

## Research Article

# Fruit Classification Model Based on Residual Filtering Network for Smart Community Robot

Yulin Chen,<sup>1</sup> Hailing Sun ,<sup>1</sup> Guofu Zhou,<sup>1,2</sup> and Bao Peng<sup>3</sup>

<sup>1</sup>Guangdong Provincial Key Laboratory of Optical Information Materials and Technology & Institute of Electronic Paper Displays, South China Academy of Advanced Optoelectronics, South China Normal University, Guangzhou 510006, China

<sup>2</sup>Shenzhen Guohua Optoelectronics Tech. Co. Ltd., Shenzhen 518110, China

<sup>3</sup>Shenzhen Institute & Information Technology, Shenzhen, China

Correspondence should be addressed to Hailing Sun; sunsmile1225@163.com

Received 23 February 2021; Revised 6 March 2021; Accepted 11 March 2021; Published 28 March 2021

Academic Editor: Xin Liu

Copyright © 2021 Yulin Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of computer vision and robot technology, smart community robots based on artificial intelligence technology have been widely used in smart cities. Considering the process of feature extraction in fruit classification is very complicated. And manual feature extraction has low reliability and high randomness. Therefore, a method of residual filtering network (RFN) and support vector machine (SVM) for fruit classification is proposed in this paper. The classification of fruits includes two stages. In the first stage, RFN is used to extract features. The network consists of Gabor filter and residual block. In the second stage, SVM is used to classify fruit features extracted by RFN. In addition, a performance estimate for the training process carried out by the  $K$ -fold cross-validation method. The performance of this method is assessed with the accuracy, recall, F1 score, and precision. The accuracy of this method on the Fruits-360 dataset is 99.955%. The experimental results and comparative analyses with similar methods testify the efficacy of the proposed method over existing systems on fruit classification.

## 1. Introduction

China has achieved rich results in the field of smart cities through years of construction. These achievements have played an important role in alleviating the “big city disease,” improving the quality of urbanization, realizing refined and dynamic management, enhancing the effectiveness of urban management, and improving the residents’ quality of life. Community is the “smallest cell” of social management. It is an important aspect of the modernization of urban management system and management capacity to enhance the community management capabilities [1, 2]. With the increasing scale of community, there are many large communities in China. These communities need a lot of fruits and other living materials every day. A lot of labors are required to tasks such as fruit sorting. Labor force has dropped sharply during the epidemic. Therefore, demand for community robot with automatic sorting and store management is particularly urgent. It can not only make up for the shortage of labor but also save costs and improve efficiency [3].

Fruit sorting is the core function of intelligent community robot. It has been highly concerned by researchers and achieved a lot of research results in recent years. A fruit classification method was proposed by Arivazhagan et al. [4]. It extracted different features of fruits and used BP classification to classify. The accuracy of this method was reported to be 86%. A fuzzy logic and  $k$ -means clustering method for sorting fruit and vegetable was proposed by George [5]. The method achieved an accuracy of 86%. Kuang et al. [6] proposed a fruit detection method based on multiple-color channels. The method fused the histogram of oriented gradient, local binary pattern (LBP), and Gabor wavelet-based LBP features. The features were fused and used to train SVM on previously divided blocks of image to produce optimal results.

In addition, deep learning models based on fruit classification had been proposed by some researchers. Yu et al. [7] proposed a 13-layer convolutional neural network (CNN) for fruit recognition classification. The method used three types of data enhancement methods: image rotation, gamma

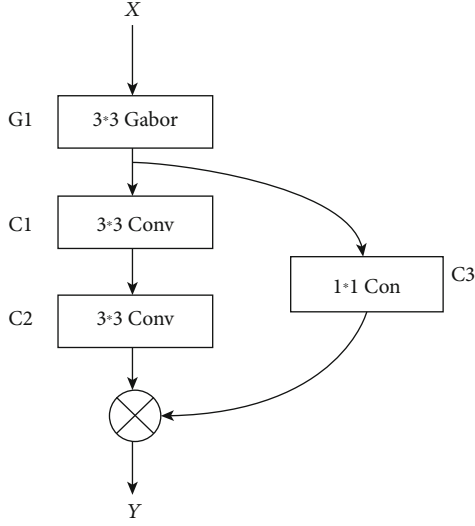


FIGURE 1: Residual filtering block structure. The G1 layer is a convolutional layer using a  $3 \times 3$  Gabor filter. Residual block is constituted by C1, C2, and C3.

correction, and noise injection. It also compared maximum pooling and average pooling. The gradient descent algorithm was used to optimize the CNN network. And the method achieved an accuracy of 94.94%. The method was 5% higher than the accuracy of the common CNN recognition method. An improved CNN to detect the appearance of cherries was proposed by Momeny et al. [8]. The method combining maximum pooling and average pooling was used to classify cherries after getting preprocessed images. Gao et al. [9] proposed a multicategory apple detection method based on Fast Region-based Convolutional Network (Fast R-CNN) method. The method achieved an average accuracy of 0.909, 0.899, 0.858, and 0.848 under five different shading conditions, respectively.

Traditional methods of fruit classification have achieved good classification results. But there had been many problems. Machine learning of fruit classification is a complex process. It needs to extract features of fruit [10]. However, the method of deep learning for fruit classification has problems of deep network layer and slow speed. Therefore, a fruit classification method based on the above problems is proposed in this paper. It is based on RFN and SVM for classification. RFN and SVM are used as the methods to extract features and classification, respectively.

