Machine Learning Methods for Assessing Freshness in Hydroponic Produce

Konlakorn Wongpatikaseree, Narit Hnoohom and Sumeth Yuenyong
Department of Computer Engineering
Faculty of Engineering, Mahidol University
Nakhon Pathom, Thailand
{konlakorn.won, narit.hno, sumeth.yue}@mahidol.ac.th

***Abstract*—Smart farms are increasing in both number and level of technology used. Image processing had been applied to hydroponic farms to detect disease in plants, but detecting the freshness of vegetable had not been addressed as much. In this work we applied image processing and machine learning technologies to the task of distinguishing between fresh and withered vegetable. We compared 3 classical machine learning classifier: decision tree, Naive Bayes, Multi-Layer Perceptron; and one type of deep neural network. Manual feature extraction was performed for the classical machine learning, while the input to the deep neural network was the raw images. We collected the data by taking one image of the vegetable every 10 minutes for one week each time. We labeled the data by considering vegetable from day 1 and day 2 to be fresh while from day 3 onward was considered wither. Experiment results show that the best model for this task was decision tree with a test accuracy of 98.12%. Deep neural network did not perform as well as expected. We hypothesize that the reason is due to overfitting of the training data since the training accuracy for deep neural network was as high or even higher than other classifiers.**

***Index Terms*—smart farm, image processing, deep learning, freshness, machine learning**

I. Introduction

Food production is an important challenge in the 21st century [1]. This is especially true for an agricultural based economy such as Thailand's. However at present Thai agricultural sector faces many challenges such as overuse of chemical fertilizer and pesticides, increasingly unpredictable weather, and land overuse. A popular solution is hydroponic, where plants are grown without soil in a controlled and closed environment. Technology is applied in hydroponic farms order to increase yield and quality while minimizing water and chemical use. For example, real time monitoring of growing parameters such as water/air temperature, air humidity, and concentration of fertilizer in the water. Such monitoring is a common feature for smart hydroponic farms.

Beside from monitoring the growing parameters, recently there are many studies which investigate monitoring of the plants themselves using image processing and computer vision technologies. Usually the purpose for such monitoring is to detect disease in plants. Because many diseases affect the leaf, image processing is applied to images of leaves in order to detect diseases. Examples includes [2]–[4]. The reader is referred to [5]–[7] for an overview of relevant image processing techniques in this area.

On the other hand, determining whether a vegetable is fresh through images had not received much attention. The difference between fresh and slightly withered vegetable can be subtle even to humans. Being able to sort vegetables according to their freshness can be beneficial to both consumers and producers - consumers get fresh product and producers can reduce waste. In this study we focus on detecting the freshness of hydroponic produce through image processing and machine learning techniques. We compared 3 popular machine learning methods for image classification, as well as deep neural network [8], which have become the state-of-the-art in image recognition in recent years. The experiment result shows that deep neural network does not perform better than all the traditional machine learning methods for this particular task, probably due to overfitting the relatively small size of the training data.

II. Methodology

We had two set of red oak lettuce that we took images of. The difference between the two sets is one has moist wrapping at the root and the other set does not. The reason for this is that some supermarket keep these kind of lettuce moist while other do not. Each set consists of two red oak lettuce and they set are placed in a separate opaque box so that the lighting inside can be controlled. Inside each box there is a Raspberry Pi board, LED light and a camera. We programmed the Raspberry to take an image every 10 minutes continuously for 7 days. At the start of the week both sets of lettuce were freshly picked, and we just let them wither naturally over the week. This was repeated for about 8 weeks. Images from the first two days of each week were considered fresh, while images from day 3 onward were considered withered. Fig. 1 shows an example of fresh vegetable, while Fig. 2 shows an example of withered vegetable.

Fig. 3 and Fig. 4 show time lapse photo of vegetable with and without moist wrapping, respectively. There are 16 images in each figure, where each image corresponds to about 10.5 hours. Visually inspecting the time lapse photo, it can be seen that the vegetable starts to wither around the 6th image, which corresponds to around 63 hours. To make a rule that is simple to remember, we choose 72 hours or three days as the cutoff point of fresh vs. wither for both moist and dry conditions.

