Application of Feature Extraction through Convolution Neural Networks and SVM Classifier for Robust Grading of Apples

Yuan CAI[1], Clarence W. DE SILVA[2], Bing LI[1], Liqun WANG[1], Ziwen WANG[1]

(1. *Harbin Institute of Technology at Shenzhen*, *Shenzhen* 518000;
2. *The University of British Columbia*, *Vancouver*, *BC*, *Canada V6T* 1Z4)

Abstract: This paper proposes a novel grading method of apples, in an automated grading device that uses convolutional neural networks to extract the size, color, texture, and roundness of an apple. The developed machine learning method uses the ability of learning representative features by means of a convolutional neural network (CNN), to determine suitable features of apples for the grading process. This information is fed into a one-to-one classifier that uses a support vector machine (SVM), instead of the softmax output layer of the CNN. In this manner, Yantai apples with similar shapes and low discrimination are graded using four different approaches. The fusion model using both CNN and SVM classifiers is much more accurate than the simple k-nearest neighbor (KNN), SVM, and CNN model when used separately for grading, and the learning ability and the generalization ability of the model is correspondingly increased by the combined method. Grading tests are carried out using the automated grading device that is developed in the present work. It is verified that the actual effect of apple grading using the combined CNN-SVM model is fast and accurate, which greatly reduces the manpower and labor costs of manual grading, and has important commercial prospects.

Key words: Apple Grading, k-nearest Neighbour Method, Convolutional Neural Network, Support Vector Machine, Machine Learning.

1 Introduction

At present apple grading is primarily done manually, relying on the human vision to distinguish such features as the color, shape, size and defects in apples. This manual sorting is subjective, non-repeatable, and slow. This together with other human factors such as physical fatigue of this labor intensive practice makes the grading accuracy rather poor in general. Furthermore, the seasonal nature of the fruit harvest makes it difficult and costly to retain trained graders year-round. When the harvesting season of apples arrives, it is difficult to train new grades. It follows that automated, high speed, and efficient apple grading technology is favored by fruit farmers. Clearly, such automated sorting technology can overcome the problem of the traditional, manual grading of apples. The automated sorting technology can maintain continuity, repeatability, accuracy, and uniformity in apple grading. In view of the practicality of automated sorting technologies, more and more practical engineering projects are devoted to building an intelligent, reliable, fast, and efficient apple grading system. Sofu et al.[1] proposed an automated apple grading system that was divided into different categories based on the weight, color, and size of apples. The collected and processed images showed the color, size, projected area and the defect area of an apple, and the apples were graded by the information gain ratio C4.5 classifier. Gaikwad et al.[2] proposed a grading system that used a specific threshold and performed feature reduction to extract the RGB color and texture features of fruit. A multi-layer backpropagation neural network (BPNN) classifier was used. Kavdir et al.[3] studied different technical treatments of apple grading processing. They defined features such as color, shape defects, hardness, weight, and the blush ratio. Classification techniques such as decision rules, neural networks, decision trees and multilayer perceptron were studied. Piao et al.[4] used near-infrared spectroscopy and three feature extraction methods: principal component analysis, fisher discriminant analysis and non-correlation discriminant transformation, and used the k-nearest neighbor grading algorithm to an establish apple grading model. Wang et al.[5] proposed a grading method for apples based on the measurement of size, color and surface defects. There, a fuzzy logic algorithm was proposed to grade apples. Yang et al.[6] calculated the gray value of the image when distinguishing calyx, stem and defect of apple, adopted the method of image feature fusion, and proposed the clustering algorithm of unsupervised learning k-means to achieve apple classification. Pan et al.[7] proposed a near-infrared hyperspectral imaging technique to capture apple images, linear discriminant analysis (LDA), and gradient boosting decision tree (GBDT) models to grade apples. In all above methods, the traditional feature extraction is used and the resolution feature needs to be manually extracted, and hence the pre-processing of the data is complicated and slow. But the proposed technique includes one of the prominent features. It shows advantages over all previous methods. Specifically, the present method uses the convolutional neural network in machine learning to extract features automatically. The model has a strong generalization ability and feature representation ability. This kind of machine learning-based methods uses convolutional neural networks for feature extraction. This classification method has high accuracy and good generalization ability, particularly in the case of big data. The convolutional neural network-based extraction feature, as proposed in the present paper, uses the SVM classifier to complete the automated apple grading, where the apples are on a moving conveyor belt. The automated grading device of the present work consists of a controller on single chip, an image processing unit on a personal computer (PC), and a conveying mechanism with conveyor belt and an actuator (motor). Fig. 1 shows the basic grading schematic diagram of an automated grading device.

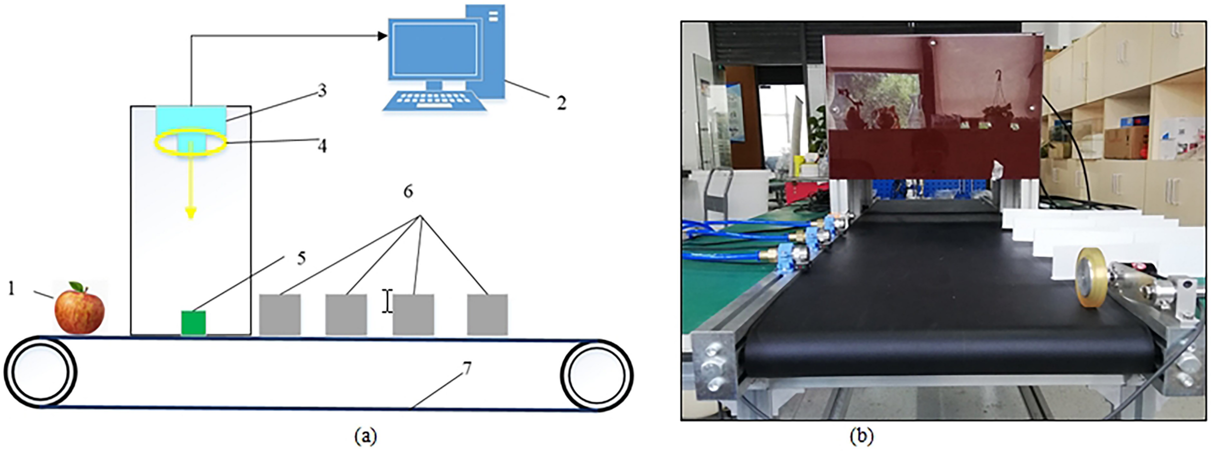


Fig. 1 (a) Schematic diagram of the grading system; (b) Grading system built in the present project. 1-apple sample, 2-computer, 3-CCD camera, 4-light source, 5- photoelectric sensor, 6-high pressure jets, 7-Motor-driven conveyor.