The organization of the paper is as below. The principle of the method introduced in this paper is discussed in Section 2. The experimental process and result analysis are introduced in Section 3. The paper is concluded in Section 4.

## 2. Materials and Methods

Traditional methods of fruit classification need to extract features of fruits by manual work. Therefore, a method for fruit classification based on RFN and SVM is proposed in this paper. This section introduces the process and principle of fruit classification.

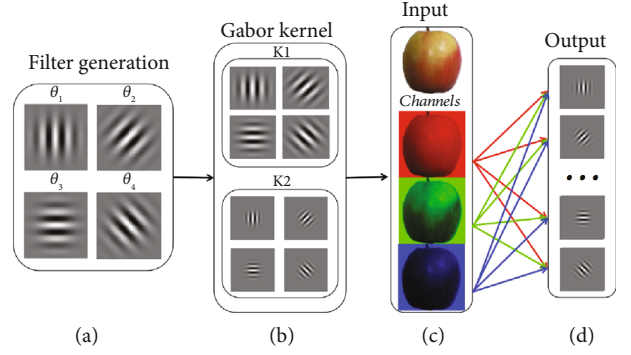


FIGURE 2: Gabor filter operations. (a) Filters are generated by rotating a sampled grid over four orientations  $\theta(\theta_1, \theta_2, \theta_3, \theta_4)$ . (b) Then, Gabor filters K1 and K2 are constructed. Finally, the Gabor filter operations (c) and (d) separately convolve every filter in Gabor kernel with every channel from the input.

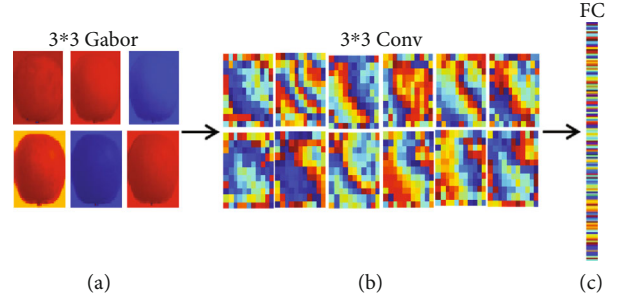


FIGURE 3: Residual filtering network structure. The G1 layer is a convolutional layer using a  $3 \times 3$  Gabor filter. Convolutional layers C1, C2, and C3 constitute a residual block. P1 and C4 layers are the pooling layer and convolutional layer, respectively. FC is a fully connected layer. And softmax is the output layer.

**2.1. Feature Extraction.** Feature extraction is an important process in fruit classification. RFN is proposed in this paper for feature extraction. It consists of residual filtering block. Residual filtering block is composed of Gabor filter and residual block. Gabor filter is used in residual filtering block to replace the convolution kernel. It can extract features of fruits in different directions and scales [11]. And residual block can solve problems of feature loss and gradient disappearance. The structure of the residual filtering block is shown in Figure 1.  $X$  and  $Y$  are input and output of the residual filtering block, respectively. The G1 layer is a convolutional layer using a  $3 \times 3$  Gabor filter. The residual block is constituted by C1, C2, and C3.

The Gabor filter can be convolved with image to obtain the local information in the spatial domain [12, 13]. The Gabor filter can be generated according to

$$G_{\theta}(x', y', \sigma, \gamma, \lambda, \psi) = e^{-a^2(x'^2 + y'^2 + y'^2)} \cos(\lambda x' + \psi), \quad (1)$$

where  $x$  and  $y$  mean the location of the pixels in the spatial domain [14]. The positions  $x'$  and  $y'$  after pixel rotation are calculated by Equations (2) and (3).  $G_{\theta}(x', y', \sigma, \gamma, \lambda, \varphi)$

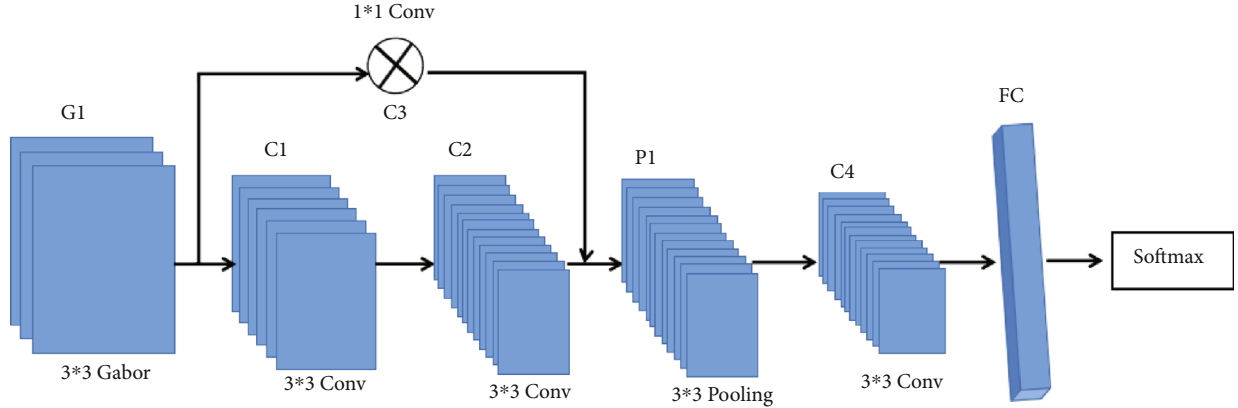


FIGURE 4: Feature map visualization. (a) The feature maps are obtained by convolution of the input image and the Gabor filter. (b) Feature maps output by the convolutional layer. (c) Feature maps output by the FC layer.