978-1-7281-0164-4/18/$31.00 ©2018 IEEE

Fig. 1. An example of fresh vegetable.

Fig. 2. An example of withered vegetable.

Fig. 3. Time lapse photo of vegetable with moist wrapping.

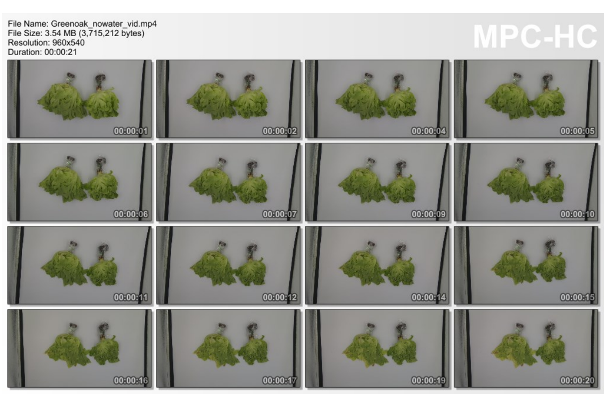

Fig. 4. Time lapse photo of vegetable without moist wrapping.

Once the images had been gathered, for the traditional machine learning methods feature extraction from raw camera images is necessary. There are 21 features total as follows:

1) Maximum value of the red channel (RGB color space)
2) Mean value of the red channel
3) Standard deviation of the red channel
4) Maximum value of grayscale image (converted from color image)
5) Mean value of grayscale image
6) Standard deviation of grayscale image
7) Maximum value of the blue channel
8) Mean value of the blue channel
9) Standard deviation of the blue channel
10) Maximum value of the green channel
11) Mean value of the green channel
12) Standard deviation of the green channel
13) Maximum value of hue (HSV color space)
14) Mean value of hue
15) Standard deviation of hue
16) Maximum value of saturation
17) Mean value of saturation
18) Standard deviation of Saturation
19) Maximum of value
20) Mean of value
21) Standard deviation of value

Feature extraction was performed using MATLAB and its image processing toolbox. For the deep neural network classifier, the input is the resized raw images.

A. Classical Machine Learning Classifiers

For the classical machine learning classifiers, we considered Naive Bayes[1], Decision Tree and Multi-Layer Perceptron (MLP). The particular algorithm for building decision tree was C4.5 [9] which is implemented in the Weka [10] machine learning package as J48. We also used Weka's implementation of Naive Bayes and MLP classifiers. Weka has 4 methods of evaluating classifiers: "use training set", "supplied test set", "cross-validation", and "percentage split". Use training set uses the same data for training and testing, supplied test set uses a separate user-supplied data for testing, cross-validation performs n-fold cross validation and percentage split reserves a specific portion of the data for testing. For supplied test set, we used the first 10 images of each class from each week as the test set. For cross-validation, we set the number of folds to 10.

B. Deep Neural Network Classifier

For the deep neural network classifier, we used the Inception-V3 architecture [8] pre-trained on the ImageNet dataset [11]. This pre-trained model is available in the Tensorflow [12] library in the tf.keras.applications.InceptionV3 module. Due to deep neural network taking much longer to train than classical machine learning models, we only used one type of testing scheme for testing the deep neural network

[1]https://en.wikipedia.org/wiki/Naive_Bayes_classifier

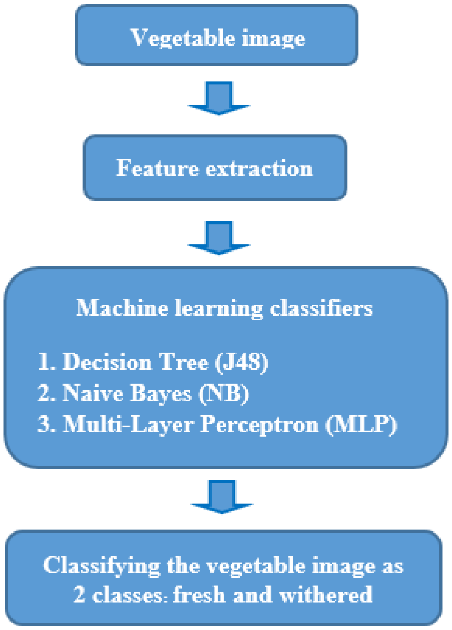

Fig. 5. The processing flow for experiments using classical machine learning classifiers.

