

A Tomato Quality Identification Method Based on Raman Spectroscopy and Convolutional Neural Network

Wu Yin^{1,a}, Chenying He^{1,b} and Qibao Wu^{2,c}

¹Shenzhen Wissea Electronic Technology Co., Ltd, 1202, 2nd Chinese overseas scholar venture building, High-tech park, Nanshan district, Shenzhen, Guangdong, China,

²School of Intelligent Manufacturing and Equipment, Shenzhen Institute of Information Technology, Shenzhen, China

Email: ^ayinwu@wissea.com, ^bhechenying@wissea.com, ^cwuqb@szit.edu.cn

Abstract: in recent years, more and more technologies have been applied in monitoring growth and production efficiency of plants, i.e. agricultural Internet of Things (IOT) and new information-aware technologies. The architecture of the IOT is divided into four layers, i.e., the sensing layer, network layer, processing layer and application layer[1-4]. Among them, the perception layer is the facial features and the skin of the IOT, which is the basis of the IOT [5]. Raman spectroscopy technology has the advantages of fast, simplicity, accuracy, non-destructive and automatic identification, which has become a powerful analytical verifying method. The method of tomato quality identification that based on the Raman spectroscopy combined with convolutional neural network (CNN)[6]was explored. The Raman spectrum of tomato was collected by Raman sensor to construct a neural network with deep network structure. Through repeatedly learning and training in Raman map, we can determine the map recognition model of high quality tomatoes and use matplotlib to realize the identification simulation.

1. Introduction

As agricultural safety accidents have aroused people's high concern for food safety,[7] solving the problem of agricultural product safety has become an urgent task for the government and society. In recent years, IOT technology has been applied to many fields of agriculture, including agricultural environmental monitoring, greenhouse production control, water-saving irrigation, meteorological monitoring, product safety and traceability, as well as equipment intelligent diagnosis and management. Through the data transmitted by sensors, crop information can be grasped at anytime and anywhere, and the crops can be remotely managed. Experts can diagnose scientifically on the pests and diseases of the crops as well. This is important for the resolution of food safety issues. Nowadays, the technologies of big data and cloud computing have been investigated extensively that applied in various fields. The IOT is actually a combination of objects and Internet through the sensing devices communication. Information exchange, communication, data acquisition, positioning, tracking, monitoring and operation are carried out in the terminals to realize intelligent identification and management of the network [8].

It is well known that the smart agriculture is based on network information technology to realize the connection between IOT technology and traditional agriculture. This paper proposes to apply Raman sensing technology to connect with other software and mobile service platforms through terminals communication and analyze data by using convolutional neural network algorithm, and to monitor



agricultural production process effectively. The IOT and Raman sensing technology provide low-cost, high efficient modules for crop growing areas to monitor plant growth status and the environment situation in real time. In addition, the goal of this technology is to discover various problems in the process of crop growth. Through the integrated feedback service system, the intelligent control system can be launched in real time to realize the rational use of agricultural resources, improve the ecological environment, and achieve high yield and quality of agricultural products. [9] The IOT promotes the entire agricultural production more efficient and intelligent.[10]

2. Experimental materials and instruments

The samples that applied in this application are from the Pingshan Agricultural Park Centre in Shenzhen, China. The surveillance instrument is one multi-function Raman sensor that developed by our team, as shown in Figure 1. The fiber optic probe of the Raman spectrometer is designed to be rotatable and movable, which increases its flexibility and makes it more widely availability. In addition, control modules and wireless transmission modules have been applied in the spectrometer and laser to make it more intelligent.