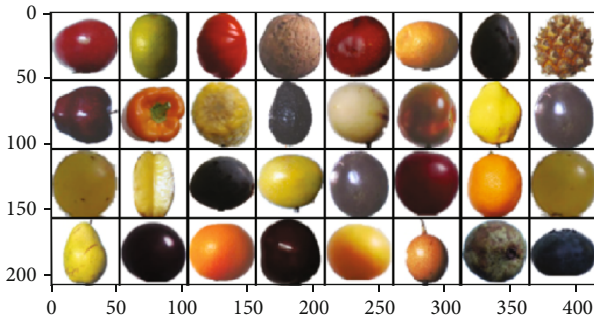


FIGURE 5: Dataset sample.

denotes the response of a Gabor filter with standard deviation  $\sigma$ , aspect ratio  $\gamma$ , wavelength  $\lambda$ , and phase offset  $\psi$  to an image at point  $(x', y')$  on the image plane [15].

$$x' = x \cos \theta - y \sin \theta, \quad (2)$$

$$y' = x \sin \theta + y \cos \theta. \quad (3)$$

The convolution process of the Gabor filter is shown in Figure 2. The parameters of the Gabor filter can be adjusted during BP. It is similar to the common convolution kernel. The Gabor filter can only convolve a single channel of image. It is different from common convolution kernel. The feature extracted by the Gabor filter is more robust than that extracted by common convolution kernel.

Feature loss and gradient disappearance occur in network transmission process [16]. These problems are solved by adding residual network in this paper. The output of the residual network is calculated by Equations (4) and (5). The equation of  $h(x_l) + F(x_l + W_l)$  can be realized through feedforward neural networks with “shortcut connections.”  $x_l$  and  $X_{l+1}$  are input and output of the convolutional layers  $l$ , respectively.  $h(x_l)$  is used when matching dimensions [17]. The dimension of input features can be increased or decreased by a convolutional layer when the number of input and output features of the residual network is different. The equation  $F(x_l, W_l)$  represents the residual mapping to be learned [18].

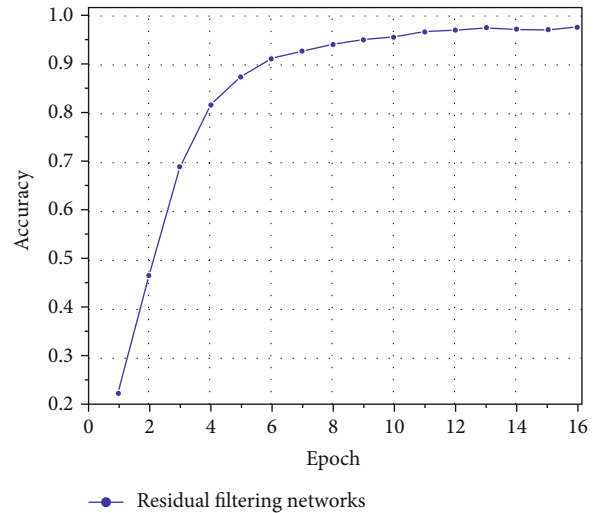


FIGURE 6: The relationship between accuracy of the training residual filtering model and epoch.

$W_l$  and  $W'_l$  are different square matrices, respectively. The forward neural network uses two convolutional layers. And ReLU activation function is used to increase nonlinearity of the network.

$$X_{l+1} = h(x_l) + F(x_l, W_l), \quad (4)$$

$$h(x_l) = W'_l x_l. \quad (5)$$

Gradient of the residual block corresponding to chain rule is calculated by Equation (6), where  $x_1$  denotes input of the entire neural network.  $x_n$  denotes the output parameter closest to the loss function. The chain rule is used to convert the derivation of  $x$  into the derivation and multiplication in Equation (6). As a result, the gradient is difficult to disappear [19].

$$\frac{\partial \text{loss}}{\partial x_1} = \frac{\partial \text{loss}}{\partial x_n} \cdot \frac{\partial x_n}{\partial x_1} = \frac{\partial \text{loss}}{\partial x_n} \cdot \left( 1 + \frac{\partial}{\partial x_n} \sum_{i=1}^{n-1} F(x_i, w_i) \right). \quad (6)$$

