doi:10.3772/j.issn.1006-6748.2021.02.011

Multi-view recognition of fruit packing boxes based on features clustering angle①

Li Xinning (李鑫宁), Wu Hu, Yang Xianhai②
(School of Mechanical Engineering, Shandong University of Technology, Zibo 255000, P. R. China)

Abstract

In order to realize the intelligent mechanization of the last process of the fruit industry chains, the identification of fruit packing boxes is researched. A multi-view database is established to describe the omnidirectional attitudes of the fruit packing boxes. In order to reduce the data redundancy caused by multi-view acquisition, a new binary multi-view kernel principal component analysis network (BMKPCANet) is built, and a multi-view recognition method of fruit packing boxes is proposed based on the BMKPCANet and support vector machine (SVM). The experimental results show that the recognition accuracy of proposed BMKPCANet is 12.82% higher than PCANet and 3.51% higher than KPCANet on average. The time consumption of proposed BMKPCANet is 7.74% lower than PCANet and 29.01% lower than KPCANet on average. This work has laid a theoretical foundation for multi-view recognition of 3D objects and has a good practical application value.

Key words: boxes recognition, kernel principal component analysis (KPCA), binary hashing and clustering, multi-view clustering (MVC)

0 Introduction

As the largest fruit producer in the world[1], China is still at an initial stage in the development of fruit industry mechanization. More and more researchers have made amazing achievements in orchard fruit identification and fruit picking. The handling of the fruits packing boxes is time-consuming and labor-intensive, so the demand for intelligent handling is becoming increasingly urgent. The first thing to be solved is to identify specific types of fruit boxes based on machine vision.

Since 3D recognition adds a spatial dimension more than 2D, the most advanced 3D object detection algorithm is much slower than 2D method[2]. Therefore, it is a better choice that multiple 2D views are used to describe 3D objects. Su et al.[3] established the multi-view-based convolutional neural network (MVCNN). The features of multiple views were fused by the maximum pooling layers. Gao et al.[4] proposed end-to-end group-pair convolutional neural networks (GPCNN), which can solve the small scale problem. Considering the complementary information between views, they proposed a novel pairwise multi-view convolutional neural network (PMV-CNN) subsequently[5], in which the feature extraction and target recognition are unified into convolutional neural network (CNN). The team added the Slice Layer and Concat layer to the network for separating and integrating data, and established a multi-view discrimination and pairwise convolutional neural network (MDPCNN)[6], which could discriminate multi-batch input. Li et al.[7] developed a multi-view-based siamese convolutional neural network for 3D object retrieval. Yang et al.[8] constructed the multi-view semantic learning network (MVSLN) for laser radar point cloud features. The above researches have achieved good results and promoted the development of multi-view target recognition. However, the deep learning network models with complex structure and long training cycle are unfavorable to the practical application of handling robots for fruit packing boxes. Therefore, how to build a simple and efficient boxes identification model is the focus of this paper.

Chan et al.[9] put forward to simplify the principal component analysis of network (PCANet) aiming at the long training time of the classic CNN parameters. The network with simple model and high computing power had been widely used. Kernel principal component analysis (KPCA)[10] can realize the nonlinear dimension reduction and extract the nonlinear characteristics of data. Wu et al.[11] proposed the KPCANet model for

① Supported by the National Natural Science Foundation of China (No. 52075306).
② To whom correspondence should be addressed. E-mail: yxh@sdut.edu.cn
Received on Dec. 22, 2020

image classification, which achieved better classification results. Therefore, features extraction for fruits boxes are carried out based on KPCANet. Aiming at the multi-view recognition of fruits boxes, the model defined as BMKPCANet is proposed to reduce the data redundancy. The experiment results show that the recognition performance of the proposed method is superior to PCANet and KPCANet.

1 Multi-view clustering algorithms

The multi-view clustering (MVC) has always been one of the current research hotspots. Clustering algorithms can be divided into graph-based clustering, subspace-based learning clustering and binary learning clustering[12]. In Refs[13-15] the graph-based clustering algorithms, Nie et al. proposed the parameter-free auto-weighted multiple graph learning (AMGL)[13], the self-weighted multi-view clustering (SwMC)[14], and the multi-view clustering with adaptive neighbors (MLAN)[15] successively, which can reduce the influence of the selection of hyper-parameters on the clustering effect. In the subspace-based learning clustering algorithms, Zhang et al.[16] proposed the latent multi-view clustering (LMSC) method. Linear LMSC and generalized LMSC equations[17] were deduced to explore the potential complementary information of multiple views. Peng et al.[18] proposed a cross-view matching clustering (COMIC) algorithm, which eliminated the impact of artificial selection of parameters. Wang et al.[19] proposed an exclusivity-consistency regularity multi-view subspace clustering (ECMSC), which combined subspace learning and spectral clustering into a unified framework. Zhang et al.[20] proposed the one-stage partition-fusion multi-view subspace clustering (OP-MVSC) algorithm. The algorithm synthesized subspace learning, information fusion, and spectral clustering into a unified framework. It eliminated the influence of noise and redundant information in the original data. According to the advantages of the fusion framework, the feature extraction, coding and clustering were integrated in this work. For the problems of high computational cost and large memory consumption in most of the clustering methods, Shen et al.[21] proposed a compressed K-means (CKM). The computation was further reduced by compressing high-dimensional data into binary code. It was verified that the method was superior to most other clustering algorithms in terms of computation and memory consumption while ensuring clustering accuracy. Zhang et al.[22] applied binary code learning to multi-view clustering and proposed binary multi-view clustering (BMVC) method. The final optimization target was determined by constructing two partial losses of collaborative discrete representation learning and binary cluster structure learning. It was not difficult to find that the binary coding learning method could greatly improve the speed of multi-view clustering and save the storage space while ensuring the clustering performance. Based on this, this work combines the binary clustering learning method in the binary Hash coding stage of KPCANet model, and establishes a network model suitable for multi-view target recognition, which is defined as BMKPCANet model.

2 Recognition algorithm of fruits packing boxes

2.1 Recognition model construction of fruit packing boxes

In this work, the multi-view feature method is adopted to collect images. Multiple two-dimensional projections of different views are obtained to describe the characteristics of the fruits packing boxes under the principle of ensuring that the set of projected views is as small as possible and can represent multiple common attitudes of the boxes. The training dataset is input into the two-layer BMKPCANet network for feature extraction in this work, which is shown in Fig. 1. The model is composed of four stages. In the first stage, kernel principal component analysis is performed on the preprocessed input image. In the second stage, kernel principal component analysis is performed on the output matrix of the first stage. In the third stage, binary Hash coding and clustering are performed at the same time. In the last stage, block histogram transformation is performed on the output clustering feature and the output is obtained. The model will be introduced in detail in the following section.