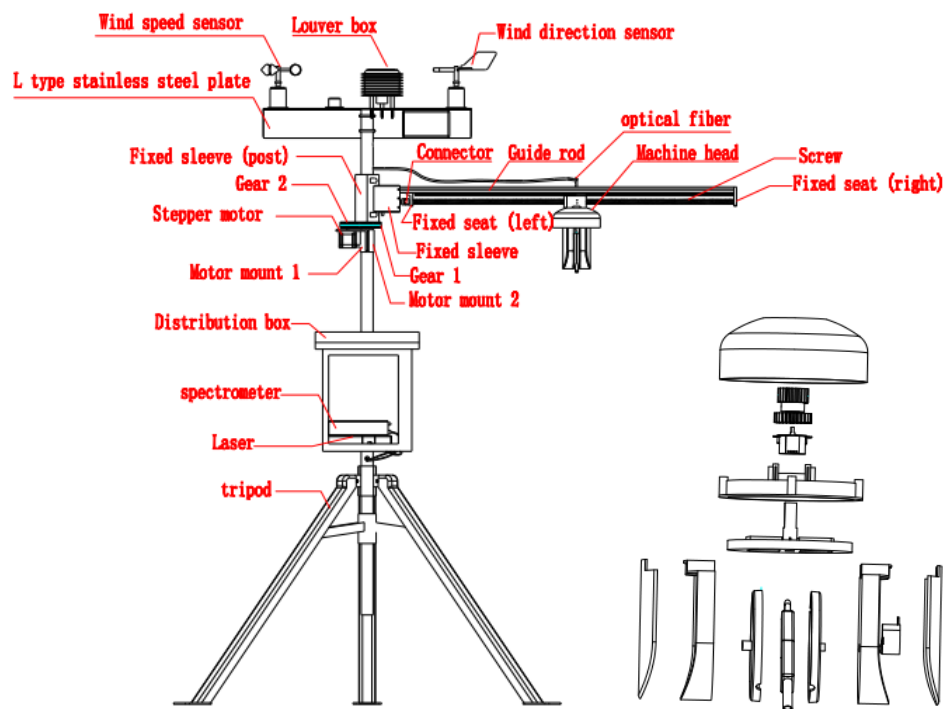


Figure 1. Multi-functional Raman sensing equipment. The sensor device is based on Raman spectrometer and integrates a variety of sensors

The data in this application from the above Raman sensors. The data were acquired by every 10 minutes for an interval of 30s. by rotating or moving the Raman probe every ten minutes, the location of each acquisition can be different. Finally, we use the average of multiple acquisitions in one day as the data information of the samples.

3. Neural network construction

The neural network structure design is shown in Figure.2. It is mainly divided into three parts: input layer, hidden layer and output layer. [11] The total number of layers is 47, which shown as follows: 5(front)+3(block1_module1)+3(block1_module2)+3(block1_module3)+3(block2_module1)+5(block2_module2)+5(block2_module3)+5(block2_module4)+5(block2_module5)+4(block3_module1)+3(block3_module2)+3(block3_module3)

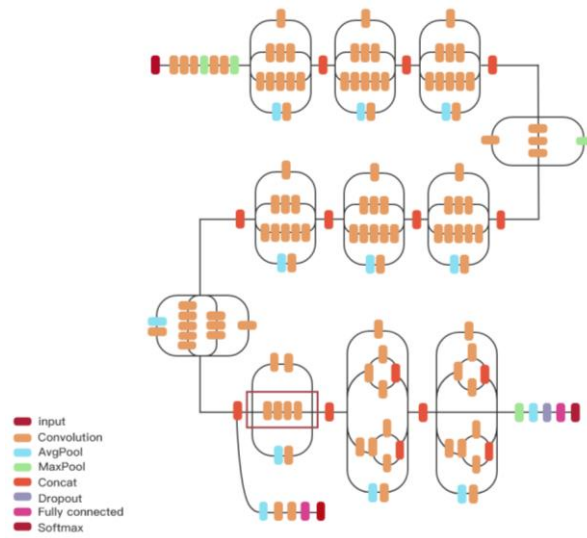


Figure.2. Neural network hierarchy diagram: input layer, hidden layer and output layer. The hidden layer is composed of convolution layer

The project model applied the GoogLeNet Inception V3, [12,13] which introduces the idea of Factorization into small convolutions. Splitting a large two-dimensional convolution into two smaller one-dimensional convolutions, such as splitting a 7*7 convolution into a 1*7 convolution and a 7*1 convolution, or a 3*3 convolution Split into a 1*3 convolution and a 3*1 convolution. On the one hand, reducing parameters and improving computational efficiency to prevent over-fitting. On the other hand, applying a layer of nonlinear extended model expression ability. It can be seen that the result of this asymmetric convolution structure splitting is more obvious than splitting into several identical small convolution nuclei. Compared with the VggNet, this network can handle more spatial features and increase the diversity of features (Vgg16) [14], furthermore the accuracy rate can be improved by about 0.8%.