Fig. 2 shows a flow chart of the present automated grading system for apples. The steps of the grading process are as follows: First, start the grading software program, and the PC waits for the CCD camera to take an image of an apple. After that, the apples are placed continuously at the feeding side of the conveyor belt. The images of the apples are collected as they pass through the image acquisition device. The image acquisition device is composed of a CCD camera, a photoelectric sensor and a ring light source. The photoelectric sensor is connected to the input end of the single chip microcomputer and the system receives the signal of the photoelectric sensor through the single chip microcomputer. After taking a photo by the CCD camera, the image is stored in the corresponding folder of the PC. At the same time, the program analyzes the image to determine the grade corresponding to the apple. The speed of the conveyor belt is governed by its drive motor, and is controlled by adjusting the frequency of the inverter of the motor controller. While waiting for the apple to reach the corresponding grading port (pneumatic jet), the photoelectric sensor detects the apple. When an apple reaches the proper jet, the single-chip microcomputer controls the cylinder of the jet to perform its action, pushing the apple into the corresponding bin. Rest of the paper is organized as follows: Section 2 presents the theoretical analysis. Section 3 describes the measurement method and the proposed grading process. The conclusion of the paper is given in Section 4.

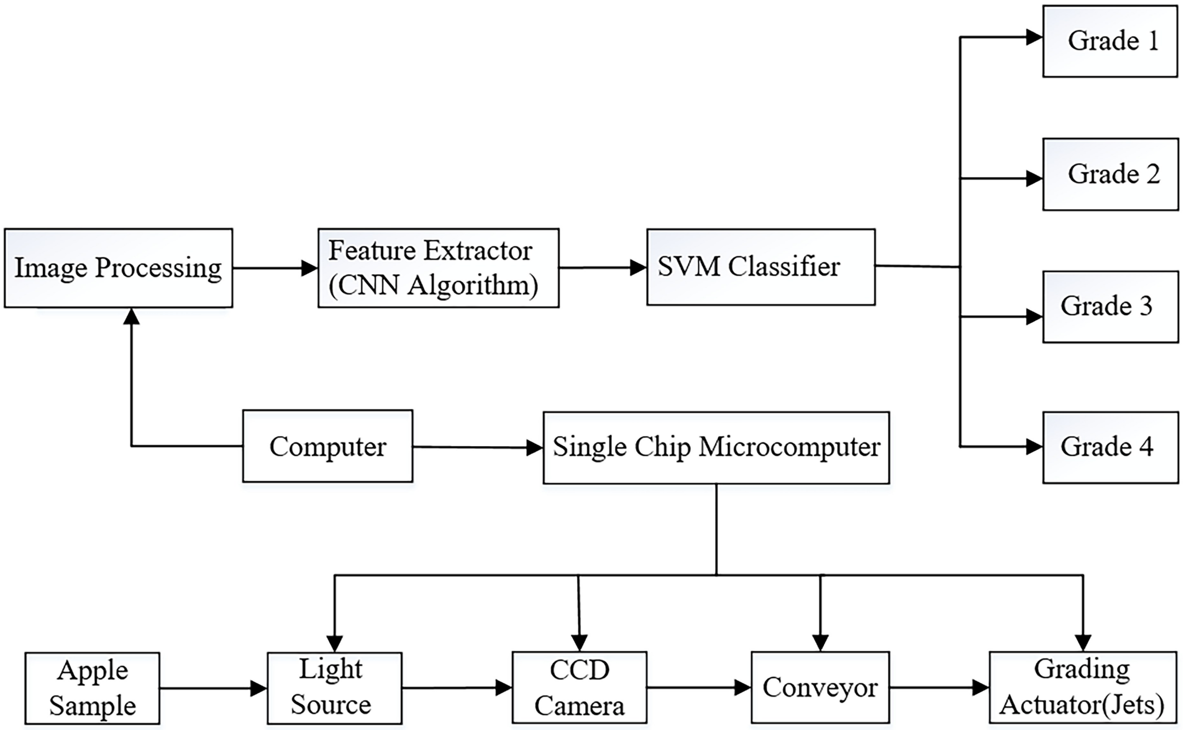


Fig. 2 Flow chart of the grading system

2 Theoretical Analysis

2.1 Convolutional Neural Network Feature Extraction

The most notable achievement of convolutional neural networks is their feature extraction capabilities. This is exploited in the present work. After the original apple image is cropped, enhanced, and normalized, the convolutional neural network enters a set of tensors with a dimension of 100 × 100 × 3 and the input is a picture with red, green, and blue (RGB) channels. The input data is an apple image that converts the image into an array and normalizes the tensor. The preprocessed image serves as the data for the input layer for the convolutional neural network.

2.1.1 Convolution layer

The convolution layer mainly performs feature extraction, i.e., the convolution kernel is convoluted with the input image of the previous layer, and then outputs a set of feature maps. The convolution kernel is equivalent to a feature extractor. The expression for the convolution operation is,

$$x_j^l = f(\sum_{i \in M_j} x_i^{l-1} * w_{ij}^l + b_j) \qquad (1)$$

where, x_j^lrepresents the j -th feature map of the l-th layer, l represents the current number of layers, x_i^{l-1}represents the i -th feature map of the l − 1-th layer,

2.1.2 Pooling layer

The pooling operation in CNN is essentially a process of downsampling. The features of different positions of the x_i^{l-1}represents the i -th feature map of the l − 1th layer, w_{ij}^l represents the weight matrix of the convolution kernel , $b_j$represents the offset term corresponding to the eigenvalue, $*$ represents the convolution operation operator, M_j represents the number of the feature map of the l − 1th layer, and$f$ represents the nonlinear activation function.