2.2 BMKPCANet clustering features extraction

2.2.1 The first KPCA

Take the obtained image of size $m \times n$ as the input layer, and assume that the patch size is $k_1 \times k_2$. The sampling is done by sliding selection. All sampling blocks are collected and cascaded. If the jth block of the ith image is $x_{i,j}$, the ith image can be represented as

$$\boldsymbol{X}_i = [\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \cdots, \boldsymbol{x}_{i,\hat{m}\cdot\hat{n}}] \tag{1}$$

where, $\hat{m}$ and $\hat{n}$ are the number of patches on rows and columns, respectively. Then, the patch mean is subtracted from each patch, i.e.,

$$\bar{\boldsymbol{x}}_{i,j} = \boldsymbol{x}_{i,j} - \sum_{j=1}^{\hat{m}\cdot\hat{n}} \boldsymbol{x}_{i,j} / \hat{m} \cdot \hat{n} \tag{2}$$

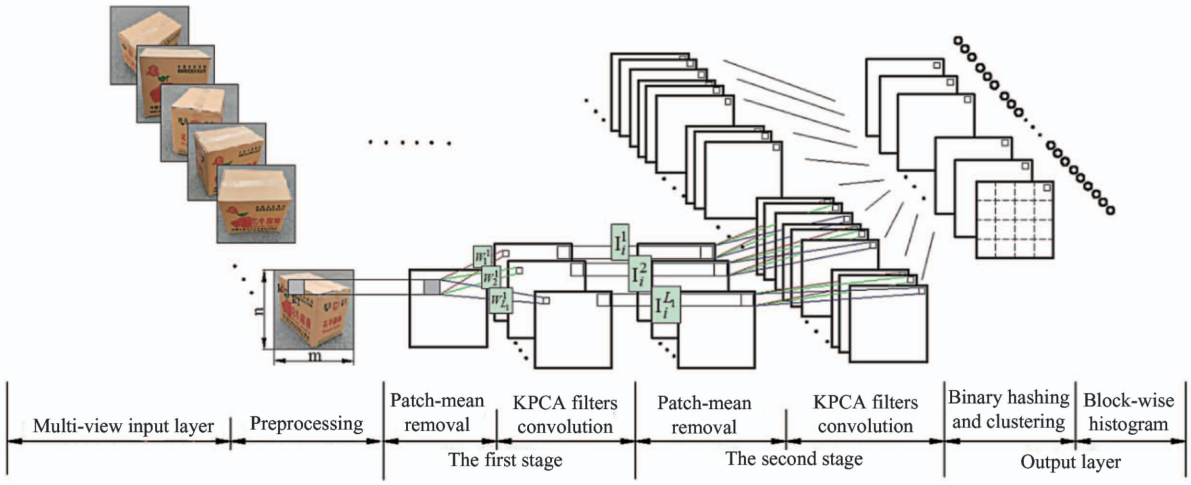


Fig. 1 Two-layer BMKPCANet clustering features extraction model

The local feature matrix of the ith image is expressed as

$$\overline{\boldsymbol{X}_i} = [\bar{\boldsymbol{x}}_{i,1}, \bar{\boldsymbol{x}}_{i,2}, \cdots, \bar{\boldsymbol{x}}_{i,\hat{m}\cdot\hat{n}}] \tag{3}$$

Do the same for other images in the training set and get a new matrix:

$$\boldsymbol{X} = [\overline{\boldsymbol{X}_1}, \overline{\boldsymbol{X}_2}, \cdots, \overline{\boldsymbol{X}_N}] \tag{4}$$

where N is the total number of training set images. Each column represents a patch, and there are $k_1 \times k_2$ elements. There are $N \times mn$ columns in all. Then the kernel transformation of the matrix is carried out to create the kernel matrix:

$$\boldsymbol{K}_1 = \text{ConstructKernelMatrix}(\boldsymbol{X}) = [\boldsymbol{K}_{11}, \boldsymbol{K}_{12}, \cdots, \boldsymbol{K}_{1N}] \tag{5}$$

The commonly used kernel functions include polynomial kernel function, Gaussian kernel function, PolyPlus kernel function, and polynomial kernel function, etc.[23]. The specific selection was introduced in the experimental part. The kernel matrix is centered to obtain K_{c1}, and then eigenvalue decomposition is performed for the K_{c1}. The eigenvectors corresponding to the previous L_1 large eigenvalues of the covariance matrix of K_{c1} are taken as the filter convolution kernel W_l^1. It can be presented as

$$W_l^1 = mat_{k_1,k_2}(q_l(\boldsymbol{K}_{c1}\boldsymbol{K}_{c1}^{\mathrm{T}})) \in R^{k_1,k_2}, \quad l = 1,2,\cdots,L_1 \tag{6}$$

For the input images $\boldsymbol{I}_i$, the output after principal component analysis of the first layer is

$$\boldsymbol{I}_i^l = \boldsymbol{I}_i \otimes W_l^1, \ i = 1,2,\cdots,N; \ l = 1,2,\cdots,L_1 \tag{7}$$

2.2.2 The second KPCA

The output of the first layer is the input of the second layer. For the each input image, L_1 features output matrixes have produced in the first layer, so there are L_1 inputs in the second layer. Then $L_1 \times N$ output matrixes can be obtained for the N images. The mapping process is the same as the first layer basically, which includes patches sampling, cascading and mean-removing. The matrix can be gotten after convolution of N images and the filter as

$$\boldsymbol{Y}^l = [\overline{\boldsymbol{Y}_1^l}, \overline{\boldsymbol{Y}_2^l}, \cdots, \overline{\boldsymbol{Y}_N^l}] \tag{8}$$

All filter outputs are cascaded together to obtain:

$$\boldsymbol{Y} = [\boldsymbol{Y}^1, \boldsymbol{Y}^2, \cdots, \boldsymbol{Y}^{L_1}] \tag{9}$$

Similarly, each column represents a patch for the matrix $\boldsymbol{Y}$, which contains $k_1 \times k_2$ elements, and there are $L_1 \times N \times mn$ columns. The kernel transformation of $\boldsymbol{Y}$ matrix is also performed, and kernel matrix $\boldsymbol{K}_2$ is created as

$$\boldsymbol{K}_2 = \text{ConstructKernelMatrix}(\boldsymbol{Y}) = [\boldsymbol{K}_{21}, \boldsymbol{K}_{22}, \cdots, \boldsymbol{K}_{2N}] \tag{10}$$

Then $\boldsymbol{K}_{c2}$ is obtained by centering the kernel matrix $\boldsymbol{K}_2$, and eigenvalue decomposition is carried out. The eigenvectors corresponding to the first L_2 large eigenvalues of the covariance matrix of the centralized kernel matrix $\boldsymbol{K}_{c2}$ are taken as the filter convolution kernel of the second layer, and then the convolution operation is performed on the image at the first layer. The output can be presented as

$$\boldsymbol{O}_i^l = \boldsymbol{I}_i^l \otimes W_\ell^2 = \boldsymbol{I}_i \otimes W_l^1 \otimes W_\ell^2,$$
$$\text{s.t. } i = 1,2,\cdots,N; \ l = 1,2,\cdots,L_1; \ \ell = 1,2,\cdots,L_2 \tag{11}$$

For each sample, the two layers PCA can produce $L_1 \times L_2$ features output matrixes. Based on the experience sets of relevant principal component analysis network models[9-10], the proposed network structure set two filters.