In terms of loss functionality, this network input can be 229*229*3 or 224*224*3, which use the Label Smoothing Regularization (LSR) method. The LSR is a method of reducing the model over-fitting by adding noise to the output y to constrain the model. Usually when the model trained, the label $q(k/x)$ takes the form of one-hot, and the output of the model is the probability distribution after the softmax is normalized. The purpose of the training is to make the distribution of $p(k/x)$ as close as possible to $q(k/x)$, but this method is prone to over fitting. In order to solve this problem, weights were added to a certain probability distribution in the original label to constitute a new label. Through this way, it can prevent the model prediction value from being excessively concentrated on the category with higher rate, and will increase the number of small probability categories. Note that the loss function becomes as shown in equation (1):

$$H(q', p) = -\sum_{k=1}^K \log(p(k)q'(k)) = (1-\alpha)H(q, p) + \alpha H(u, p) \quad (1)$$

As it can be seen from the loss function, the LSR is equivalent to use two losses. When u obeys a uniform distribution, $H(u, p)$ is a constant, which can measure the degree of dissimilarity between the predicted distribution p and the uniform distribution, and plays a regularization role.

In order to quickly obtain self-learning results, the gradient descent method is used to correct the weighting coefficients of the network. Before the network model training, the data set is divided into a training set and a test set, the ratio is 8:2. The cross-checking method is adopted, and the number of iterations is 8000 times.

4. Experimental results and analysis

By randomly adding noise, cropping, flipping, mirroring and Gaussian blur, the data in the training data is to be enhanced to improve the generalization ability of the image. The training of the model adopts the sliding average model and other optimization methods to prevent over-fitting and training. When the number of times is less than 2000, the model converges faster. After 2000 trainings, the model converges slowly and the error loss changes little. The training of the network model is basically completed, as shown in Figure.3

4.1. Training of neural network models

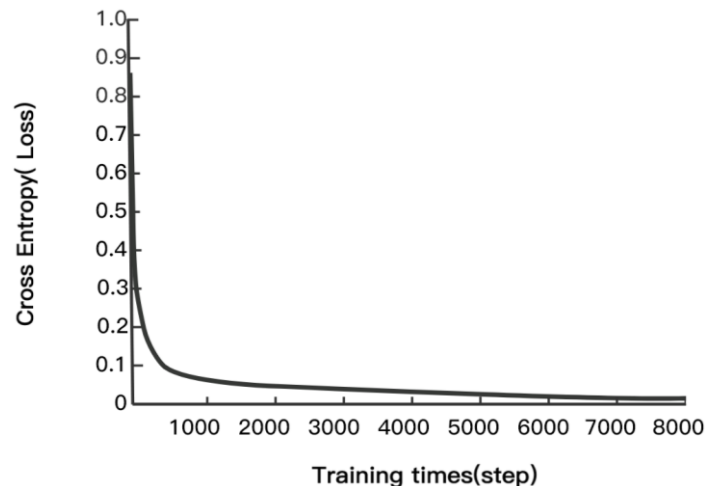


Figure 3. Neural network training performance chart. In the Figureure, the curve converges slowly, indicating that the training is basically completed.

It can be seen from Figure 3 that the total error of the output is significantly decreased during the training period from 0 to 2000 times; the total output error is basically stable during 2000 to 8000 times.

4.2. Deep neural network model prediction performance

To verify the predictive performance of the network model, the model was applied to identify tomato quality. 1600 tomatoes were randomly selected from the plantation for verification and identification analysis. As shown in Figure 4, the accuracy of the network model reached 94.6%, while the verified results are shown in Figure 5.