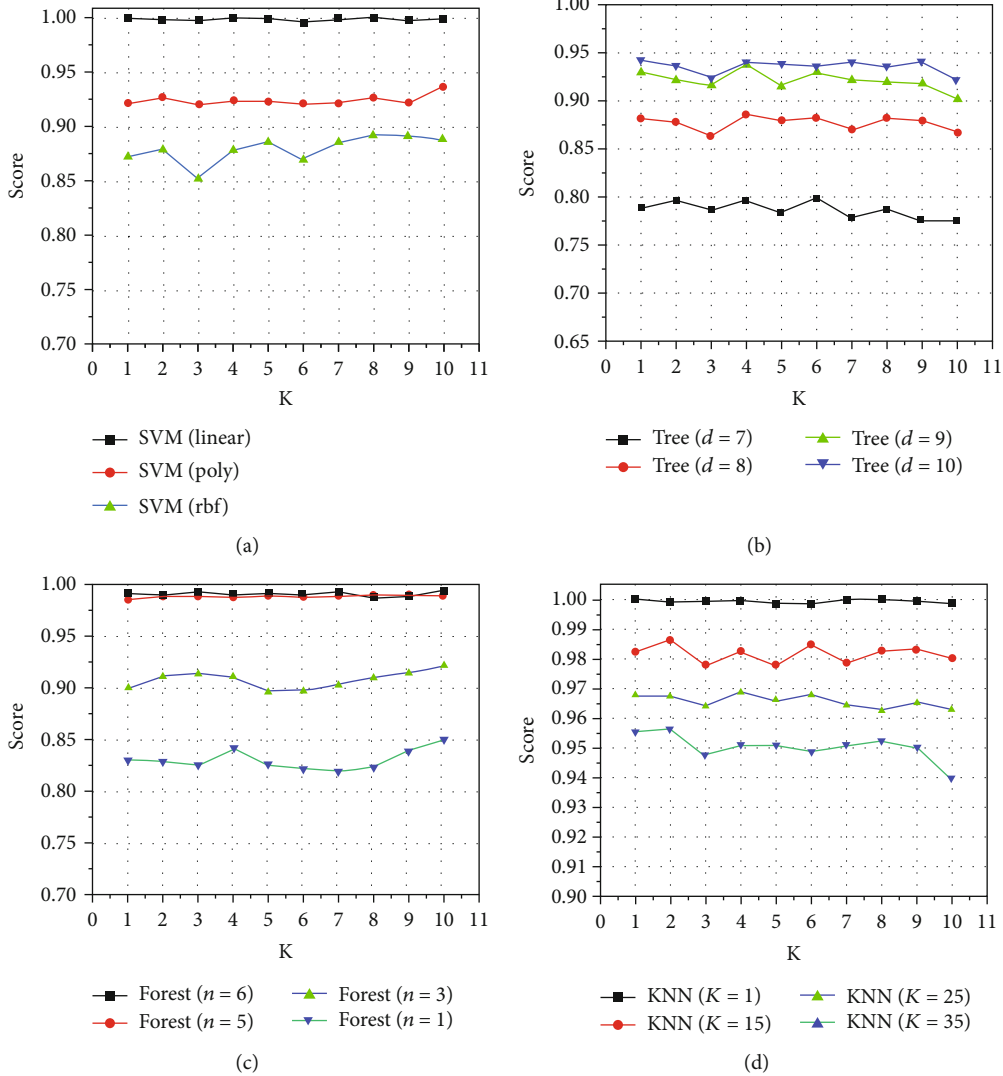


FIGURE 7: The relationship between the scores of different models and value of parameter  $K$ : (a) SVM; (b) DT; (c) RF; (d) KNN.

The feature extraction network in this paper is shown in Figure 3. It consists of residual filtering block, pooling layer, convolutional layer, fully connected (FC) layer, and softmax layer. The feature is flattened before entering the FC layer. RFN includes two processes during training: forward propagation (FP) and BP. The current layer is a convolutional layer during FP. Output  $C^l$  of the layer is calculated by Equation (7), where  $X$  denotes feature vectors of the input.  $W^l$  denotes the weight parameter of the layer  $l$ .  $b^l$  denotes the value of bias.

$$C^l = \text{ReLU}(XW^l + b^l). \quad (7)$$

Finally, data is inputted into the RFN after it is trained. And feature vectors are outputted at the FC layer of the network. The feature vectors outputted in the FC layer are more robust than the output after flattening. Because the dimension of data after flattening is too high. It is difficult to calculate. Then, many neurons after flattening are inactive due to

the ReLU function used in the network. So the data obtained is sparse. Therefore, feature vectors are outputted by the FC layer in this method. The feature vectors of the FC layer are spliced using the feature vectors. And they are calculated by Equation (8), where  $\oplus$  is concatenation and  $C_m^L$  denotes the feature vectors of the  $m$ th channel of the  $L$  layer.