A powerful representation of convolutional neural networks is reflected in the nonlinearity of its activation function. The weight matrix is the same for all inputs at different locations, which is the convolution layer " weight sharing" feature [8]. In addition to this, a bias is usually incorporated. Common activation functions are sigmoid function, tanh function, and rectified linear unit (ReLU). The ReLU function solves the gradient saturation effect and contributes to the convergence of the stochastic gradient descent method, [9] which can prevent the over-fitting [10] problems to some extent. The paper uses ReLU as the activation function, and its expression is,

$$f(z_i) = \max\{0, z_i\} = \begin{cases} 0, & z_i < 0 \\ z_i, & z_i \geq 0 \end{cases} \qquad (2)$$

where, $z_i = \sum_{i \in M_j} x_i^{l-1} * w_{ij}^l + b_j$

feature map are extracted twice, and the feature invariance is achieved by reducing the resolution of the feature image [11]. This operation focuses on the characteristics of the image. In CNN, the pooling operation can achieve a dimensional reduction in the spatial scope. Pooling operations can also reduce training parameters and the amount of calculations, and can prevent overfitting (over-training). In order to improve the learning performance of the pooling layer, the parametersβ and b are used to calculate,

$$x_j^l = \beta_j^l downsample(x_j^{l-1}) + b_j^l \qquad (3)$$

where, x_j^l represents the$j$ -th feature map of the l -th layer, $\beta_j^l$and b_j^lrepresent the pooling layer parameters, $downsample(.)$ represents the pooling function, and x_j^{l-1}represents the j -th feature map of the l-1-th layer .

2.1.3 Fully connected layer

The fully connected layer of a CNN connects all the nodes of the upper layer with all the nodes of the next layer, and assembles the high-dimensional features into one-dimensional feature vectors. The fully connected structure reflects the mapping relationship between the extracted features and the output category labels. The loss function of the fully connected layer is used to predict the error between the actual output and the predicted output. Softmax regression is a generalized form of logistic regression, and is often used in multi-classification. The Softmax layer appears behind the fully connected layer in a CNN, and the number of nodes in the output layer must match the actual total number of grading classes. The Softmax function converts linear predictions to category probabilities, and it is defined as [12],

$$P(t^{(i)} = j \mid x^{(i)}; W^{(L)}) = \frac{1}{\sum_{l=1}^{k} e^{(W_l(L))^{T} x(i)}} \begin{bmatrix} e^{(W_1(L))^{T} x(i)} \\ e^{(W_2(L))^{T} x(i)} \\ \cdots \\ e^{(W_k(L))^{T} x(i)} \end{bmatrix} \qquad (4)$$

where, $t^{(i)} \in \{1,2,\dots,k\}$ is the label set; $x^{(i)} \in R^n$is the sample set, and $j = 1,2,\dots,k$, $W^{(L)} = [W_1{}^L, W_2{}^L, \dots, W_k{}^L]$ are the weights of the classifier. P is the probability that $t^{(i)}$ is the output in the case of class j , and $1/\sum_{l=1}^{k} e^{(W_l(L))^{T}x(i)}$ is the normalization process in computing probability.

The Softmax regression loss function is defined as [12],

$$J(W^{(L)}) = \\ -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{t^{(i)} = j\} \log\frac{e^{(W_j(L))^{T}x(i)}}{\sum_{l=1}^{k} e^{(W_1(L))^{T}x(i)}}\right] \quad (5)$$

where, m is the total number of samples. {.} is the indicator function, where if the condition is false, 0 is returned; otherwise, 1 is returned.

2.1.4 Backpropagation algorithm

The basic training mechanism of a convolutional neural network contains two main processes: forward learning and backpropagation. The forward learning process first performs feature learning in a bottom up manner through the input layer, and finally outputs the predicted result at the output layer, which is a process of learning and classification using features. The reverse adjustment process is a parameter adjustment process. It compares the data with the label and the result of the forward learning, and transmits the error between the two, from the top to the bottom. The CNN parameters are adjusted and fine-tuned in this manner.

The backpropagation algorithm [13] uses the chained derivative to calculate the partial derivative of the loss function for each weight, and then updates the weight according to the gradient descent strategy. In order to predict the error between the actual output and the predicted output, a loss function is used. For this purpose, the cross-entropy function is used in the present paper, and is defined as,

$$J(f(z_i), \hat{y}_i) = -\sum_{i=1}^{m}[\hat{y}_i \log f(z_i) + \\ (1-\hat{y}_i)\log(1-f(z_i))] \quad (6)$$

where, $f(z_i)$ is the actual output, $\hat{y}_i$ is the predicted output, and m is the number of samples. The weight w and the bias$b$ are updated as follows according to the chain derivation formula:

$$w_i = w_i - \eta\frac{\partial}{\partial w}J(w,b) \quad (7)$$

$$b_i = b_i - \eta\frac{\partial}{\partial b}J(w,b) \quad (8)$$

where, η is the learning rate.

2.2 Support Vector Machine Classifier

The support vector machine (SVM) [14] is a method of supervised learning based on generalized linear grading of sample data. The condition of the decision boundary is that the distance from the point closest to the dividing line (data separating line for classification) to the dividing line itself is the largest. The point closest to the dividing line is called the support vector. In fact, in the SVM algorithm, only the support vector is involved in the calculation, and the non-support vector does not participate in the calculation. In the present paper, the SVM classifier uses the one-versus-one (OVO) method to design a SVM classifier, so k samples are needed to design$k \times (k-1)/2$ separate binary classifiers.

The i -th classifier is used to distinguish whether each sample can be classified as the i -th class. When training this classifier, the labels need to be rearranged into " i -th class label" and " non- i -th class label". When an unknown sample is classified, the decision is finally made by voting, and the category with the most votes is the category of the unknown sample. The model for the optimization problem associated with SVM is,

$$\begin{aligned} &\min_{w,b,\xi} imize\ \frac{1}{2}w^{T}w + c\sum_{i=1}^{m}\xi_i \\ &subjecty_i(w^{T}\varphi(x_i) + b) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \quad (9)$$

where, ξ_i is the slack variable, c is the hyperparameter, and $\varphi(x_i)$ is a mapping based on a kernel function:

$$\kappa(x_i, x_j) = <\varphi(x_i), \varphi(x_j)> \quad (10)$$

In practical applications, the inner product of the original sample map in the feature space can be replaced by a kernel function [15]. For the SVM classifier, a linear kernel function is used for linearly separable data and a Gaussian kernel function may be used for linearly non-separable data. The linear kernel function expression is given by,

$$\kappa(x_i, x_j) = x_i{}^T x_j \tag{11}$$

The Gaussian kernel function is given by,

$$\kappa(x_i, x_j) = \exp(-\frac{\| x_i - x_j \|^2}{2\sigma^2}) \tag{12}$$

The Lagrangian multiplier method is used to obtain its "dual problem" [16] and then Sequential Minimal Optimization (SMO) [17] is used to solve the optimal solution of the problem. The advantage of this is that the dual problem often makes the problem easier to solve; the kernel function is introduced and then extended to the nonlinear classification problem.