2.2.3 Binary Hash clustering

Binary multi-view clustering algorithm uses binary encoding technology to solve the clustering problem of multiple views. It can simultaneously optimize binary encoding and clustering of multiple views, which can

solve the problems of big data storage and long-time operation of previous multi-view clustering algorithms. It can greatly reduce the calculation time and storage space, and improve the calculation speed and efficiency. The specific operations are as follows.

(1) Hash encoding of multi-view data

Suppose the obtained multi-view dataset is divided into M views according to the region. Taking the mth view as an example, this work defines the output of the second layer principal component analysis as $\boldsymbol{O}_i^{l^m}$. P sample points $\{\boldsymbol{a}_j^m\}_{j=1}^p$ of the mth view are randomly selected, and the encoding of the data can be done based on nonlinear RBF mapping as follows:

$$\phi(\boldsymbol{O}_i^{l^m}) = [\exp(-\|\boldsymbol{O}_i^{l^m} - \boldsymbol{a}_1^m\|^2/\sigma), \cdots, \exp(-\|\boldsymbol{O}_i^{l^m} - \boldsymbol{a}_p^m\|^2/\sigma)]^{\mathrm{T}} \quad (12)$$

Where σ is the kernel width, $\phi(\boldsymbol{O}_i^{l^m}) \in \mathbb{R}^p$ is a p-dimensional nonlinear embedding for the ith sample from the mth view. The binary Hash function for $\boldsymbol{O}_i^{l^m}$ is defined as

$$\boldsymbol{h}_i^{l^m} = \mathrm{sgn}(\boldsymbol{U}^m \phi(\boldsymbol{O}_i^{l^m})) \quad (13)$$

where $\mathrm{sgn}(\cdot)$ is a step function that returns an integer variable and indicates the sign of the parameters, and $\boldsymbol{U}^m$ is the mapping matrix for the mth view. In order to make coding better reflect the correlation and complementarity between multiple views, the optimization function is designed as

$$\min_{\boldsymbol{U}^m, \boldsymbol{b}_i, \boldsymbol{\alpha}} \sum_{m=1}^M (\boldsymbol{\alpha}^m)^r \left(\sum_{i=1}^n \|\boldsymbol{b}_i - \boldsymbol{h}_i^{l^m}\|_F^2 + \beta \|\boldsymbol{U}^m\|_F^2 - \gamma \sum_{i=1}^n \mathrm{var}(\boldsymbol{h}_i^{l^m}) \right)$$
$$\text{s.t.} \quad \sum_m \boldsymbol{\alpha}^m = 1, \ \boldsymbol{\alpha}^m > 0, \ \boldsymbol{b}_i \in \{-1, 1\}^{q \times 1} \quad (14)$$

where $\boldsymbol{b}_i$ is the collaborative binary code for the ith instance, $\boldsymbol{\alpha}^m$ is the weight of the mth, different views have different weights, and $r > 1$ is a scalar quantity controlling the weights. The mapping matrix is expected to be as simple as possible, so minimize its L_2 paradigm as $\min \|\boldsymbol{U}^m\|_F^2$. In order to equalize the distribution of binary code, the variance is maximized that $\max \gamma \sum_{i=1}^n \mathrm{var}(\boldsymbol{h}_i^{l^m})$, i. e., $\min(-\gamma \sum_{i=1}^n \mathrm{var}(\boldsymbol{h}_i^{l^m}))$, γ is a nonnegative constant.

(2) Hash encoding co-clustering model

The clustering model adopts the method of matrix decomposition. Each encoding $\boldsymbol{b}$ is represented by the product of a clustering center $\boldsymbol{C}$ and an index vector $\boldsymbol{g}$. In order to minimize the decomposition error, the objective optimization function is defined as follows.

$$\min_{\boldsymbol{C}, \boldsymbol{g}_i} \|\boldsymbol{b}_i - \boldsymbol{C}\boldsymbol{g}_i\|_F^2$$
$$\text{s.t.} \quad \boldsymbol{C}^{\mathrm{T}}\boldsymbol{1} = 0, \ \boldsymbol{C} \in \{-1, 1\}^{q \times c}, \ \boldsymbol{g}_i \in \{0, 1\}^c, \ \sum_j \boldsymbol{g}_{ji} = 1 \quad (15)$$

Since coding and clustering are carried out simultaneously in this model, optimization functions Eqs(14) and (15) are combined together, and the total optimization function is

$$\min F(\boldsymbol{U}^m, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{G}, \boldsymbol{\alpha}) = \sum_{m=1}^M (\boldsymbol{\alpha}^m)^r (\|\boldsymbol{B} - \boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m})\|_F^2 + \beta \|\boldsymbol{U}^m\|_F^2 - \frac{\gamma}{n} tr((\boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m}))(\boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m}))^{\mathrm{T}}) + \lambda \|\boldsymbol{B} - \boldsymbol{C}\boldsymbol{G}\|_F^2 \quad (16)$$

$$\text{s.t.} \quad \boldsymbol{C}^{\mathrm{T}}\boldsymbol{1} = 0, \sum_m \boldsymbol{\alpha}^m = \boldsymbol{1}, \boldsymbol{\alpha}^m > 0, \boldsymbol{B} \in \{-1, 1\}^{q \times n}, \boldsymbol{C} \in \{-1, 1\}^{q \times c}, \boldsymbol{G} \in \{0, 1\}^{c \times n}, \sum_j \boldsymbol{g}_{ji} = 1 \quad (16)$$

where $\boldsymbol{B} = [b_1, \cdots, b_n]$, $\boldsymbol{G} = [g_1, \cdots, g_n]$, and λ is regularization parameter.

