), (b), (c), and (d) show a sample of an acquired image, the image in the H channel, the image in the S channel, and the image in the V channel, respectively.

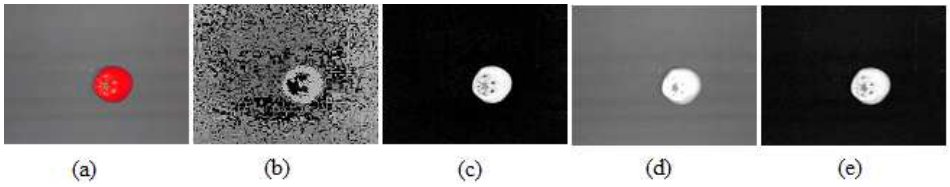

Figure 3 (a) Acquired image, (b) image in H channel, (c) image in S channel, (d) image in V channel, (e) gray scale image.

As can be seen in Figures 3(c) and 3(d), the produce can be easily separated from its background in the S and V channels. Therefore, only images in the S and V channels were used to produce the gray scale image. The gray scale image was constructed using the average of the image in the S and V channels. To reduce noise in the gray scale image, a 5 × 5 Gaussian filter was then applied. The resulted gray scale image is shown in Figure 3(e).

2.3 Segmentation

Segmentation is the step in which the produce is separated from its background. The result of this step is a binary image consisting of white pixels with binary value 1 for the produce and black pixels with binary value 0 for the background. The binary image is then used as mask to decompose the acquired image into two parts: produce and background.

The binary image is constructed from the gray scale image using thresholding. The threshold value, T, was determined automatically using a simple iteration as described in Gonzalez and Woods [17]. The steps of the iteration are as follows:

1. Determine the initial value of T (T_0).
2. Segment the gray scale image using T_0 to produce two mutually exclusive sets of pixels in the gray scale image: the set of background pixels (S_1) and the set of object pixels (S_2).
3. Calculate the mean of S_1 and S_2 to produce m_1 and m_2, respectively.
4. Calculate the threshold value $T = (m_1 + m_2)/2$.
5. Calculate $\Delta T = |T - T_0|$

6. Set $T_0 = T$
7. Repeat steps 2 through 6 until $\Delta T < 0.5$

A pixel in the gray scale image was classified as a produce pixel if its gray scale value was greater than *T*, otherwise it was classified as a background pixel. Sometimes, after thresholding there are a few misclassified pixels. To overcome this problem, morphological opening and closing operators with an ellipse structural element were used to remove white spots in the background pixels and black spots in the produce pixels, respectively. The result of the segmentation step is shown in Figure 4.

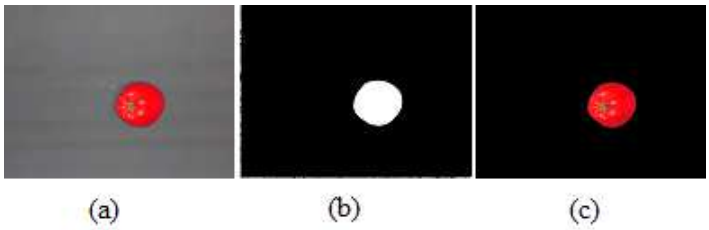

Figure 4 (a) Acquired image, (b) binary image, and (c) segmented image.

2.4 Feature Extraction

This step aims to extract the features of the produce that will be used in the classification step. In this study, two features were extracted from the segmented image, i.e. color and shape.

2.4.1 Color Features

According to Zheng and Sun [18], color features extracted from HSV color space have better performance than color features extracted from RGB color space in food product classification. Therefore, color features consisting of the statistical color features of produce pixels in HSV color space were used in this study. The features represent the distribution of produce color in the H, V, S channels.

Four statistical color features were extracted from the intensity values of the produce pixels in each channel, including mean, standard deviation, skewness, and kurtosis. In total, 12 color features were extracted from the images. Mean, standard deviation, skewness, and kurtosis were calculated using Eqs. (1), (2), (3) and (4), respectively.

$$\bar{f} = \frac{1}{N}\sum_{i=1}^{N} f_i \tag{1}$$

$$S = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}\left(f_i - \bar{f}\right)^2\right)} \tag{2}$$

$$Skewness = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(f_i - \bar{f}\right)^3}{S^3} \tag{3}$$

$$Kurtosis = \frac{\frac{1}{N}\sum_{i=1}^{N}\left(f_i - \bar{f}\right)^4}{S^4} \tag{4}$$

where, N is the number of produce pixels, f_i is the intensity value of the i^{th} pixel in the H, S, or V channel, for $i = 1, 2, \ldots, N$.