2.3 *k*-Nearest Neighbor

The *k*-nearest neighbor (KNN) classification algorithm is a relatively simple algorithm compared to SVM and CNN. The algorithm sorts the training set by measuring the distance between different samples, and then classifies according to the nearest *k*neighbors. The KNN algorithm is suitable for multi-classification problems, and the algorithm is simple and easy to understand. However, there are two shortcomings: first, when the sample is extremely unbalanced, a large number of samples with a large number of classifications become dominant, resulting in a lower accuracy of classification. Second, the KNN algorithm needs to calculate the distance of each sample to determine the *k* value, which is computationally intensive. In the KNN algorithm, there are three main steps: calculating the distances, selecting the neighbors, and making the decisions. In the present work, the *k*-nearest neighbor classifier from Scikit-Learn is used.

2.3.1 Calculating distance

The distance between the measured value and each data in the sample set has to be computed. The feature space *X*is an *n*-dimensional real number vector spaceR^n , which is defined as the Minkowski metrics, and the corresponding formula is,

$$L_p = (x_i, x_j) = (\sum_{i=1}^{n} | x_i^{(l)} - x_j^{(l)} |^p) \frac{1}{p} \tag{13}$$

where, $x_i, x_j \in X$, $x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)})^T$, $x_j = (x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(n)})^T$ In this formula, when$p = 1$, we have the Euclidean distance; and when$p = 2$, we have the Manhattan distance. The Euclidean distance is used in the present work.

2.3.2 Selecting neighbors

Here, the steps are sorting the calculated distances and selecting the *k* nearest sample points. Choosing the right *k* value is especially important for the grading performance of the algorithms. If the *k* value is too low, the classifier is susceptible to noise in the training data; if the *k* value is too high, the classifier may misgrade the test sample. In the present work, cross-validation is used to choose the *k* value, and $k = 5$ was determined on this basis.

2.3.3 Making decisions

After obtaining the list of neighbors, the test sample is classified by the majority vote method [18]. In the majority vote, each neighbor has the same effect on the grading, which makes the algorithm sensitive to the choice of *k* .

2.4 Evaluating Classifier Performance

Classification (grading) is a common task in machine learning. Common evaluation indicators for grading tasks include confusion matrix, accuracy, precision, recall and F1_Score. P in the confusion matrix means positive, i.e., positive samples; and N means negative, i.e., negative samples. TP represents the number of samples that are actually positively predicted; FP represents the number of samples that are actually negative but are predicted to be positive; and TN and FN are similar [19]. Using this notation, we can define the following performance metrics for the classifier [20]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$\text{F1\_Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (17)$$

The accuracy rate is the ratio of the correctly classified samples to the total number of samples. The precision rate indicates the proportion of the samples with the correct classification as a proportion of the positive sample determined by the classifier. The recall rate indicates the proportion of the correct number of samples to the true positive samples. F1_score is the harmonic mean of precision and recall. Precision and Recall are contradictory, and uniform indicators will have both high. F1_Score is a combination of Precision and Recall. The larger the F1_Score, the better the robustness of the model.

3. Experiments and Discussion

3.1 Experimental Data

In this paper, the same types of Shandong Yantai apples of similar origin are selected to carry out the apple grading experiments. Due to the similarity of their appearance, Yantai apples have little difference in the ruddy color of apples, which greatly increases the difficulty in manual grading.Experiments are carried out to study the performance of the developed prototype system for apple grading. Based on the experiments, the features extracted by CNN, SVM are selected as inputs to the classifier, which is a grading method of the supervised learning category, and developed in the present work. In the beginning of the experiment, pictures of apples were taken manually for training the machine learning process. After that, the apples were divided into 4 categories in advance, and the grade of each category of apples was determined based on the weight, red coloration, texture characteristics and the roundness. The characteristics of these four groups of apples, categorized as Grade 1 to Grade 4, are given in Table 1:

Table 1 The governing characteristics of four grades of apple.

Apple Quality Grade	Number of Apples	Weight Feature	Red Stain Percentage Feature	Shape Roundness Feature
Grade 1	800	230~250g	≥ 90%	≥ 85%
Grade 2	800	200~230g	80%~90%	80%~85%
Grade 3	800	180~200g	55%~80%	70%~80%
Grade 4	800	150~180g	≤55%	≤70%

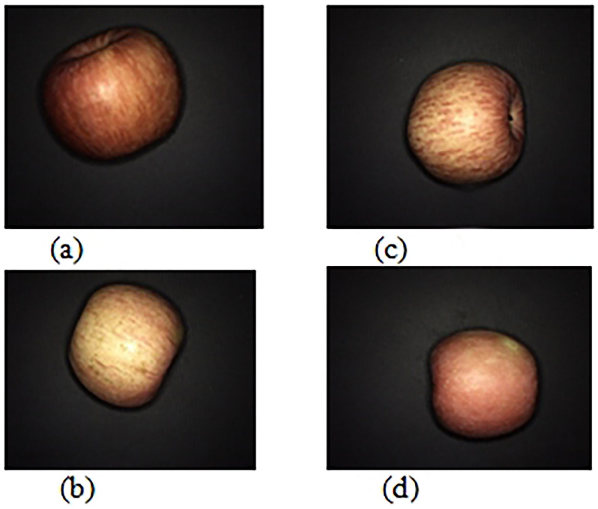


Fig. 3 Sample images of the four groups of apples used in the experimental studies.

3.2 Experimental CNN_SVM model

In the present experiment, we use CNN to extract features in the images, and SVM as a classifier to grade the apples using those features, through the combined model of CNN_SVM. Through an algorithm of deep learning, CNN automatically extracts the features of apples, using its convolution layer. SVM is used for the final classification of apples. It is easy to adjust the parameters in this model. It has strong generalization abilities in comparison with traditional classification methods, and is very suitable for the classification of small samples. The data in the present experiments is not extensive; so, using SVM instead of softmax can avoid the over-fitting problem in CNN to some extent. The description of the structure of CNN_SVM is given now.