(3) Optimization

This work divides the optimization problem into several sub-problems. The optimization process adopts alternate optimization strategy. In other words, when a variable is updated, other variables are fixed, and it is a cyclic updating method in which each variable is updated alternately.

1) Updating $\boldsymbol{U}^m$. By fixing other quantities unchanged, the optimization problem Eq. (16) is transformed into

$$\min F(\boldsymbol{U}^m) = \|\boldsymbol{B} - \boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m})\|_F^2 + \beta \|\boldsymbol{U}^m\|_F^2 - \frac{\gamma}{n} tr((\boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m}))(\boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m}))^{\mathrm{T}}) \quad (17)$$

Define the derivative is zero by derivation, i. e., $\frac{\partial F(\boldsymbol{U}^m)}{\partial \boldsymbol{U}^m} = 0$, the optimal $\boldsymbol{U}^m$ can be obtained as follows:

$$\boldsymbol{U}^m = B\phi^{\mathrm{T}}(\boldsymbol{O}^{l^m})\boldsymbol{W} \quad (18)$$

where $\boldsymbol{W} = ((1 - \gamma/n)\phi(\boldsymbol{O}^{l^m})\phi^{\mathrm{T}}(\boldsymbol{O}^{l^m}) + \beta \boldsymbol{I})^{-1}$.

2) Updating $\boldsymbol{B}$. By fixing other quantities unchanged identically, the optimization problem Eq. (16) is transformed into

$$\min_{\boldsymbol{B}} \sum_{m=1}^M (\boldsymbol{\alpha}^m)^r (\|\boldsymbol{B} - \boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m})\|_F^2) + \lambda \|\boldsymbol{B} - \boldsymbol{C}\boldsymbol{G}\|_F^2 = tr[\boldsymbol{B}^{\mathrm{T}}(\sum_{m=1}^M (\boldsymbol{\alpha}^m)^r \boldsymbol{I} + \lambda \boldsymbol{I})\boldsymbol{B}] - 2tr[\boldsymbol{B}^{\mathrm{T}}(\sum_{m=1}^M (\boldsymbol{\alpha}^m)^r \boldsymbol{U}^m \phi(\boldsymbol{O}^{l^m}) + \lambda \boldsymbol{C}\boldsymbol{G})] + con \quad (19)$$

where con is the constant value with respect to $\boldsymbol{B}$. $(\sum_{m=1}^M (\boldsymbol{\alpha}^m)^r \boldsymbol{I} + \lambda \boldsymbol{I})$ is the multiplication of constant value and unit matrix, which is the constant value, and $tr(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}) = ql$ is also a constant value, so the optimization problem Eq. (19) can be rewritten as

$$\max_{\boldsymbol{B}} 2tr[\boldsymbol{B}^{\mathrm{T}}(\sum_{m=1}^{M}(\boldsymbol{\alpha}^m)^r\boldsymbol{U}^m\phi(\boldsymbol{O}^{l^m})+\lambda\boldsymbol{C}\boldsymbol{G})] \tag{20}$$

Because $\boldsymbol{B}$ are encodes, the optimization problem can be transformed into a sign problem, i. e.,

$$\boldsymbol{B}=\mathrm{sgn}(\sum_{m=1}^{M}(\boldsymbol{\alpha}^m)^r\boldsymbol{U}^m\phi(\boldsymbol{O}^{l^m})+\lambda\boldsymbol{C}\boldsymbol{G})) \tag{21}$$

3) Updating $\boldsymbol{C}$ and $\boldsymbol{G}$. By fixing other quantities unchanged but $\boldsymbol{C}$ and $\boldsymbol{G}$ identically, the optimization problem Eq. (16) is reduced to

$$\begin{aligned}&\min_{\boldsymbol{C},\boldsymbol{G}}\|\boldsymbol{B}-\boldsymbol{C}\boldsymbol{G}\|_F^2\\&\text{s.t. }\boldsymbol{C}^{\mathrm{T}}1=0,\ \boldsymbol{C}\in\{-1,1\}^{q\times c},\ \boldsymbol{G}\in\{0,1\}^{c\times n},\\&\sum_j\boldsymbol{g}_{ji}=1\end{aligned} \tag{22}$$

Same as the traditional K-means clustering algorithm, $\boldsymbol{C}$ and $\boldsymbol{G}$ are iteratively optimized. Fixing the other variables but $\boldsymbol{C}$, the work reformulates Eq. (22) as

$$\begin{aligned}\min F(\boldsymbol{C})&=\|\boldsymbol{B}-\boldsymbol{C}\boldsymbol{G}\|_F^2+\rho\|\boldsymbol{C}^{\mathrm{T}}\boldsymbol{1}\|^2\\&=-2tr(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{G})+\rho\|\boldsymbol{C}^{\mathrm{T}}\boldsymbol{1}\|^2+con\end{aligned} \tag{23}$$

where ρ is an arbitrarily large number. Due to the discrete constraint, the discrete proximal linearized minimization (DPLM) algorithm[24] is used for gradient descending, and the optimization problem of updating $\boldsymbol{C}$ is signed. $\boldsymbol{C}$ is updated in the $(P+1)$ iterations by

$$\boldsymbol{C}^{p+1}=\mathrm{sgn}(\boldsymbol{C}^p-1/\mu\ \nabla F(\boldsymbol{C}^p)) \tag{24}$$

where $\nabla F(\boldsymbol{C}^p)$ is the gradient of $F(\boldsymbol{C}^p)$.

Updating $\boldsymbol{G}$. Calculate the Hamming distance $H(\boldsymbol{b}_j,\boldsymbol{c}_i)$ between the jth binary encode $\boldsymbol{b}_j$ and the ith cluster centroid $\boldsymbol{c}_i$, find the nearest index vector $\boldsymbol{g}_i$ for the jth binary encode $\boldsymbol{b}_j$, then assign weight to 1 and the others to 0. The optimization solution of updating $\boldsymbol{G}$ can be obtained by

$$\boldsymbol{g}_{js}^{p+1}=\begin{cases}1 & s=\arg\min_i H(\boldsymbol{b}_j,c_i^{p+1})\\0 & \text{otherwise}\end{cases} \tag{25}$$

4) Updating the weight $\boldsymbol{\alpha}^m$. Fixing other quantities unchanged, the work defines as

$$\begin{aligned}z^m&=\|\boldsymbol{B}-\boldsymbol{U}^m\phi(\boldsymbol{O}^{l^m})\|_F^2+\beta\|\boldsymbol{U}^m\|_F^2\\&-\frac{\gamma}{n}tr((\boldsymbol{U}^m\phi(\boldsymbol{O}^{l^m}))(\boldsymbol{U}^m\phi(\boldsymbol{O}^{l^m}))^{\mathrm{T}})\end{aligned} \tag{26}$$