2.4.2 Shape Features

The derivative of radius function, $dr/d\theta$ was used as shape feature. A radius function, $r = f(\theta)$, is defined as the distance from the center of an object to its boundary in direction θ, where θ is the angle between a horizontal line through the center of the object and a line connecting the center of the object with its boundary, measured counter clockwise [8]. Figure 5 illustrates the value of radius function $r = f(\theta)$ in direction θ. Before calculating the value of radius function for a certain θ, the center of the object (x_c, y_c) needs to be determined. The center of the object was determined from the binary image of the produce using the following Eqs. (5) and (6) [16]:

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}} \tag{5}$$

where m_{pq} is the (p,q) moment of the binary image calculated using the following equation:

$$m_{pq} = \sum_{y=0}^{H-1}\sum_{x=0}^{W-1} I(x,y)x^p y^q \tag{6}$$

where, H and W are height and width of the image, respectively; $I(x,y)$ is the value of the binary image in (x,y); p and q are the x-order and y-order of moment, respectively.

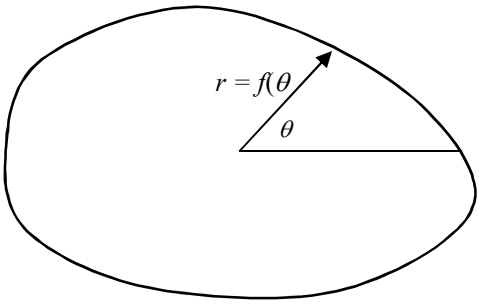

Figure 5 Value of radius function $r = f(\theta)$ in direction θ.

The value of the radius function in direction θ was calculated by finding r in $[0, r_{\max}]$ such that the point (x_b, y_b) is located on the boundary of the produce. The values of x_b, y_b and $r_{\max}$ are calculated using the following equations:

$$x_b = x_c + r\cos\theta \tag{7}$$

$$y_b = y_c + r\sin\theta \tag{8}$$

$$r_{\max} = \frac{1}{2}\sqrt{l^2 + w^2} \tag{9}$$

where l and w are the length and width of the minimum bounding rectangle of the object, respectively.

The radius function is sensitive to the size and orientation of the object. Two objects with the same shape and orientation can have different radius functions, if these objects have different sizes. However, the difference between these radius functions is almost constant for every θ. Therefore, these functions have the same derivative for every θ. This shows that the derivative of radius function is robust to size differences. Thirty-six values of the radius function were measured from the object for $0 \le \theta \le 2\pi$ with $\Delta\theta = 2\pi/36$ by using forward difference, as in the following equation:

$$\frac{dr}{d\theta} = \frac{1}{\Delta\theta}\left(f(\theta + \Delta\theta) - f(\theta)\right) \tag{10}$$

All values of the radius function were then summarized by calculating mean, standard deviation, skewness, and kurtosis of these values using Eqs. (1), (2), (3) and (4), respectively. This summary will ensure that the shape features are robust to orientation differences. Therefore, four shape features were extracted from the image, so in total 16 features were extracted from the image.

2.5 Classification

The aim of this step is to recognize a produce based on the features extracted from the image of the produce. A hybrid neural network (NN) and linear model based on a Kalman filter (LMKF), called NN-LMKF [15], was used to classify the produce. The structure of the NN used in this study consisted of an input layer containing 16 neurons that correspond to the object features, a hidden layer, and an output layer containing 10 neurons that correspond to the object class. The number of neurons in the hidden layer was determined empirically from 10 to 16 neurons such that the best classification accuracy was achieved. A sigmoid function was used as transfer function both from the input layer to the hidden layer and from the hidden layer to the output layer. The NN was then trained using back propagation with momentum.