In order to avoid the problem of gradient dispersion and gradient explosion in the convolution operations, ReLU function is used as the activation function. The sparse expression of neural networks is due to the unilateral inhibition of ReLU as well. Before entering the pooling layer, the data needs to be batching normalized; i.e., the entire batch data distribution is forced to have 0 mean and a standard deviation of 1, which reduces the fitting ability of the model and enhances the generalization ability of the model. After batch normalization, the data enters the max pooling layer. The size of the max pooling layer is 2 * 2, and the step size is 2. The size of the fully connected layer is 1024. Since the fully connected layer is connected to the dropout layer, the risk of overfitting is reduced and the generalization ability is enhanced. After the sixth convolution operation and pooling, the output is provided to the fully connected layer, which generates the features. These features are expanded from the multi-dimensional vector into a one-dimensional vector. The used CNN optimizer is Adam. The learning rate is 0.0001 and the loss function is categorical _ crossentropy. Finally, the features extracted by the fully connected layer are sent to the SVM classifier for classification. Fig. 4 shows the basic structure of the CNN_SVM model. Table 2 presents the parameters of the six convolutional layers. In the apple grading task, random subsamples with verification are used. Data sets are randomly divided into subsets of training, validation, and testing. Table 3 presents information on the apple data set. The proportion of each subset is given by their percentages as 60%, 20%, and 20%. To fit the data to the model, the model is trained using the training data set. The parameters of the classifier are set as well. After training multiple models using the training data set, the verification data set is used to identify each model, and determine the model accuracy rate and the error curves, select the parameters corresponding to the best performance model and select the corresponding model. Fig. 5 shows behavior of the loss function and the accuracy of a training set and a validation set. After the best-performing model is determined from the training set and the verification set, the test set is used for model prediction and performance evaluation of the model.

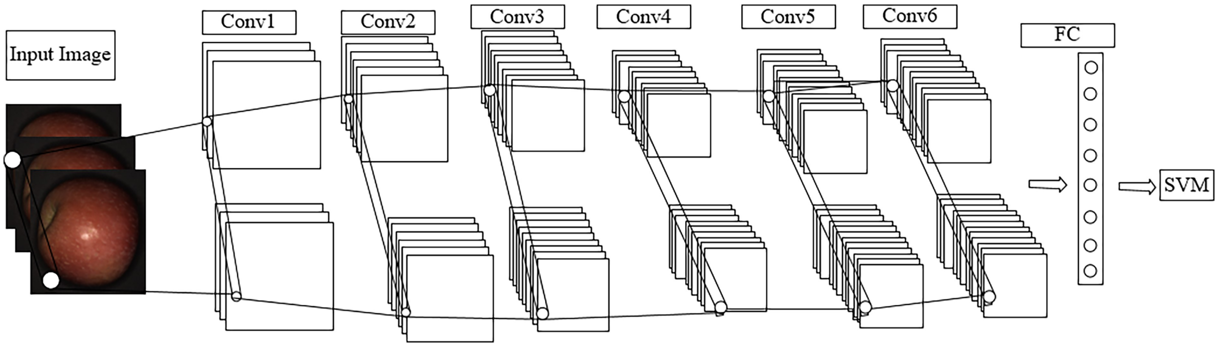


Fig. 4 The architecture of the CNN and SVM model for apple grading (CONV: Convolution layer; FC: Fully connected layer).

Table 2 Hyper parameters of the convolution layers.

Layers	Kernel size	Number of kernels	Stride
Conv 1	(6,6)	32	(1,1)
Conv 2	(6,6)	32	(1,1)
Conv 3	(3,3)	64	(1,1)
Conv 4	(3,3)	64	(1,1)
Conv 5	(5,5)	128	(2,2)
Conv 6	(5,5)	128	(2,2)

Table 3 Details of the apple dataset.

Apple Type	Grade Label	Training Samples	Validation Samples	Testing Samples
Grade 1	0	480	160	160
Grade 2	1	480	160	160
Grade 3	2	480	160	160
Grade 4	3	480	160	160