Then the optimization weight equation can be written as

$$\min_{\boldsymbol{\alpha}^m}\sum_{m=1}^{M}(\boldsymbol{\alpha}^m)^rz^m\quad\text{s.t.}\quad\sum_m\boldsymbol{\alpha}^m=1,\ \boldsymbol{\alpha}^m>0 \tag{27}$$

The problem is solved by Lagrange multiplier method. By introducing Lagrange multiplier, the equation of optimizing $\boldsymbol{\alpha}^m$ is transferred into

$$\min F(\boldsymbol{\alpha}^m,\boldsymbol{\eta})=\sum_{m=1}^{M}(\boldsymbol{\alpha}^m)^rz^m-\eta(\sum_{m=1}^{M}\boldsymbol{\alpha}^m-1) \tag{28}$$

Derivation with respect to α^m and η can be obtained as

$$\begin{cases}\dfrac{\partial F}{\partial\boldsymbol{\alpha}^m}=r(\boldsymbol{\alpha}^m)^{r-1}z^m-\eta\\[2ex]\dfrac{\partial F}{\partial\eta}=\sum_{m=1}^{M}\boldsymbol{\alpha}^m-1\end{cases} \tag{29}$$

Defining Eq. (29) as 0, the optimal solution of $\boldsymbol{\alpha}^m$ can be written as

$$\boldsymbol{\alpha}^m=(z^m)^{1/r}/\sum_m(z^m)^{1/r} \tag{30}$$

The optimization of binary Hash clustering has been completed. Binary Hash clustering flow chart is shown in Fig. 2.

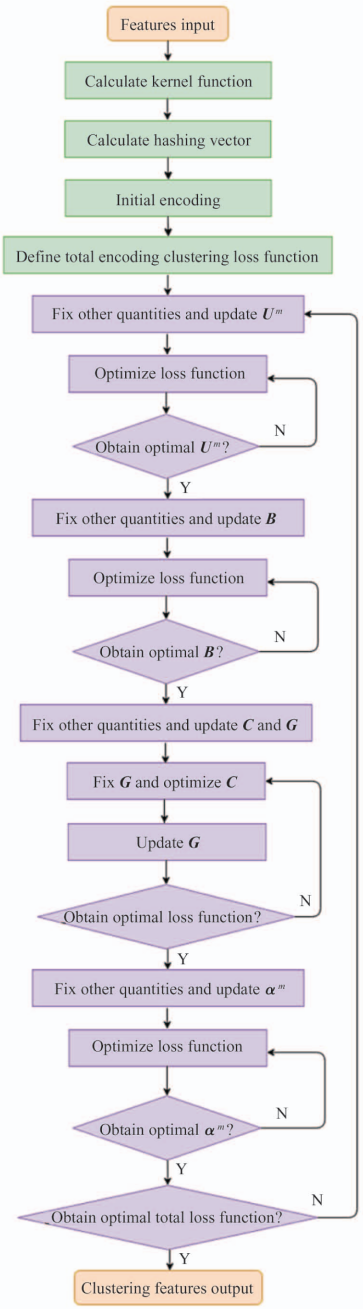


Fig. 2 Binary Hash clustering flow chart

2.2.4 Block histogram feature output

Since each input $\boldsymbol{I}_i^l$ in the second stage will have L_2 outputs, the L_2 outputs binary cell vector is taken as a whole to participate in clustering optimization. It is necessary that each optimized clustering feature is converted into decimal code with Eq. (31).

$$T_i^{l^m} = \sum_{l=1}^{L_2} 2^{l-1}(h_i^{l^m}) \quad \text{s.t. } l \in [1, L_1] \tag{31}$$

Each pixel of $T_i^{l^m}$ is an integer, and the range is $[0, 2^{L_2-1}]$. Each pixel $T_i^{l^m}$ is divided into Z blocks. Histogram statistics $Zhist(T_i^{l^m})$ are performed on each block, and the connection vector is quantified to obtain.

$$\boldsymbol{f}_i^m = [Zhist(T_i^{l^m}), \cdots, Zhist(T_i^{L_1^m})]^{\mathrm{T}} \in \mathbb{R}^{(2^{L_2})L_1 Z} \tag{32}$$

where, $\boldsymbol{f}_i^m$ is the final clustering feature vector in the mth view of the ith sample . The whole identification process of fruits packing boxes based on BMKPCANet clustering features extraction method is shown in Fig. 3.

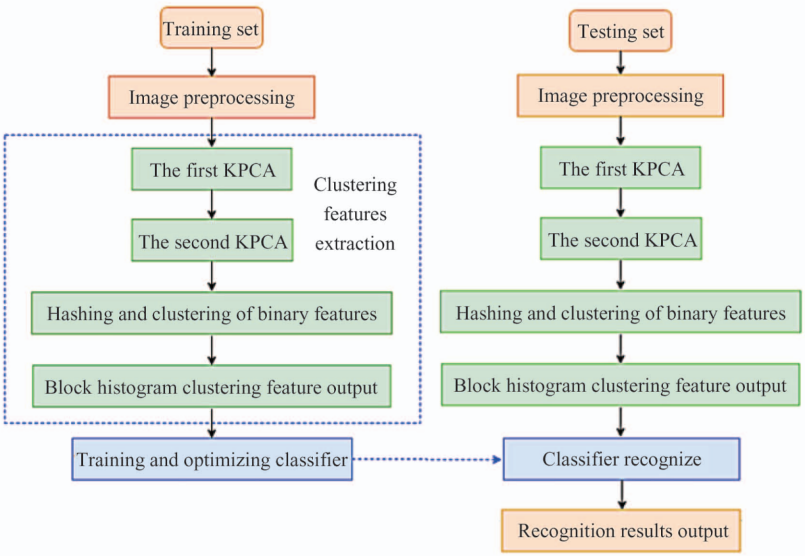


Fig. 3 Identification process of fruits packing boxes

3 Experiments and analysis

3.1 Experimental datasets

This work established the multi-view fruits packing boxes. In addition, in order to test the performance of the proposed model, the experiment also selected the public datasets ETH-80 and Coil-100. The specific details of the used datasets are as follows.

(1) ETH-80[25]. There are 8 classes in the dataset. Each class has 10 image sets, and each image set has 41 images. The sample objects in each image set are images of an object from different perspectives, which contain more and more comprehensive information about the object. All images are transformed into the gray images and resized to 32 × 32 pixel. This work randomly selects 4 sub-categories in each category for training, and the other sub-categories are taken as testing set.