LMKF was used for post-processing of the predicted output of the NN to improve classification accuracy. The predicted output of the NN and the object features were used as the input variables for the linear model, while the output of the linear model was the object class. The model was constructed using a linear combination of the object features to adjust the predicted output of the NN. The construction of the model is shown by the following equation:

$$\mathbf{z} = \mathbf{A}\tilde{\mathbf{z}} + \mathbf{B}\mathbf{f} + \mathbf{v} \tag{11}$$

where, $\mathbf{z} = [z_1, z_2, \ldots, z_{10}]^T$ is object class, $\tilde{\mathbf{z}} = [\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_{10}]^T$ is the predicted output of the NN, $\mathbf{f} = [f_1, f_2, \ldots, f_{16}]^T$ is object feature, $\mathbf{A}$ is a 10×10 diagonal matrix as in Eq. (1), $\mathbf{B}$ is a 10×16 matrix as in Eq. (2). $\mathbf{A}$ and $\mathbf{B}$ are unknown parameters of the model, and $\mathbf{v}$ is the error term assumed normally distributed with mean $\mathbf{0}$ and covariance matrix $\mathbf{R}$. The values of $\mathbf{A}$ and $\mathbf{B}$ were then estimated using Kalman filter iteration. Details of LMKF parameter estimation can be found in [15]. All features were normalized into interval $[-1,1]$ before being used in the training of the NN and in the parameter estimation of LMKF in order to minimize bias and reduce training time [19].

$$\mathbf{A} = diag[a_{11}, a_{22}, \ldots, a_{1010}] \tag{1}$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{116} \\ b_{21} & b_{22} & \cdots & b_{216} \\ \vdots & \vdots & \ddots & \vdots \\ b_{101} & b_{102} & \cdots & b_{1016} \end{bmatrix} \tag{2}$$

3 Validation

Ten categories of natural produce were used to validate the proposed method. The produce consisted of red delicious apple, green apple, potato, orange, tomato, mango, egg, pear Ya, pear Peckham, and carrot. Each category comprised 20 different objects varying in size, shape, and color. Images of each object were captured five times at different positions and orientations. Therefore, each category contained 100 images and in total there were 1000 images. Figure 6 shows the samples of the images used in validation.

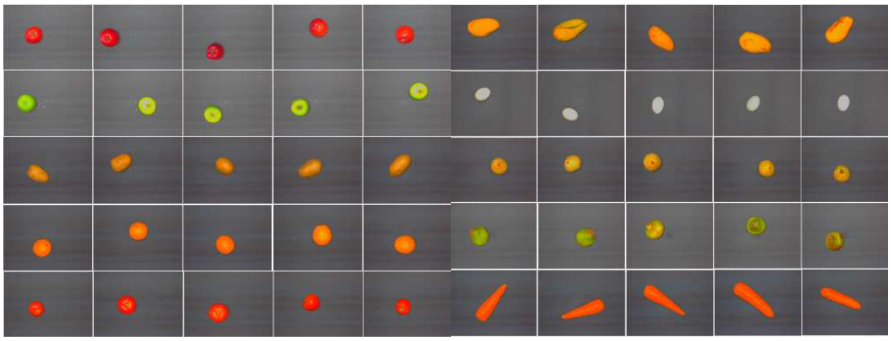

Figure 6 Samples of images used in validation.

Five-fold cross validation [20] was used to construct the training dataset and the testing dataset. The entire sample was randomly partitioned into five subsamples of the same size. Four subsamples were used as the training dataset and the remaining subsamples were used as the testing dataset. The training and testing processes were repeated five times, such that each subsample was used as testing dataset once. In 5-fold cross validation, the portioning process can be performed purely randomly. However, some subsamples may have a different class distribution. Therefore, in this study, stratified 5-fold cross validation was used to ensure that each subsample had the same class distribution [21]. The classification accuracy of both NN and NN-LMKF were calculated in classifying all five testing datasets to measure the performance of the proposed method.

4 Result and Discussion

The classification accuracy results for NN and NN-LMKF with different numbers of neurons in the hidden layer are summarized in Table 1. From Table 1, it can be observed that on average, the classification accuracy of NN-LMKF was better than the classification accuracy of NN for all numbers of neurons in the hidden layer, with increasing classification accuracy ranging from 9.00% to

43.10%. The best classification accuracies for NN and NN-LMKF were 89.40% and 98.40%, respectively, and were achieved using 12 neurons in the hidden layer. This indicates that combining NN with LMKF can improve the classification of the original NN. Furthermore, by combining NN with LMKF, the standard deviation of classification accuracy was also reduced by between 1.26% and 12.47%. This result shows that the classification accuracy of NN-LMKF had smaller variation compared to the classification accuracy of NN.

Table 1 Summary of classification accuracy for NN and NN-LMKF.