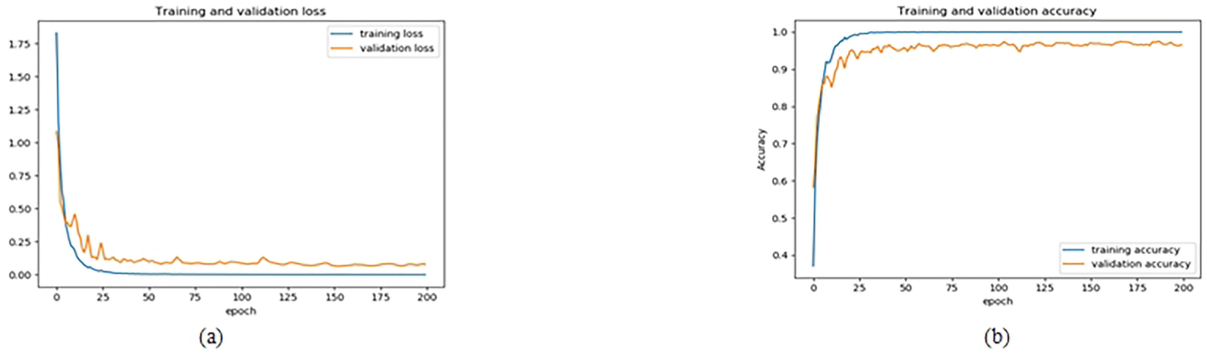


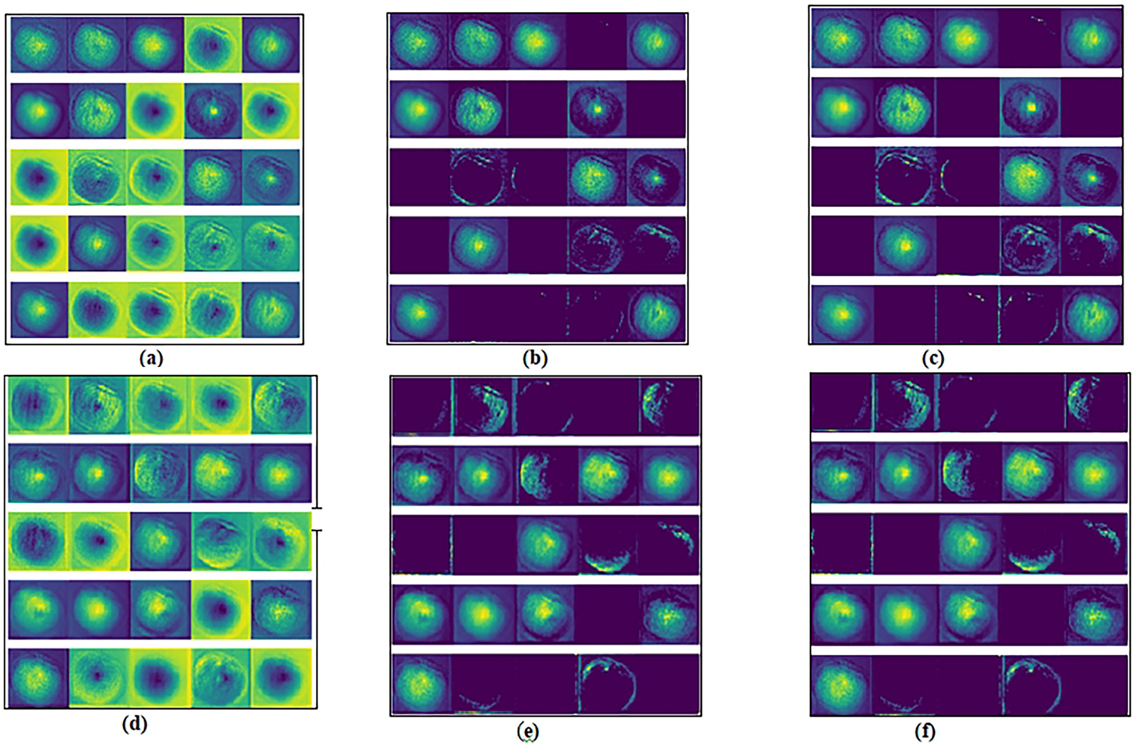
Fig. 5 (a) Loss of training and validation dataset; (b) Accuracy of training and validation dataset.

The CNN can extract image features through the convolutional layer and the pooling layer, and finally, determine the convolution kernel parameters through backpropagation, to obtain the final features. After learning features through a CNN model, the degree of activation of the background is very small. In the present paper, the features extracted by CNN are visualized, layer by layer, and key information of the apples is extracted. The feature map from the first layer is very close to the original image, indicating that it retains the features of the original image and does not learn any useful features. A solid color map begins to appear in the second and the third layers, indicating that the network begins to show sparsity and begins to extract local features, such as edge features. The feature map of the fourth layer extracts the color difference features and the apple center features. The local features in the feature maps of the fifth and the sixth layers are more obvious and the key feature identifiers of apples are extracted. In general, shallow networks contain more information and can extract detailed features. However, a deep network can effectively extract local key features. Relatively speaking, the deeper the layer, the more representative the extracted features, but the resolution of the image gets smaller and smaller with the additional layers. Fig. 6 shows the apple feature map extracted layer-by-layer of the CNN model constructed in the experiment.

The trained model is extracted through the fully connected layer. The dimension of the features extracted by CNN is still too high. The high-dimensional feature vectors are often led to dimensional disasters. In the present paper, principal component analysis (PCA) is used to reduce the multi-dimensional features into 2 dimensions, and the features extracted by the CNN have good clustering effects. The distribution of the raw apple dataset and the distribution of the apple dataset on PCA dimension reduction subsequent to CNN extraction are shown in Fig. 7. The present experiment uses an improved SVM grading algorithm. The model trained by the convolutional neural network is extracted from the fully connected layer, and a linear kernel function is selected by the SVM classifier using the OVO pattern grading. The choice of the hyperparameter C is done through experience. If the hyperparameter C is too large, it will easily cause over-fitting problems. If C is too small, it will easily cause under-fitting. Therefore, the value of C cannot be too large or too small. In the present experiment, C is set to 1 through grid search. Fig. 8 shows the partitioning of the apple dataset by the SVM when selecting linear kernel functions and Gaussian kernel functions. Since the features extracted by CNN are already linearly separable, it is sufficient to select a linear kernel function, whose boundaries are linear. The boundary of the Gaussian kernel function is curved.

3.3 Experimental Results and Discussion

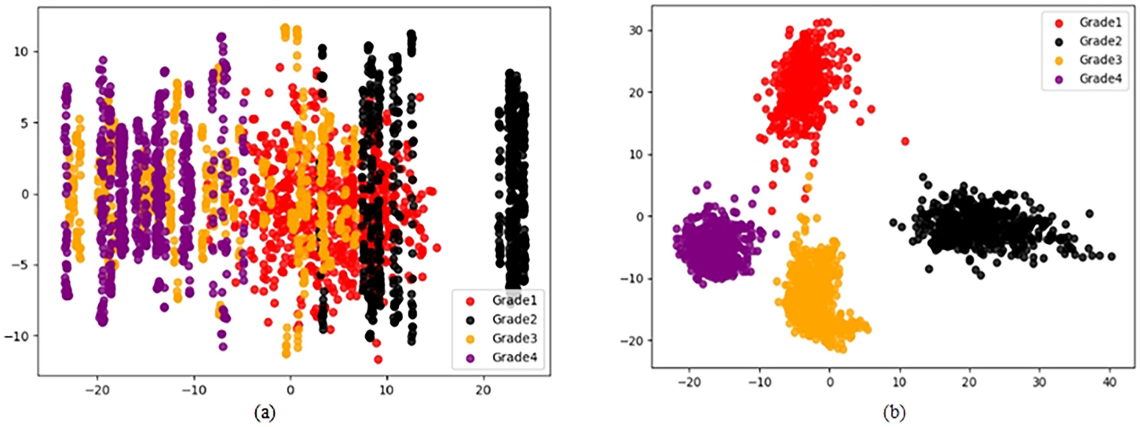
After the image acquisition and the feature extraction of apples, four different classifiers are used for prediction. The performance index of the model



(a) (b) (c)

(d) (e) (f)

Fig. 6 (a) The feature map of the first convolution layer; (b) The feature map of the second convolution layer; (c) The feature map of the third convolution layer; (d) The feature map of the fourth convolution layer; (e) The feature map of the fifth convolution layer; (f) The feature map of the sixth convolution layer.