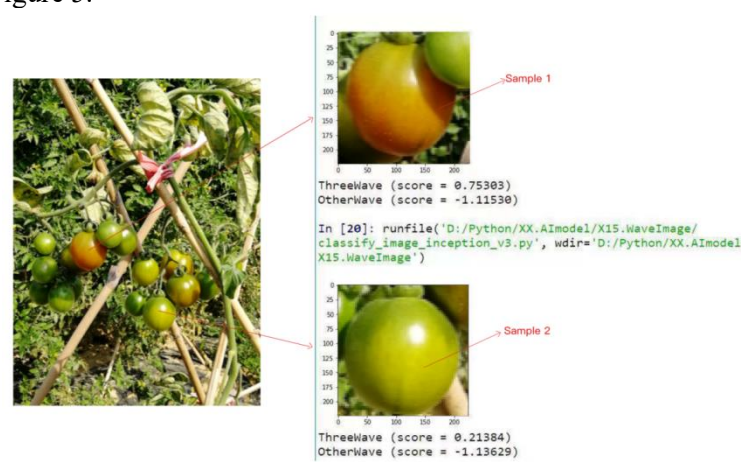


Figure 4. Identification method. The result of verification of tomato recognition, the score value reflects the excellent degree of tomato

```

Instructions for updating:
Use standard file APIs to check for files with this prefix.
10914 12:34:54.922921 7260 saver.py:1280] Restoring parameters from satellite/train_dir\model.chkpt-3753
10914 12:34:57.704079 7260 session_manager.py:500] Running local_init_op.
10914 12:34:57.791085 7260 session_manager.py:502] Done running local_init_op.
10914 12:34:58.107102 7260 deprecation.py:323] From D:\Python\Anaconda3\envs\python37\lib\site-packages\tensorflow\python\taining\monitored_session.py:110: tf.nn.conv2d is deprecated and will be removed in a future version.
Instructions for updating:
To construct input pipelines, use the 'tf.data' module.
OMP: Info #212: OMP_AFFINITY: decoding x2APIC ids.
OMP: Info #210: OMP_AFFINITY: Affinity capable, using global cpuid leaf 11 info
OMP: Info #154: OMP_AFFINITY: Initial OS proc set respected: 0:1
OMP: Info #156: OMP_AFFINITY: 2 available OS procs
OMP: Info #157: OMP_AFFINITY: Uniform topology
OMP: Info #179: OMP_AFFINITY: 1 packages x 2 cores/pkg x 1 threads/core (2 total cores)
OMP: Info #214: OMP_AFFINITY: OS proc to physical thread map:
OMP: Info #171: OMP_AFFINITY: OS proc 0 maps to package 0 core 0
OMP: Info #171: OMP_AFFINITY: OS proc 1 maps to package 0 core 1
OMP: Info #250: OMP_AFFINITY: pid 6512 tid 5232 thread 0 bound to OS proc set 0
OMP: Info #250: OMP_AFFINITY: pid 6512 tid 6448 thread 1 bound to OS proc set 1
OMP: Info #250: OMP_AFFINITY: pid 6512 tid 2784 thread 2 bound to OS proc set 0
OMP: Info #250: OMP_AFFINITY: pid 6512 tid 2088 thread 3 bound to OS proc set 1
10914 12:35:30.568417 7260 evaluation.py:167] Evaluation (10/100)
10914 12:36:15.539531 7260 evaluation.py:167] Evaluation (20/100)
10914 12:36:55.631825 7260 evaluation.py:167] Evaluation (30/100)
10914 12:37:35.276092 7260 evaluation.py:167] Evaluation (40/100)
10914 12:38:17.493507 7260 evaluation.py:167] Evaluation (50/100)
10914 12:37:23.349046 7260 evaluation.py:167] Evaluation (60/100)
10914 12:38:51.920112 7260 evaluation.py:167] Evaluation (70/100)
10914 12:39:30.125277 7260 evaluation.py:167] Evaluation (80/100)
10914 13:00:07.402429 7260 evaluation.py:167] Evaluation (90/100)
10914 13:00:51.120920 7260 evaluation.py:167] Evaluation (100/100)
eval/accuracy(0.946)

```

Figure 5. Verification results. After 8000 times of training, the verification accuracy of the model was 94.6%

4.3. Repeatable trial

Ten samples of similar quality were taken from the tomato garden, and the repeatability trial and verification as well as analysis were carried out according to the above application, and then compared with the standard maps of various qualities. The RSD of the relative peak area of each of the obtained common peaks was 5.12%, and the relative retention time of each of the shared peaks was RSD=2.35%, indicating that the method was excellent in repeatability.