Number of neurons in the hidden layer	Mean and std. dev. of classification accuracy (%)		Mean and std. dev. of increasing classification accuracy (%)
	NN	MN-LMKF	
10	75.70 ± 14.86	97.20 ± 2.39	21.50 ± 14.44
11	78.60 ± 6.88	97.00 ± 3.00	18.40 ± 4.29
12	89.40 ± 8.19	98.40 ± 1.14	9.00 ± 8.31
13	70.40 ± 11.73	96.70 ± 2.93	26.30 ± 10.07
14	68.90 ± 13.55	97.00 ± 1.12	28.10 ± 14.61
15	54.30 ± 8.76	97.63 ± 1.11	43.10 ± 8.34
16	63.20 ± 2.99	97.50 ± 1.73	34.30 ± 2.31

The classification accuracy of NN-LMKF with 12 neurons in the hidden layer for each category can be seen in Figure 7. Figure 7 contains the confusion matrix of NN-LMKF with 12 neurons in the hidden layer. A confusion matrix contains information about the actual class label and the class label output from the classifier. It is used to describe the performance of the classification of a classifier. The elements in the i^{th} row and the j^{th} column represent the percentages of the samples in the j^{th} class that were classified into the i^{th} class. The elements on the diagonal represent the percentages of correctly classified samples. The higher the value of the diagonal elements, the better the classification performance achieved.

As can be seen from the diagonal of the confusion matrix in Figure 7, all samples in the 7th class (egg) and the 10th class (carrot) were correctly classified. For the other categories, there were only a few misclassified samples. 1% of the samples in the first class (apple red delicious) were classified into the 5th class (tomato), 1% of the samples in the 2nd class (green apple) were classified into the 8th class (pear Ya), 1% of the samples in the 3rd class (potatoes) were classified into the 8th class (pear Ya), 1% of the samples in the 4th class (orange) were classified into the 6th class (mango Chukanan), and 1% of the sample in the 9th class (pear Packham) were classified into the 3rd class (potato). In the 8th class (pear Ya), 1% of the samples were classified into the 3rd class (potato) and 1% of the samples were classified into the 9th class (pear Packham). In the 5th class (tomato), 2% of the samples were classified into the 4th class (orange) and 1% of the samples were classified into the 6th class (mango Chukanan).

Misclassification was highest in the 6th class (mango Chukanan): 6% of the samples in this class were classified into the 4th class (orange).

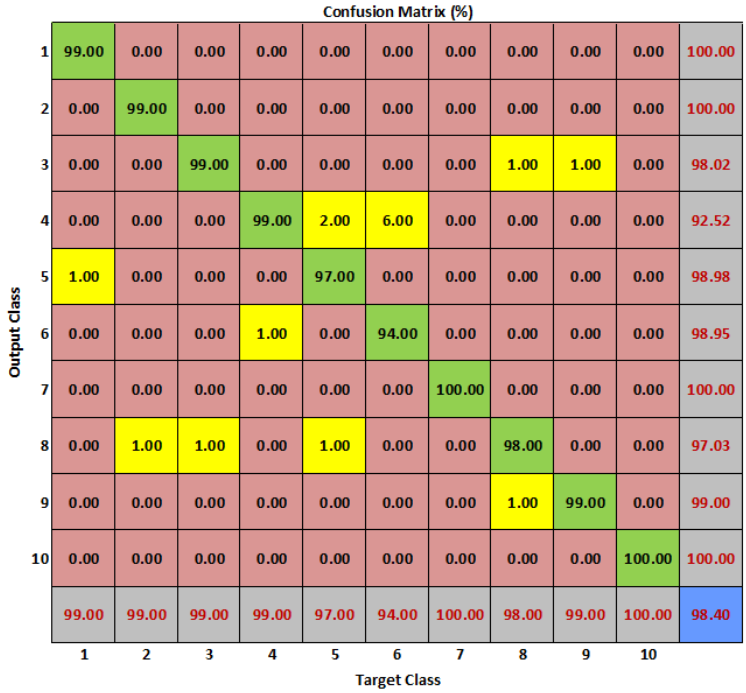

Figure 7 Confusion matrix of NN-LMKF with 12 neurons in the hidden layer.

For comparison, all samples were also classified using *k*-nearest neighborhood (*k*-NN) classifier. To obtain the best classification accuracy for *k*-NN, the value of k was determined empirically, varying from $k=1$ to $k=10$. The best classification accuracy of *k*-NN was 77.40%, achieved using $k=1$ and $k=2$. However, the best classification accuracy of *k*-NN was lower than the classification accuracy of NN-LMKF. This result shows that the hybrid neural network and linear model based on a Kalman filer together with statistical color features and the derivative of radius function can be used to recognize natural produce accurately.