(a)

(b)

Fig. 7 (a) Raw characters of the apples; (b) Features of apples after CNN training.

is evaluated according to accuracy, precision, recall, and F1_Score. The used four classifiers are: KNN, SVM, CNN, and CNN_SVM. The characteristics of KNN and SVM are selected based on the original features. The features from CNN and CNN_SVM are based on: the results from softmax after CNN training, for the former; and the results from the SVM classifier, for the latter. Table 4 presents the accuracy, precision, recall, and F1_Score from the four different grading algorithms. It is seen that, among them, the worst is the KNN algorithm whose accuracy level is only about 77%, and the best is the CNN_SVM algorithm whose accuracy level is 97%. In the separate CNN algorithm, the features from the fully connected layer are sent to the softmax layer. The accuracy level of this grading method is 95%. The accuracy level of the CNN_SVM algorithm is nearly 3% higher than that of CNN.

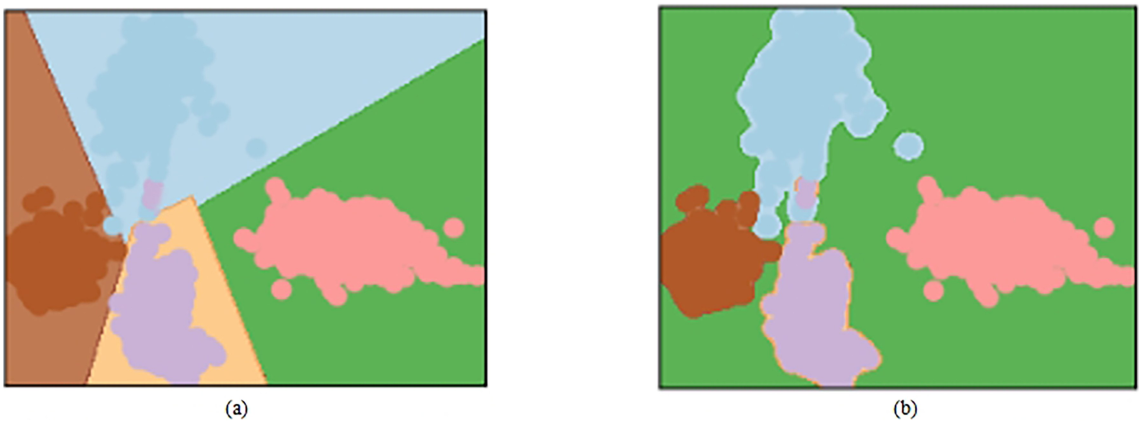


Fig. 8 (a) Linear kernel function features of graded apples; (b) Gaussian kernel function features of graded apple.

Table 4 Comparison of the performance of different approaches of apple grading.

Grading Method	Accuracy (100%)	Precision (100%)	Recall (100%)	F1_Score (100%)
KNN	77.46	79.44	76.45	76.90
SVM	92.14	92.30	91.76	91.90
CNN	95.00	95.27	95.26	95.10
CNN _SVM	97.97	98.10	98.10	98.00

To further evaluate the validity and feasibility of the CNN_SVM method, the accuracy of the four categories of apples grading is visualized in the form of a histogram.

Fig. 9 shows the accuracy of grading the four batches of apples, using different methods. It is seen that the CNN_SVM is superior to the CNN classifier on Grade 1, since the accuracy level of the CNN_SVM grading is about 97%, and the grading accuracy level of CNN is only about 84%.

4. Conclusion

This paper proposed a new method for automated grading of apples. The method is based on s feature fusion using a combined convolutional neural network (CNN) and support vector machine (SVM). A prototype experimental system was developed for the performance evaluation of the developed method. Experimental results showed that the developed the method was able to accurately grade

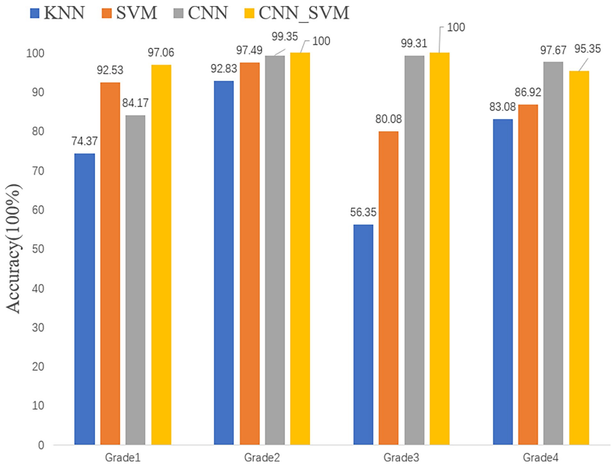


Fig. 9 The accuracy of grading four groups of apples using different approaches.

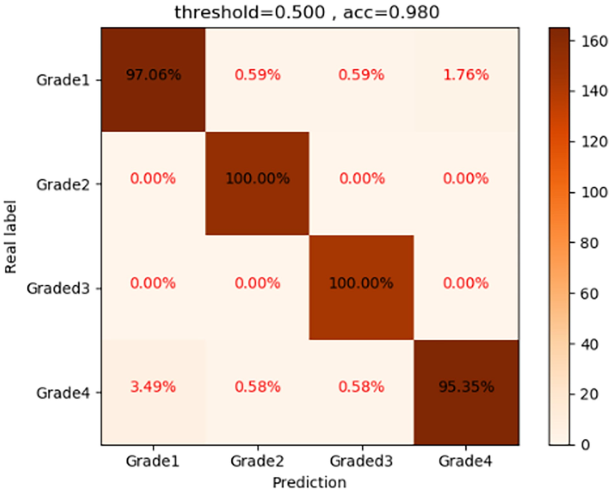


Fig. 10 Conditional grading confusion matrix for apple grading using the CNN_SVM model.

Yantai apples having similar color, size and shape, which are difficult to distinguish manually. After the original image is cropped, and the resulting data are enhanced and normalized, they were sent to the input layer of the CNN. By training the CNN model, the representative features were directly learned from the original image, without manually extracting the features. The characteristics extracted by the CNN were sent to the SVM classifier. It was found that the accuracy of the developed CNN_SVM grading model was 97.97%, the precision was about 98%, the recall rate was about 98%, and the F1_Score was also 98%. The performance of the developed method was compared with that of three other methods, and the developed method was found to be superior to that of the other methods.