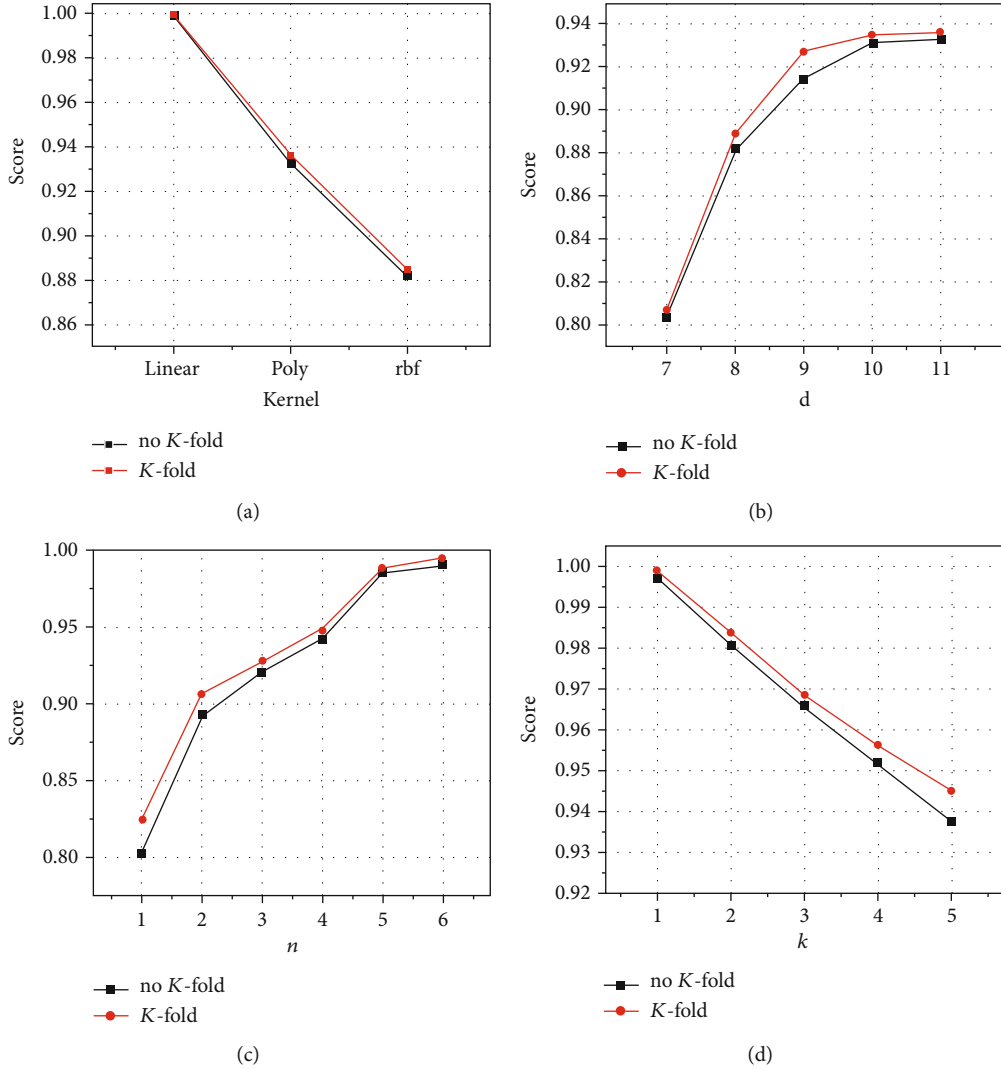
$$C = C_1^L \oplus C_2^L \oplus \dots \oplus C_m^L. \quad (8)$$

The extracted feature maps are shown in Figure 4. The features are extracted through the Gabor filter and multiple convolutional layers. Then, they are concatenated through the FC layer. The features extracted by the RFN are more scientific and robust compared with precision of manual feature extraction.

**2.2. Classification.** Common CNN uses softmax for classification. The output of softmax is the predicted probability of each category. The sum of the probability category is one [20]. Therefore, this method is susceptible to impact of global data

TABLE 1: Comparison of different classifiers.

	Accuracy (Avg)	Recall (Avg)	F1 score (Avg)	Precision (Avg)
SVM (linear)	99.955%	99.958%	99.962%	99.967%
DT ( $d = 10$ )	94.652%	94.633%	94.362%	94.415%
Forest ( $n = 6$ )	98.898%	98.899%	98.852%	98.848%
KNN ( $k = 1$ )	99.926%	99.931%	99.927%	99.923%

FIGURE 8: Comparison of whether to use  $K$ -fold cross-validation method during training: (a) SVM; (b) DT; (c) RF; (d) KNN.

classification for each sample at the same time. SVM is used to replace softmax of the CNN for classification in this paper.

$$\text{Minimize } \left( \frac{1}{2} |w|^2 + c \sum \varepsilon_j \right). \quad (9)$$

The decision boundary of SVM is to solve the maximum margin hyperplane for sample. The parameters can be obtained by Equation (9) in order to obtain the best hyperplane  $w \cdot x + b = 0$  in the second classification, where  $c$  is the penalty coefficient and  $\varepsilon$  is the error [21].

Only the inner product ( $x_a^T x_b$ ) between samples needs to be obtained in the dual problem of linear SVM. And there is no need to specify a nonlinear transformation [22]. Therefore, the inner product of samples needs to be replaced by kernel function in a nonlinear classification problem. There is a mapping from the input space to the feature space as  $\phi(x)$  [23]. The kernel Equation (10) is satisfied for  $(x_a, x_b)$  in any input space. Therefore, it is used to replace the mapped sample inner product [24]. SVM classification can be obtained by Equation (11), where  $N$  is number of features. Labels of the  $i$ th feature vectors  $a_i^*$  and  $b_i^*$  are

TABLE 2: Comparison with different features using SVM.

	Accuracy (Avg)	Recall (Avg)	F1 score (Avg)	Precision (Avg)
RFN-SVM	99.955%	99.958%	99.962%	99.967%
RGB-SVM	92.415%	92.296%	92.188%	92.772%
CNN-SVM	97.166%	96.825%	96.944%	97.472%

TABLE 3: Comparison with different features using DT.

	Accuracy (Avg)	Recall (Avg)	F1 score (Avg)	Precision (Avg)
RFN-DT	94.652%	94.633%	94.362%	94.415%
RGB-DT	87.555%	87.276%	87.136%	87.391%
CNN-DT	93.636%	93.454%	93.375%	93.476%

TABLE 4: Comparison with different features using RF.

	Accuracy (Avg)	Recall (Avg)	F1 score (Avg)	Precision (Avg)
RFN-RF	98.898%	98.899%	98.852%	98.848%
RGB-RF	90.015%	89.902%	89.744%	89.893%
CNN-RF	97.948%	97.914%	97.865%	97.945%

TABLE 5: Comparison with different features using KNN.