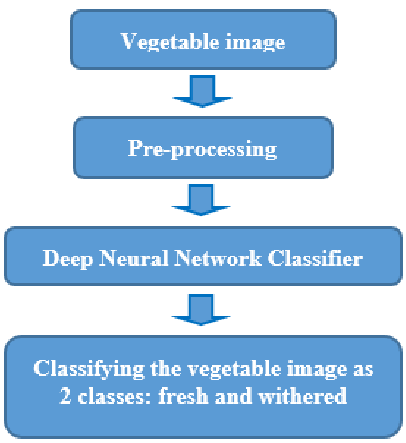

Fig. 6. The processing flow using deep neural network as classifier.

classifier, which corresponds to percentage split in Weka. We choose the split to be 80/20 for both classical machine learning and deep neural network. Thus, between classical learning and deep neural network, we can only compare them using the percentage split scheme and for the same number of classes for fair comparison. The processing flows for classical machine learning and deep learning are shown in Figs. 5 and 6 respectively.

III. Experiments

A. Results for Classical ML Classifiers

We performed several rounds of experiments, three for the classical machine learning algorithms and five for deep neural network. In the first round, the lettuce were divided into with moist wrapping and without. the size of the training data was 2,400 images. The classes were withered and fresh. The classifiers were J48 and MLP. The result for this round of experiments is shown in Table I

For the second round of experiments, the details are similar to round one except that the size of the training data was increased to 18,936 images and Naive Bayes classifier (NB) was also included. The classes were still withered and fresh. The result for this round is shown in Table II

In round three of experiments, we wanted to see if the classifiers can tell the difference between vegetables that are fresh and just one day old. Thus the classes for this round are: day 1 and day 2. The size of the training data for this round was 19,523 images. The classifiers were the same as in round two. The result is shown in Table III

Considering the results in Tables II and III, it can be seen that the J48 and MLP classifiers can distinguish the difference between fresh vs. withered vegetable (round 2) and fresh vs. one day one (round 3) with accuracies above 90%. The Naives Bayes classifier did not perform very well, this is likely because the features set strongly violates the independent assumption made by Naives Bayes. That is, the intensity of each color channel is likely to be correlated.

Having completed the experiments for classical ML classifiers, we wanted to see if deep neural network can achieve better result. Thus we performed more experiments using deep neural network

TABLE I
Results for round one of experiments.

		Cross-Validation	Percentage Split
J48	set 1	97.88%	97.29%
	set 2	97.18%	97.06%
	set 3	97.25%	95.69%
MLP	set 1	97.13%	96.67%
	set 2	96.86%	97.45%
	set 3	96.90%	97.65%

TABLE II
Results for round two of experiments.

	Use Training Set	Supplied Test Set	Cross-Validation	Percentage Split
J48	99.65%	54.18%	98.10%	98.12%
NB	55.03%	40.59%	54.95%	53.92%
MLP	93.87%	56.75%	93.66%	93.36%

TABLE III
Results for round three of experiments.

	Use Training Set	Supplied Test Set	Cross-Validation	Percentage Split
J48	99.61%	62.41%	98.06%	97.92%
NB	59.52%	62.76%	59.49%	58.08%
MLP	93.58%	63.01%	93.94%	94.57%

TABLE IV
EXPERIMENTAL RESULT FOR THE DEEP NEURAL NETWORK CLASSIFIER.