(2) Coil-100[26]. Coil-100 contains 7200 images of 100 objects from different views. The size of image is 32 × 32 pixel. Each image is taken by rotating the object by 5 ° in the 360 °circle range, and each object has 72 images of different angles. This work randomly selects 20 types of objects as the research targets. After graying process, 36 images of each object are randomly collected as the training sets, and the remaining images are selected as the testing sets.

(3) Fruits packing boxes dataset. The images are collected at the fruit whole sale market in Zibo, China. Taking the gray cement floor as the background, video recording of the boxes at a uniform speed is carried out in the 180 ° hemisphere range above the ground to obtain the top and side images. It is necessary to guarantee that the boxes occupy more than 70% of the image. Then the images are captured in accordance with the time evenly within the videos, and the multi-view boxes images sets are obtained. The dataset consists of 15 kinds of the fruits packing boxes, which is defined as apple 1, apple 2, apple 3, watermelon 1, watermelon 2, orange 1, orange 2, hami melon 1, hami melon 2, pomegranate, pear, durian, coconut, banana and pineapple. The packing box of each category is shown in Fig. 4. Every category contains 200 images from different views. The images is adjusted to 32 × 32 pixel and

transformed into gray images. 50 images of each category are randomly selected for training, and the remaining images for testing.

Fig. 4 Categories of fruits packing boxes

3.2 Experimental parameters setting

Since the purpose of this algorithm is to identify fruits boxes, the fruits boxes dataset is taken as experimental object in the parameters selection. The experimental results are the average value obtained after 10 times. The recognition accuracy is taken as the evaluation index. The above experiment was based on Intel (R) Xeon(R) CPU E5-1650 v4@3.6 GHz, 64 GB RAM and NVIDIA GeForce GTX 10808G GPU platforms under the environment of Matlab 2017b and Python integrated environment Anaconda 3.

3.2.1 Selection of kernel functions

This work compares the performance of commonly kernel functions such as linear, polynomial, PolyPlus, Gaussian and Sigmoid kernel functions under the same BMKPCANet network parameters. Table 1 shows the expression of each kernel function and the setting of experimental parameters. Here $\boldsymbol{v}_i$, $\boldsymbol{v}_j \in \mathbb{R}^{mn}$ are row vectors of the matrix to be converted.

The patch size is set as 5×5, the block size is set as 8×8, and the overlap ratio of block is set as 0.5. According to the parameters in Table 1, the comparison of different kernel functions on the fruits boxes dataset is carried out. As shown in Fig. 5, it can be seen that accuracy with the Gaussian kernel function is the highest compared with other kernel functions.

Table 1 Common kernel functions and parameters selection

Functions	Expressions	Parameters
Linear	$K(\boldsymbol{v}_i, \boldsymbol{v}_j) = \boldsymbol{v}_i \boldsymbol{v}_j^{\mathrm{T}} + c$	$c = 0$
Polynomial	$K(\boldsymbol{v}_i, \boldsymbol{v}_j) = (\boldsymbol{v}_i \boldsymbol{v}_j^{\mathrm{T}})^d$	$d = 3$
PolyPlus	$K(\boldsymbol{v}_i, \boldsymbol{v}_j) = [(\boldsymbol{v}_i \boldsymbol{v}_j^{\mathrm{T}}) + 1]^d$	$d = 3$
Gaussian	$K(\boldsymbol{v}_i, \boldsymbol{v}_j) = \exp(-\Vert \boldsymbol{v}_i - \boldsymbol{v}_j \Vert^2 / 2\sigma^2)$	$\sigma = 1$
Sigmoid	$K(\boldsymbol{v}_i, \boldsymbol{v}_j) = \tanh(\alpha \boldsymbol{v}_i \boldsymbol{v}_j^{\mathrm{T}} + c)$	$\alpha = 1/2$ $c = -1$

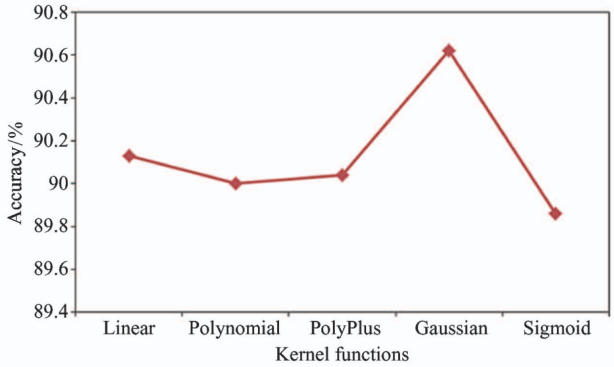


Fig. 5 Performance comparison of kernel functions on the fruits boxes dataset

3.2.2 Selection of model parameters

The following parameters are selected based on the same SVM classifier.

(1) Selection of filters number

The patch size is set as 5×5. The block size is 8×8. The overlapping ratio of block is 0.5. The number of filters changes from 2 to 14. The recognition accuracy of fruits boxes with different number of filters is shown in Fig. 6. The line with entitysquares refers to the recognition accuracy when the number of the first KPCA filter ranges from 2 to 14. The line with bordersquares refers to the recognition accuracy when the number of the second KPCA filter ranges from 2 to 14, while the number of the first KPCA filter is 8. It can be seen that when the number of KPCA filters is L_1, $L_2 \geqslant 8$, the method has high accuracy. However, larger number of filters will result in longer running time of feature extraction, the number of two layers of filters is set to 8 in this work, i.e., $L_1 = L_2 = 8$.

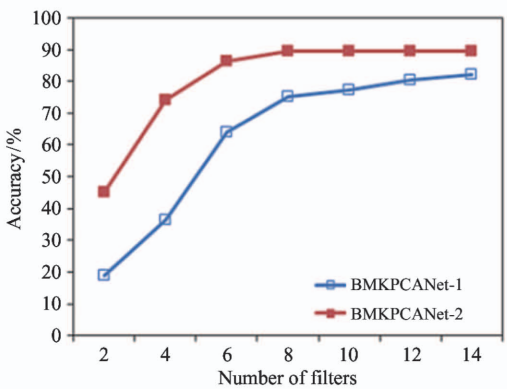


Fig. 6 Influence of filter on accuracy

(2) Selection of patch size

Under the premise of selecting $L_1 = L_2 = 8$, the PCA filter needs to meet the condition $k_1 k_2 \geqslant L_1, L_2$, so the minimum patch size is 3×3, and the maximum patch size is 13×13. The influence of different patch size on the recognition accuracy of the fruits boxes is

shown in Fig. 7. When the patch size is 5×5, the recognition accuracy is the highest relatively.