To support the comparison of classification accuracy between NN, NN-LMKF, and *k*-NN, analysis of variance (ANOVA) was performed with level of significance $\alpha=0.05$. ANOVA was used to show that the mean classification accuracies for NN, NN-LMKF, and *k*-NN are significantly different. This analysis used the following hypothesis:

$$H_0 : \mu_1 = \mu_2 = \mu_3$$
$$H_1 : \text{ at least two of } \mu_1, \mu_2, \mu_3 \text{ are not equal}$$

where $\mu_1, \mu_2,$ and μ_3 are the mean classification accuracies for NN, NN-LMKF, and *k*-NN, respectively. The result of ANOVA is shown in Table 2. As can be seen in Table 2, since the value of *p* is less than 0.05, the decision of ANOVA = H_0 is rejected. Therefore, it can be concluded that at least two means of NN, NN-LMKF, and *k*-NN are different.

Table 2 ANOVA Result.

Source	Sum of squares	Degrees of freedom	Mean squares	F	p
Columns	1110	2	555	21.62	0.0001
Error	308.1	12	26.675		
Total	1418.1	14			

Furthermore, a multiple comparison test was used to determine which pairs of means are significantly different. Several paired mean comparison tests were performed using the following hypothesis:

$$H_0 : \mu_i - \mu_j = 0$$
$$H_1 : \mu_i - \mu_j \neq 0$$

where $i, j = 1,2,3, i \neq j$, $\mu_1, \mu_2,$ and μ_3 are the mean classification accuracies for NN, NN-LMKF, and k-NN, respectively. The result of the multiple comparison test is shown in Table 3. As can be seen in Table 3, since all 95% confidence intervals for mean difference do not contain zero, the decision of multiple comparison = H_0 is rejected. Therefore, it can be inferred that all mean classification accuracies for NN, NN-LMKF, and k-NN are significantly different.

Table 3 Multiple comparison test result.

Groups		Mean difference	95% confidence interval for mean difference	
i	j		Lower bound	Upper bound
1	2	-9.00	-15.98	-2.02
1	3	12.00	5.02	18.98
2	3	21.00	15.02	27.98

5 Conclusion

In this paper, a method for recognizing natural produce using computer vision was proposed. The proposed method consists of several steps, including image

acquisition, pre-processing, segmentation, feature extraction, and classification. The produce images were captured from the top using a camera. The captured images were processed prior to feature extraction. The proposed method uses 12 statistical color features in HSV color space and 4 shape features derived from the derivative of radius function. A hybrid neural network and linear model based on a Kalman filter (NN-LMKF) was used to classify the produce based on the extracted features.

One thousand images acquired from ten categories of natural produce were used to validate the proposed method. Five-fold cross validation was used to construct training and testing data sets. The experiment results show that NN-LMKF achieves better performance than original NN. The best classification accuracy for NN-LMKF was 98.40%, achieved by using 12 neurons in the hidden layer. Furthermore, the classification accuracy of NN-LMKF was greater than the classification accuracy of *k*-NN. The result of statistical analysis also showed that all mean classification accuracies for NN, NN-LMKF, and k-NN are significantly different. Although it was shown that the proposed method can recognize natural produce accurately, the proposed method can only be applied when the acquired image contains one single object. If the acquired image contains more than one object then the shape feature cannot correctly represent the object. This limitation is a motivation for future research.

References

[1] Bolle, R.M., Connell, J.H., Haas, N., Mohan, R. & Taubin, G., *VeggieVision: A Produce Recognition System*, in Proc. of Proceedings 3rd IEEE Workshop on Applications of Computer Vision, 1996, WACV '96, pp. 244-251, 1996.

[2] Roomi, S.M.M., Priya, R.J., Bhumesh, S. & Monisha, P., *Classification of Mangoes by Object Features and Contour Modeling*, in Proc. of 2012 International Conference on Machine Vision and Image Processing (MVIP) pp. 165-168, 2012.

[3] Wang, S., Yang, X., Zhang, Y., Phillips, P., Yang, J. & Yuan, T-F., *Identification of Green, Oolong and Black Teas in China via Wavelet Packet Entropy and Fuzzy Support Vector Machine*, Entropy, **17**(10), pp. 6663-6682, 2015.