References

[1] M.M.Sofu, O. Er, M.C. Kayacan. and B. Ceti li. (2016). Design of an automatic apple sorting system using machine vision[J]. *Computers and Electronics in Agriculture*, 127, pp.395-405.

[2] Chaitali Gaikwad. (2015). Review on Normal and Affected Fruit Classification[J]. *International Journal on Recent and Innovation Trends in Computing and Communition*. 3(5), pp.3425-3428.

[3] Kavdır, I. and Guyer, D.E. (2008). Evaluation of different pattern recognition techniques for apple sorting. *Biosystems Engineering* 99(2), pp. 211 – 219.

[4] Xibin Piao, Bing Wu and Hongwen Jia. (2013). Feature extraction and classification of apple near infrared spectrum[J]. *Computer engineering and applications*. 49(2), pp.170-193.

[5] Xin Wang, Ying Zhao and Jian Yang. (2014). Design of apple sorting system based on visual technology [J]. *Chinese journal of agricultural mechanization.* 35(05), pp.169-172.

[6] YangYu, SergioA. Velastin. and FeiYin. (2019). Automatic grading of apples based on multi-features and weighted K-means clustering algorithm[J]. *Information Processing in Agriculture*.

[7] Xuyang Pan, Laijun Sun, Yingsong Li, *et al*(2019). Non-destructive classification of apple bruising time based on visible and near - infrared hyperspectral imaging[J]. *Journal of the Science of Food and Agriculture* ,99(4).

[8] Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep learning (Vol. 1). *Cambridge: MIT press*, pp.326-366.

[9] Mandic, D. P.. (2004). A generalized normalized gradient descent algorithm[J]. *Signal Processing Letters IEEE*, 11(2), pp.115-118.

[10] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. *et al* (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J. *Mach. Learn. Res.*, *vol.* 15, pp. 1929 – 1958.

[11] Dominik Scherer, Andreas Müller. and Sven Behnke. (2010). Evalution of pooling operations in convolutional architectures for object recognition. *Lect. Notes Comput. Sci.* (*including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics*), vol. 6354 LNCS, no. PART 3, pp. 92 – 101.

[12] Min Xia, Teng Li, Lin Xu, *et al* (2018).Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks. *IEEE/ASME Transactions on Mechatronics*.

[13] Rumelhart D E, Hinton G. E. and Williams R J. (1986). Learning representations by back-propagating errors[J]. *Nature*, 323(3), pp.533-536.

[14] Cortes, C., and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20(3), pp.273 – 297.

[15] Daehyon Kim, Seungjae Lee, Seongkil Cho. (2007). Input Vector Normalization Methods in Support Vector Machines for Automatic Incident Detection. *Transportation Planning and Technology*.

[16] Bottou, L., and Lin, C. J. (2007). Support vector machine solvers. *Large scale kernel machines*, pp. 301-320.

[17] Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR*-98-14, *Microsoft research*.

[18] Joseph K. Nuamah and Younho Seong. (2019). A Machine Learning Approach to Predict Human Judgments in Compensatory and Noncompensatory Judgment Task. *IEEE Transactions on Human-Machine Systems*.

[19] Yunyun Liang and ShengLi Zhang. (2017). Predict protein Structural class by incorporating two different modes of evolutionary information into Chou's general pseudo amino acid composition. *Journal of Molecu-*

lar Graphics and Modeling.

[20] Guttormsen, Erik, Bendik Toldnes, Morten Bond, *et al* (2016). A Machine Vision System for Robust Sorting of Herring. *Fractions*, *Food and Bioprocess Technology*.

Authors' Biographies

Yuan CAI, is a Master degree candidate of Harbin Institute of Technology at Shenzhen (HITSZ). Her main research interest is mechatronic engineering.

Email: 1227756186@qq.com.

Clarence W. de Silva, received Ph.D. degrees from Massachusetts Institute of Technology, Cambridge, USA, in 1978, and the University of Cambridge, Cambridge, U. K., in 1998, the Honorary D. Eng. degree from the University of Waterloo, Waterloo, ON, Canada, in 2008, and the higher doctorate (Sc. D.) from the University of Cambridge in 2020. He has been a Professor of Mechanical Engineering and at the University of British Columbia, Vancouver, BC, Canada, since 1988. His appointments include the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation, Professorial Fellow, Peter Wall Scholar, Mobil Endowed Chair Professor, and NSERCBC Packers Chair in Industrial Automation. He has authored 25 books and about 560 papers, approximately half of which are in journals. His recent books published by Taylor & Francis/CRC are: *Modeling of Dynamic Systems—with Engineering Applications* (2018); *Sensor Systems* (2017); *Senors and Actuators—Engineering System Instrumentation*, 2^{nd} *edition* (2016); *Mechanics of Materials* (2014); Mechatronics—A Foundation Course (2010); Modeling and Control of Engineering Systems (2009); Sensors and Actuators— Control System Instrumentation (2007); VIBRATION—Fundamentals and Practice (2nd ed., 2007); Mechatronics—An Integrated Approach (2005); and by Addison Wesley: Soft Computing and Intelligent Systems Design—Theory, Tools, and Applications (with F. Karray, 2004). Prof. de Silva is a Fellow of: The Institute of Electrical and Electronics Engineers (IEEE), American Society of Mechanical Engineers (ASME), the Canadian Academy of Engineering, and the Royal Society of Canada.

Email: desilva@mech.ubc.ca

Bing Li, (SM'16) received the Ph.D. degree from Hong Kong Polytechnic University, Hong Kong, in 2001. He was a Professor of Mechatronics in 2006. He is currently the Head of the School of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen, China. His research interests include parallel manipulators and control, and mechanical vibration and control. He is serving as an Associate Editor for the International Journal of Mechanisms and Robotic Systems.

Email: libing.sgs@hit.edu.cn

Liqun WANG, received the B.S. degree from the Harbin Institute of Technology, China, in 2018, where he is currently pursuing the M.S. degree. His research interests include robot control and mechatronic engineering.

Email: 317310167@qq.com

Ziwen WANG, Now is a master in Harbin Institute of Technology at Shenzhen (HITSZ). His main research interest is mechatronic engineering and sensor.

Email: wangziwenhit@163.com