5. Conclusion

In this research, the Raman spectroscopy information of tomato has been acquired and analyzed by the Raman spectroscopy and the tomato quality identification model that based on the neural network has been constructed. The results show that the neural network with deep network structure has been applied to train a large number of tomato map data, while the trained model was compared with the predicted performance. Figure 4 and Figure 5 show that the output results of different quality tomatoes are obviously different. The accuracy of the feature data identification of the test set can be reached to 94.6%. At the same time, this proposed network model has been trialed repeatedly. The RSD of the peak relative area of the map was less than 5.12%, and the relative retention time was less than 2.35%. Therefore, it can be observed that the method based on the Raman spectroscopy combined with the convolutional neural network to identify tomato quality is more feasible and accurate.

Based on the Raman spectroscopy, this study preliminarily explored the method of identifying tomato quality by the tomato Raman spectroscopy. The method proved to be simple and fast, meanwhile the sample does not need to be sampled with high accuracy. However, in order to achieve full control of tomato quality, it is also suggested to combine other analytical testing methods and monitoring of environmental parameters. In conclusion, the identification method based on the Raman spectroscopy is feasible and controllable.

References

- [1] Wu, Z., Liu, L.H., Huang, Y.S., Wu, X.M.: A Community Health Service Architecture Based on the Internet of Things on Health-Care. *World Congress on Medical Physics and Biomedical Engineering*, vol. 39, pp.1317-1320, 2013.
- [2] Mashal, I., Alsaryrah, O., Chung, T.Y., Yang C.Z., Kuo, W.H., and Agrawal, D. P.: Choices for interaction with things on Internet and underlying issues, *Ad Hoc Networks*, vol. 28, pp. 68– 90, 2015.
- [3] Said, O., Masud, M.: Towards internet of things: survey and future vision, *International Journal of Computer Networks*, vol. 5, no. 1, pp. 1–17, 2013.
- [4] Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., and Du, H.Y.: Research on the architecture of internet of

- things, in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, vol. 5, pp. V5-484–V5-487, IEEE, Chengdu, China, August 2010.
- [5] Zhang, Y.F.: Internet of things technology and application[J]. *Communications and information technology*, 2010(1):50-53.
- [6] LeCun, Y., Jackel, L.D., Boser, B., et. al.: Hand written digit recognition: Applications of neural network chips and automatic learning[J]. *IEEE Communications Magazine*, 1989, 27(11):41-46.
- [7] Ji, M.S.: Pesticide residues in agricultural products and food safety[J]. *New agricultural*, 2012(1):45-47.
- [8] Huang, Y., Zhang, W.: *Intelligent agriculture monitoring system based on Internet of things*[J]. *Internet of things technology*, 2017,7(4):33-34.
- [9] Liu, Y.J.: Informationization helps farmers to overtake the corner. *Agricultural Products Weekly*, 2015-04-02.
- [10] Xiao, X.Q., Chen Y.: Market-oriented development of students' computational thinking ability. In *the computer age*, 2016-07-15.
- [11] Lee, C.Y., Xie, S.N., Gallagher, P., Zhang, Z., Tu, Z.W.: Deeply-supervised nets. In: *Eprint Arxiv*, 2014. 562–570
- [12] Szegedy, C., Vanhoucke, V. Ioffe, S., et al.: Rethinking the inception architecture for computer vision[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016:2818 — 2826.
- [13] Szegedy, C., Liu, W., Jiang, Y.Q., et. al.: Going deeper with convolutions[C]. *Proc of the 30th IEEE Conf on Computer Vision and Pattern Recognition*. Piscatawy, NJ: IEEE, 2015:37-46.
- [14] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition[J]. *ar Xiv: 1409. 1556*, 2014.