[4] Zhang, Y., Yang, X., Cattani, C., Rao, R., Wang, S. & Phillips, P., *Tea Category Identification Using a Novel Fractional Fourier Entropy and Jaya Algorithm*, Entropy, **18**(3), pp. 77, 2016.

[5] Mishra, B.K., Bharadi, V.A., Nemade, B., Arakeri, M.P. & Lakshmana, *Computer Vision Based Fruit Grading System for Quality Evaluation of Tomato in Agriculture industry*, Procedia Computer Science, **79**, pp. 426-433, 2016.

[6] Waltner, G., Schwarz, M., Ladstätter, S., Weber, A., Luley, P., Bischof, H., Lindschinger, M., Schmid, I. & Paletta, L., MANGO – Mobile Augmented Reality with Functional Eating Guidance and Food Awareness, New Trends in Image Analysis and Processing – ICIAP 2015 Workshops Proceedings, V. Murino, E. Puppo, D. Sona, M. Cristani, C. Sansone (ed(s).), Springer International Publishing, pp. 425-432, 2015.

[7] Siswantoro, J., Prabuwono, A.S. & Abdullah, A., *Volume Measurement Algorithm for Food Product with Irregular Shape using Computer Vision based on Monte Carlo Method*, Journal of ICT Research and Applications, **8**(1), pp. 1-17, 2014.

[8] Prabuwono, A.S., Siswantoro, J. & Abdullah, A., *Natural Produce Classification Using Computer Vision Based on Statistical Color Features and Derivative of Radius Function*, Applied Mechanics & Materials, **771**, pp. 242-247, 2015.

[9] Faria, F.A., dos Santos, J.A., Rocha, A. & Torres, R.S., *Automatic Classifier Fusion for Produce Recognition*, in Proc. of Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on, pp. 252-259, 2012.

[10] Rocha, A., Hauagge, D.C., Wainer, J. & Goldenstein, S., *Automatic Fruit and Vegetable Classification from Images*, Computers and Electronics in Agriculture, **70**(1), pp. 96-104, 2010.

[11] Rocha, A., Hauagge, D.C., Wainer, J. & Goldenstein, S., *Automatic Produce Classification from Images Using Color, Texture and Appearance Cues*, in Proc. of Computer Graphics and Image Processing, 2008. SIBGRAPI '08. XXI Brazilian Symposium on, pp. 3-10, 2008.

[12] Zhang, Y., Phillips, P., Wang, S., Ji, G., Yang, J. & Wu, J., *Fruit Classification by Biogeography-Based Optimization and Feedforward Neural Network*, Expert Systems, **33**(3), pp. 239-253, 2016.

[13] Wang, S., Zhang, Y., Ji, G., Yang, J., Wu, J. & Wei, L., *Fruit Classification by Wavelet-Entropy and Feedforward Neural Network Trained by Fitness-Scaled Chaotic ABC and Biogeography-Based Optimization*, Entropy, **17**(8), pp. 5711, 2015.

[14] Zhang, Y., Wang, S., Ji, G. & Phillips, P., *Fruit Classification using Computer Vision and Feedforward Neural Network*, Journal of Food Engineering, **143**, pp. 167-177, 2014.

[15] Siswantoro, J., Prabuwono, A.S., Abdullah, A. & Idrus, B., *A Linear Model Based on Kalman Filter for Improving Neural Network Classification Performance*, Expert Systems with Applications, **49**(1), pp. 112-122, 2016.

[16] Bradski, G. & Kaehler, A., *Learning OpenCV: Computer Vision with The OpenCV Library*, O'Reilly Media, Inc., 2008.

[17] Gonzalez, R.C. & Woods, R.E., *Digital Image Processing*, 2nd ed. Prentice Hall, 2002.

[18] Zheng, C. & Sun, D-W., *3rd Chapter – Object Measurement Methods, Computer Vision Technology for Food Quality Evaluation*, S. Da-Wen (Ed.), Academic Press, pp. 57-80, 2008.
[19] Priddy, K.L. & Keller, P.E., *Artificial Neural Networks: An Introduction*, SPIE Press, 2005.
[20] Han, J., Kamber, M. & Pei, J., *Data Mining: Concepts and Techniques*, 3rd ed. Elsevier Science, 2011.
[21] May, R.J., Maier, H.R. & Dandy, G.C., *Data Splitting for Artificial Neural Networks Using SOM-Based Stratified Sampling*, Neural Networks, **23**(2), pp. 283-294, 2010.