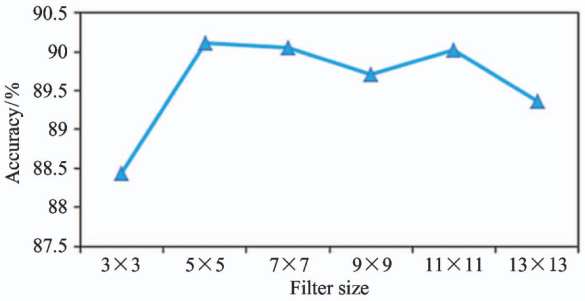


Fig. 7 Influence of patch sizes on accuracy

3.2.3 Selection of classifier

To choose a more suitable fruits boxes classifier, the work selects K-nearest neighbors with cosine similarity (KNN-cosine), K-nearest neighbors with Euclidean distance (KNN-Euclidean), linear support vector machine (linear SVM), and support vector machine based on the Gaussian radial basis function (RBF-SVM) for comparison. The patch size, block size, the overlapping ratio of block, and the number of filters are 5×5, 8×8, 0.5, and $L_1 = L_2 = 8$, respectively. The recognition accuracy of fruits boxes with different classifiers is shown in Fig. 8. As a classifier based on structural risk minimization criterion, SVM has achieved good classification results. The RBF-SVM is selected as the kernel function, that is,

$$k(x, y) = \exp(-\| x - y \|^2 / 2\sigma^2) \quad (32)$$

where σ^2 is the width of the kernel function.

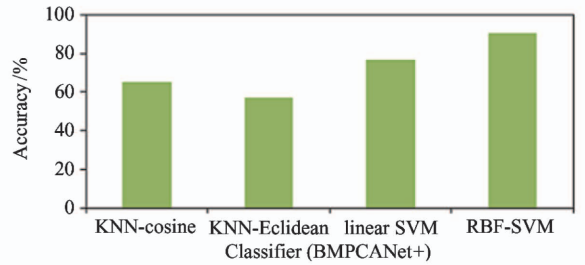
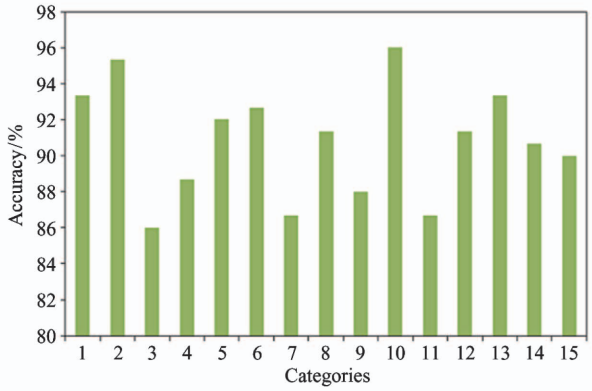


Fig. 8 Accuracy of fruits boxes with different classifiers

3.3 Experimental results

3.3.1 Recognition accuracy with different samples

Fifty images are randomly selected from each category as training samples, and the others are taken as testing samples. The RBF-SVM was used to classify them. This work selects the parameters of RBM kernel function by using grid search and cross validation methods based on the LIBSVM software package[27] and Python. The penalty coefficient C is 10 and the threshold value is 0.5. The recognition accuracy of each category is shown in Fig. 9. The overall accuracy of the 15 categories is 90.80%. The accuracies of categories 3, 4, 7, 9, and 11 are low because the sides and top surfaces of these types of boxes have no obvious characteristics, the recognition rate is low when observing the sides and top surfaces only.



1-apple 1, 2-apple 2, 3-apple 3, 4-watermelon 1, 5-watermelon 2, 6-orange 1, 7-orange 2, 8-hami melon 1, 9-hami melon 2, 10-pomegranate, 11-pear, 12-durian, 13-coconut, 14-banana, 15-pineapple

Fig. 9 Recognition accuracy of different categories

3.3.2 Comparison of recognition performance

The optimal parameters are selected for the experiment, and the recognition experiments are carried out on the ETH-80, Coil-100 and fruits packing boxes dataset. In order to reduce the data redundancy caused by multiple views, K-means, latent multi-view subspace clustering (LMSC) and cross-view matching clustering (COMIC) clustering algorithms are used to cluster features, respectively. The proposed BMKPCANet model with clustering characteristics is compared with the PCANet and KPCANet, which are combined with 3 clustering algorithms, respectively. The recognition accuracy and the consumed time of different models with the same RBF-SVM classifier are shown in the Table 2. The results are the average values after 10 runs.

It can be seen that the recognition accuracy of proposed BMKPCANet model is much higher than PCANet + K-means model, though it is slightly higher on time consumption. Although the recognition accuracy of ETH-80 dataset of BMKPCANet model is lower than that of KPCANet + LMSC model, the recognition-accuracy on COIL-100 and fruits boxes dataset is higher than that of other models. Since the BMKPCANet-model directly encodes and clusters the features vectors, its consumed time is lower than that of KPCANet + LMSC algorithm. On the ETH-80 dataset, compared with the average of PCANet combined with 3 clustering algorithms, the recognition accuracy is increased by 11.92%, and the time consumption is reduced by 5.17%. Compared with the average of KPCANet combined with 3 clustering algorithms, the recognition ac-

curacy is increased by 3.09%, and the time consumption is reduced by 29.50%. On the COIL-100 dataset, compared with the average of PCANetcombined with 3 clustering algorithms, the recognition accuracy is increased by 13.44%, and the time consumption is decreased by 9.04%. The recognition accuracy is increased by 3.54% and the time consumption is decreased by 27.04% than the average of KPCANet combined with 3 clustering algorithms. On the fruits boxes dataset, the recognition accuracy is increased by 13.11% and the time consumption is decreased by 9.00% than the average of PCANet combined with 3 clustering algorithms, and the recognition accuracy is increased by 3.90% and the time consumption is decreased by 30.50% than the average of KPCANet combined with 3 clustering algorithms. In summary, taking the average of the three datasets, the recognition accuracy is increased by 12.82% and the time consumption of BMKPCANet model is decreased by 7.74% than the average of PCANet related algorithms. The recognition accuracy is increased by 3.51% and the time consumption of BMKPCANet model is decreased by 29.01% than the average of KPCANet related algorithms. Considering the recognition accuracy and time consumption, the BMKPCANet model has the better performance compared with other models.