	Accuracy (Avg)	Recall (Avg)	F1 score (Avg)	Precision (Avg)
RFN-KNN	99.926%	99.931%	99.927%	99.923%
RGB-KNN	92.553%	92.464%	92.368%	92.613%
CNN-KNN	98.677%	98.565%	98.531%	98.569%

related parameters of the  $i$ th feature vectors. The mapped sample inner product is replaced by the kernel function, which can effectively solve the linear inseparable sample classification problem [25].

$$K(x_a, x_b) = \phi(x_a)^T \phi(x_b). \quad (10)$$

The features extracted from the previous section are used to train the SVM. The 10-fold cross-validation method is added during training. This method divides the data into 10 equally. One of the data is selected as the test set during each training. And the rest are used for the training set. The above operation is repeated ten times. And the selected test set is not repeated each time. The  $K$ -fold cross-validation method is used to reduce the occasionality and improve the generalization of the model.

$$f(x) = \text{sign} \left( \sum_{i=1}^N a_i^* y_i K(x, x_i) + b^* \right). \quad (11)$$

### 3. Experiment and Analysis

The operating system of this experiment is Windows 10. The GPU device is NVIDIA GEFORCE GTX 960M. The framework is PyTorch 1.6.

**3.1. Data Processing.** The data in this paper comes from the public dataset Fruits-360. It has 22688 pictures of 131 classes. Training set and testing set are divided in a ratio of 7 : 3. They are 15882 and 6806 pictures, respectively. The size of the image is scaled to  $50 \times 50$ . And one of the batch data is shown in Figure 5.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (12)$$

$$\text{F1score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}, \quad (13)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (14)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (15)$$

**3.2. Feature Extraction.** Feature extraction is an important step in the fruit classification process. Features from input Fruits-360 images are extracted by using RFN. Firstly, RFN proposed in this paper is trained. The dimension of the input image is  $50 \times 50 \times 3$ . And SGD is used as an optimizer. In addition, learning rate is set to 0.001. The training result of the network is shown in Figure 6. The horizontal axis and the vertical axis are epoch and accuracy, respectively. The



TABLE 6: Classes and subclasses in the dataset.

Classes	Subclasses
Apple	Apple Braeburn, Apple Crimson Snow, Apple Golden 1, Apple Golden 2, Apple Golden 3, Apple Granny Smith, Apple Pink Lady, Apple Red 1, Apple Red 2, Apple Red 3, Apple Red Delicious, Apple Red Yellow 1, Apple Red Yellow 2
Cherry	Cherry 1, Cherry 2, Cherry Rainier, Cherry Wax Black, Cherry Wax Red, Cherry Wax Yellow
Grape	Grape Blue, Grape Pink, Grape White, Grape White 2, Grape White 3, Grape White 4, Grapefruit Pink, Grapefruit White
Pear	Pear, Pear 2, Pear Abate, Pear Forelle, Pear Kaiser, Pear Monster, Pear Red, Pear Stone, Pear Williams
Potato	Potato Red, Potato Red Washed, Potato Sweet, Potato White
Tomato	Tomato 1, Tomato 2, Tomato 3, Tomato 4, Tomato Cherry Red, Tomato Heart, Tomato Maroon, Tomato not Ripened, Tomato Yellow