Round	Test Images	Learning Rate	Training Steps	Train Accuracy	Test Accuracy
1	1,400	0.1	7,000	97.7%	93.65%
2	4,000	0.1	7,000	97.7%	80.75%
3	4,000	0.01	7,000	97.2%	83.25%
4	4,000	0.001	7,000	93.9%	83.00%
5	4,000	0.1	15,000	98.3%	77.50%

B. *Results for Deep Neural Network Classifier*

The deep neural network we used was the Inception-V3 architecture [13]. We utilized the principle of transfer learning [14]. It involves taking a neural network that had been trained a large dataset, such as ImageNet [11] and fine-tune the weights by continuing to train on a smaller dataset using small learning rate. Transfer learning is essential because our data is not big enough to train a networks as large as the Inception architectures completely from scratch without overfitting the training data. The training data for the deep neural network experiments was the same 18,936 images from round 2, which is the closest in experiment configuration to the deep neural network experiments. We trained the neural network 5 times in total, each time using different learning rates and/or training steps. Due to the time it takes to train a deep neural network, we did not evaluate it using all the same methods as in the classical machine learning classifier experiments, rather we used just the percentage split evaluation at the same 80-20 proportion. The results for deep neural network is shown in Table IV. The best result was from the first configuration at 93.65%, with 1400 test images, learning rate of 0.01 and number training steps of 7,000. When the number of test images was increased to the full set of 7,000, the best accuracy was that of configuration 3 at 83.25%. It can be seen that the accuracy for deep neural network is significantly lower than that of J48 and MLP classifiers from round 2. We suspect that this is due to overfitting since the training accuracies in all cases for deep neural network was around 97-98 percent.

IV. CONCLUSION

In this paper, we presented a comparison between different machine learning classifier for the detection of fresh/withered vegetable. We collected the images in light opaque boxes over many weeks by taking images of vegetables every 10 mins continuously for a 7 days each round. For the classical machine learning methods we performed feature extraction from the images which includes 21 features. For the deep neural network we used the raw images as input. Among the classical machine learning methods, J48 was found to have the best accuracy. Comparing between classical machine learning and deep neural network was done only percentage split scheme due to the time it takes to train deep neural network. The percentage split was set at 80/20. It was found that deep neural network did not outperform J48. This may be because our training dataset is too small for deep neural network. For further work we could improve the performance of deep neural network by using data augmentation or by adding dropout [15] layers in the network.

ACKNOWLEDGMENT

This research was supported by National Research Council of Thailand (NRCT) under the project named “Developing Smart Farm and Traceability System for Agricultural Products using Internet of Things Technology”.

REFERENCES

[1] H. C. J. Godfray, J. R. Beddington, I. R. Crute, L. Haddad, D. Lawrence, J. F. Muir, J. Pretty, S. Robinson, S. M. Thomas, and C. Toulmin, “Food security: the challenge of feeding 9 billion people,” *science*, p. 1185383, 2010.

[2] S. B. Dhaygude and N. P. Kumbhar, “Agricultural plant leaf disease detection using image processing,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 2, no. 1, pp. 599–602, 2013.

[3] M. S. B. Jagtap and M. S. M. Hambarde, “Agricultural plant leaf disease detection and diagnosis using image processing based on morphological feature extraction,” *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume*, vol. 4, pp. 24–30, 2014.

[4] S. Bashir and N. Sharma, “Remote area plant disease detection using image processing,” *IOSR Journal of Electronics and Communication Engineering*, vol. 1, no. 6, pp. 31–34, 2012.

[5] J. K. Patil and R. Kumar, “Advances in image processing for detection of plant diseases,” *Journal of Advanced Bioinformatics Applications and Research*, vol. 2, no. 2, pp. 135–141, 2011.

[6] P. Chaudhary, A. K. Chaudhari, A. Cheeran, and S. Godara, “Color transform based approach for disease spot detection on plant leaf,” *International Journal of Computer Science and Telecommunications*, vol. 3, no. 6, pp. 65–70, 2012.

[7] S. N. Ghaiwat and P. Arora, “Detection and classification of plant leaf diseases using image processing techniques: a review,” *International Journal of Recent Advances in Engineering & Technology*, vol. 2, no. 3, pp. 1–7, 2014.

[8] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[9] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip *et al.*, “Top 10 algorithms in data mining,” *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. Ieee, 2009, pp. 248–255.

[12] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, “Tensorflow: a system for large-scale machine learning.” in *OSDI*, vol. 16, 2016, pp. 265–283.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.