Table 2 Comparison of recognition performance

Method	ETH-80		COIL-100		Fruits packing boxes dataset	
	Accuracy/%	Time/s	Accuracy/%	Time/s	Accuracy/%	Time/s
PCANet + K-means	74.38	**35.44**	73.39	**153.21**	72.04	**160.47**
PCANet + LMSC	84.16	56.24	85.01	287.23	80.78	249.79
PCANet + COMIC	81.37	49.37	82.89	201.14	80.24	214.55
KPCANet + K-means	85.96	47.22	84.96	193.19	82.57	195.54
KPCANet + LMSC	**92.14**	72.38	93.18	322.45	90.09	328.34
KPCANet + COMIC	88.31	70.11	92.86	284.26	88.03	294.25
BMKPCANet	91.89	44.58	**93.87**	194.53	**90.80**	189.51

4 Conclusions

The work studies the recognition of different types of fruits packing boxes with the goal of realizing intelligent handling. 3D fruits boxes recognition is transformed into the multi-view 2D plane images recognition. The dataset of 15 kinds of different fruits boxes is established. The clustering features extraction method named BMKPCANet is proposed based on binary multi-view clustering and kernel principal component analysis to reduce the large data redundancy. The two-layer KPCA feature extraction network is constructed. The multi-view features are binary Hash encoded and clustered at the same time. The recognition accuracy and the time consumption of the proposed method are compared with the PCANet and KPCANet related models, which are combined with K-means, LMSC and COMIC clustering algorithms, respectively, on the fruits boxes dataset and the open datasets ETH-80 and COIL-100. The experimental results show that the BMKPCANet model has better recognition performance than other algorithms.

References

[1] Zheng Y J, Jiang S J, Chen B T, et al. Review on technology and equipment of mechanization in hilly orchard [J]. *Transactions of the Chinese Society for Agricultural Machinery*, 2020,51(11): 1-20 (In Chinese)

[2] Hu J Q,Sun L S, Shi M, et al. Multiscale oil-water layer recognition method based on RNN-FCNN[J]. *Chinese High Technology Letters*, 2020,30(3):305-313 (In Chinese)

[3] Su H, Maji S, Kalogerakis E, et al. Multi-view convolutional neural networks for 3d shape recognition[C]//Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015), Santiago, Chile, 2015: 945-953

[4] Gao Z, Wang D, He X, et al. Group-pair convolutional neural networks for multi-view based 3D object retrieval [C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018), New Orleans, USA, 2018:2223-2231

[5] Gao Z, Wang D Y, Xue Y B, et al. 3D object recognition based on pairwise multi-view convolutional neural networks [J]. *Journal Visual Communication Image Representation*, 2018, (56): 305-315

[6] Gao Z, Xue H X, Wan S H. Multiple discrimination and pairwise CNN for view-based 3D object retrieval[J]. *Neural Networks*, 2020,125:290-302

[7] Li H S, Zheng Y P, Cao J, et al. Multi-view-based siamese convolutional neural network for 3D object retrieval [J]. *Computers and Electrical Engineering*, 2019 (78): 11-21

[8] Yang Y G, Chen F, Wu F, et al. Multi-view semantic learning network for point cloud based 3D object detection [J]. *Neurocomputing*, 2020 (397): 477-485

[9] Chan T H, Jia K, Gao S, et al. PCANet: a simple deep

learning baseline for image classification[J]. *IEEE Transactions on Image Processing*, 2015,24(12):5017-5032
[10] Schölkopf B, Smola A, Müller K R. Nonlinear component analysis as a kernel eigenvalue problem[J]. *Neural Computing*, 1998,10(5): 1299-1319
[11] Wu D, Wu J S, Zeng R, et al. Kernel principal component analysis networkfor image classification[J]. *Journal of Southeast University (English Edition)*, 2015, 31(4):469-473
[12] Fu L L, Lin P F, Vasilakos V A, et al. An overview of recent multi-view clustering [J]. *Neurocomputing*, 2020 (402): 148-161
[13] Nie F P, Li J, Li X L. Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification[C]//Proceedings of the 25th International Joint Conferences on Artificial Intelligence Organization (IJCAI 2016), New York, USA, 2016:1881-1887
[14] Nie F P, Li J, Li X L. Self-weighted multiview clustering with multiple graphs[C]//Proceedings of the 26th International Joint Conferences on Artificial Intelligence Organization (IJCAI 2017), Melbourne, Australia, 2017:2564-2570
[15] Nie F P, Cai G H, Li X L. Multi-view clustering and semi-supervised classification with adaptive neighbours[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017), San Francisco, USA, 2017: 2408-2414
[16] Zhang C, Hu Q, Fu H, et al. Latent multi-view subspace clustering[C]//Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, USA, 2017:4333-4341
[17] Zhang C Q, Fu H Z, Hu Q H, et al. Generalized latent multi-view subspace clusterin[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020,42(1): 86-99
[18] Peng X, Huang Z, Lv J, et al. COMIC: multi-view clustering without parameter selection[C]//Proceedings of the 36th International Conference on Machine Learning (ICML 2019), Long Beach, USA, 2019:8925-8934
[19] Wang X, Guo X, Lei Z, et al. Exclusivity-consistency regularized multi-view subspace clustering[C]//Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, USA, 2017:923-931
[20] Zhang P, Zhu E, Cai Z P. One-stage partition-fusion multi-view subspace clustering algorithm [J]. *Journal of Frontiers of Computer Science and Technology*, doi: 10.3778/j.issn.1673-9418.2009070, 2020:1-9 (In Chinese)
[21] Shen X, Liu W, Tsang I, et al. Compressed k-means for large-scale clustering[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017:2527-2533
[22] Zhang Z, Liu L, Shen F, et al. Binary multi-view clustering[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018 (41): 1774-1782
[23] Hu W P, Hu H F, Gu J Q, et al. Kernel principal component analysis network method for face recognition[J]. *Acta Scientiarum Naturalium Universitis Sunyatseni*, 2016, 55 (5):48-51, 56
[24] Shen F, Zhou X. A fast optimization method for general binarycode learning[J]. *IEEE Transactions on Image Processing*, 2016, 25(12):5610-5621
[25] Leibe B, Schiele B. Analyzing appearance and contour based methods for object categorization[C]//Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, USA, 2003: 409-415
[26] Nene S A, Nayar S K, Murase H. Columbia Object Image Library (COIL 20), CUCS-005-96[R]. New York: Department of Computer Science, Columbia University, 1996
[27] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. *ACM Transactions on Intelligent Systems and Technology*, 2011,2(3): 1-39

Li Xinning, born in 1986. She is currently pursuing the Ph. D degree at Shandong University of Technology. She received the B. E. degree in mechanical design, manufacturing and automation from Yantai University, China, in 2007, and the M. E. degree in mechanical design and theory from Lanzhou University of Technology, China, in 2010. Her research interests include advanced manufacturing technology, intelligent machine learning and recognition.