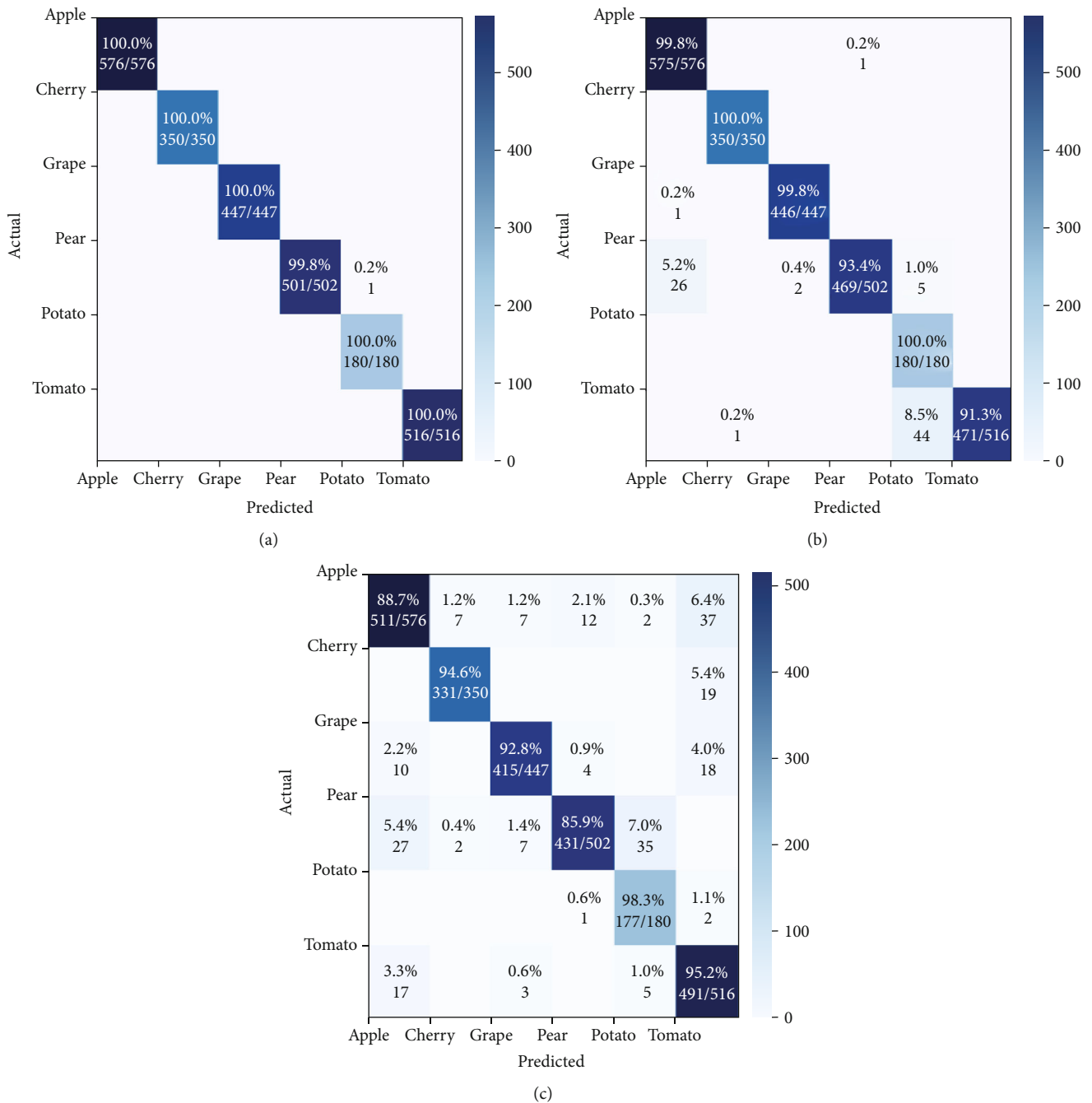


FIGURE 9: Confusion matrix is used to evaluate different methods: (a) RFN with SVM; (b) CNN with SVM; (c) RGB with SVM.

accuracy of RFN is 97%. Therefore, the features extracted by the model have certain reliability. Then, the data is inputted to the RFN. And feature vectors are outputted in the FC layer. The feature vectors are used for SVM classification.

**3.3. Results and Discussion.** The feature vectors are extracted from the RFN to train the SVM. And the indicators of evaluation are accuracy, precision, recall, and F1 score. They are calculated by Equations (12), (13), (14), and (15), respectively. True positive (TP) denotes the positive sample predicted by the model to be positive. True negative (TN) denotes the negative sample predicted to be negative by the model. False positive (FP) denotes the negative sample predicted to be positive by the model. False negative (FN) denotes the positive sample predicted to be negative by the model.

Accuracy reflects the ability of the model to judge the entire sample. Precision reflects proportion of the TP samples in the positive samples judged by the classifier. Recall reflects proportion of the positive samples correctly judged by the classifier in the total positive samples proportion. F1 score is the harmonic average of precision and recall.

The feature vectors extracted by the RFN are used to train four ML models. Figure 7 shows the relationship between the model scores of the ML models under different parameters and value of parameter  $K$ . Three different kernel functions rbf, poly, and linear are used by SVM for comparison. SVM with linear kernel function is the highest score of one. Comparing Decision Trees (DT) under different maximum depths ( $d$ ), DT stops splitting when its depth reaches the specified maximum depth threshold. The highest score for a DT with  $d = 10$  is 0.9411. Comparison of Random Forest (RF) under different number of DT ( $n$ ). The highest score for a RF with  $n = 6$  is 0.9941. Compared to KNN under different  $K$  values, the value of  $K$  means that the number of samples closed to the training sample from the prediction sample is  $K$ . The highest score for a KNN with  $K = 1$  is one. It can be concluded that the features extracted by the RFN have better robustness.

Then, the classifiers with the highest scores for each model in the above were compared. The results are shown in Table 1. The SVM with linear kernel has the best effect. Accuracy is 99.955%. Recall is 99.958%. F1score is 99.962%, and precision is 99.997%.

The  $K$ -fold cross-validation method is used for the training of four ML models. Figure 8 shows the comparison between four ML models using 10-fold cross-validation and without  $K$ -fold cross-validation. The horizontal axis denotes the different parameters of the classifier. And the vertical axis denotes the score of model performance. The red and black lines are the methods of adding and not adding  $K$ -fold cross-validation during training, respectively. It can be concluded that the  $K$ -fold cross-validation method is used to reduce the occasionality.

In the experiments, three features are compared. They are the features extracted by the RFN, the features extracted by common CNN, and the RGB features. The above three features are used to train four ML models. The results are shown in Tables 2–5.

The classification effect of the features extracted by the RFN is better than that by the other two methods. The results show that the classification method using the RFN combined with SVM has the best effect. Accuracy is 99.955%. Recall is 99.958%. F1score is 99.962%. And precision is 99.967%.

In addition, the proposed method is also evaluated to classify similar classes of category Apple, Cherry, Grape, Pear, Potato and Tomato as shown in Table 6. The results are shown in the confusion matrix in Figure 9. The confusion matrix shows the relationship between the predicted value of the classifier and true value. So the confusion matrix is used to evaluate the performance of the method. The best SVM classification effect is to use the features extracted by RFN. Accuracy is 99.966%. The method of RFN and SVM proposed in this paper for fruit classification has a good classification effect.

## 4. Conclusion

A method of RFN for smart community robot fruit classification is proposed in this paper. The features are more robust that are extracted by the RFN which combines the Gabor filter and the residual block. Four ML models (SVM, RF, DT, KNN) are used to test the features extracted by the RFN. And SVM is used to replace the softmax classification of the common CNN. It improves the classification effect of the model. And the  $K$ -fold cross-validation method is used in training. It can improve the accuracy of the model and reduce the occasionality by random classification. The method proposed in this paper is compared with other methods. The results show that the accuracy of the proposed method on the Fruits-360 dataset is 99.955%. It is an improvement compared with the original result of author. In addition, the accuracy among the 6 categories of fruits that are difficult to distinguish is 99.966%. The method for smart community robot fruit classification proposed in this paper can achieve good results. This method can replace the traditional method of artificial feature extraction for classification. And it can also be further extended to other fields.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This study was supported by Science and Technology Program of Guangzhou (No. 2019050001), project of Shenzhen Science and Technology Innovation Committee (JCYJ20190809145407809), project of Shenzhen Institute of Information Technology School-level Innovative Scientific Research Team (TD2020E001), Program for Guangdong Innovative and Entrepreneurial Teams (No. 2019BT02C241), Program for Chang Jiang Scholars and



Innovative Research Teams in Universities (No. IRT\_17R40), Guangdong Provincial Key Laboratory of Optical Information Materials and Technology (No. 2017B030301007), Guangzhou Key Laboratory of Electronic Paper Displays Materials and Devices (201705030007), and the 111 Project.

## References

- [1] E. Guzmán, V. Baeten, J. A. F. Pierna, and J. A. García-Mesa, "Infrared machine vision system for the automatic detection of olive fruit quality," *Talanta*, vol. 116, no. 22, pp. 894–898, 2013.
- [2] X. Liu, X. B. Zhai, W. Lu, and C. Wu, "QoS-guarantee resource allocation for multibeam satellite industrial Internet of Things with NOMA," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052–2061, 2021.
- [3] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5379–5388, 2020.
- [4] S. Arivazhagan, R. N. Shebiah, S. S. Nidhyandhan, and L. Ganesan, "Fruit recognition using color and texture features," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 1, no. 2, pp. 90–94, 2010.
- [5] M. George, "Multiple fruit and vegetable sorting system using machine vision," *International Journal of Advancements in Computing Technology*, vol. 6, no. 1, article 1000142, 2015.
- [6] H. L. Kuang, L. L. H. Chan, and H. Yan, "Multi-class fruit detection based on multiple color channels," in *2015 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp. 1–7, Guangzhou, China, 2015.
- [7] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," *Computers and Electronics in Agriculture*, vol. 163, article 104846, 2019.
- [8] M. Momeny, A. Jahanbakhshi, K. Jafarnejhad, and Y. D. Zhang, "Accurate classification of cherry fruit using deep CNN based on hybrid pooling approach," *Postharvest Biology and Technology*, vol. 166, p. 111204, 2020.
- [9] F. Gao, L. Fu, X. Zhang et al., "Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN," *Computers and Electronics in Agriculture*, vol. 176, article 105634, 2020.
- [10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [11] J. C. Pérez, M. Alfara, G. Jeanneret et al., "Gabor layers enhance network robustness," in *Computer Vision – ECCV 2020. ECCV 2020*, vol. 12354 of Lecture Notes in Computer Science, , pp. 450–466, Springer, 2020.
- [12] J. P. Jones and L. A. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *Journal of Neurophysiology*, vol. 58, no. 6, pp. 1233–1258, 1987.
- [13] X. Liu and X. Zhang, "Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5971–5980, 2019.
- [14] J. Yang, L. Liu, T. Jiang, and Y. Fan, "A modified Gabor filter design method for fingerprint image enhancement," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1805–1817, 2003.
- [15] L. Zhuang, F. Da, S. Gai, and M. Li, "Transformation invariant Gabor convolutional networks," *Signal, Image and Video Processing*, vol. 14, no. 7, pp. 1413–1420, 2020.
- [16] X. Liu, X. Zhang, M. Jia, L. Fan, W. Lu, and X. Zhai, "5G-based green broadband communication system design with simultaneous wireless information and power transfer," *Physical Communication*, vol. 28, pp. 130–137, 2018.
- [17] Y. Jiang, Y. Li, and H. Zhang, "Hyperspectral image classification based on 3D separable ResNet and transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1949–1953, 2019.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.
- [19] J. Zhang, X. Wei, C. Che, Q. Zhang, and X. Wei, "Breast cancer histopathological image classification based on convolutional neural networks," *Journal of Medical Imaging and Health Informatics*, vol. 9, no. 4, pp. 735–743, 2019.
- [20] W. Lu, Y. Gong, X. Liu, J. Wu, and H. Peng, "Collaborative energy and information transfer in green wireless sensor networks for smart cities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1585–1593, 2018.
- [21] M. Lapin, M. Hein, and B. Schiele, "Learning using privileged information: SVM+ and weighted SVM," *Neural Networks*, vol. 53, pp. 95–108, 2014.
- [22] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [23] X. Li, J. Cervantes, and W. Yu, "A novel SVM classification method for large data sets," in *2010 IEEE International Conference on Granular Computing*, pp. 297–302, San Jose, CA, USA, 2010.
- [24] F. Friedrichs and C. Igel, "Evolutionary tuning of multiple SVM parameters," *Artificial Neural Networks*, vol. 64, pp. 107–117, 2005.
- [25] J. Cervantes, F. Garcia-Lamont, L. Rodriguez, A. López, J. R. Castilla, and A. Trueba, "PSO-based method for SVM classification on skewed data sets," *Neurocomputing*, vol. 228, pp. 187–197, 2017.