IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 20, 2019, accepted September 7, 2019, date of publication September 13, 2019, date of current version September 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2941321

A SVM Multi-Class Image Classification Method Based on DE and KNN in Smart City Management

WANNENG SHU[1], (Member, IEEE), AND KEN CAI[2]
[1]College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
[2]College of Automation, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China

Corresponding author: Ken Cai (icken@zhku.edu.cn)

This work was supported by the National Nature Science Foundation of China under Grant 61603420.

ABSTRACT When directly operated in an image, good results are always difficult to achieve via conventional methods because they have poor high-dimensional performance. Support vector machine (SVM) is a type of machine learning method with solid foundation that is developed based on traditional statistics. It is also a theory for statistical estimation and predictive learning of objects. This paper optimizes the structure of SVM classification tree with differential evolution (DE) and designs the corresponding DE algorithm to effectively solve the problem of image classification of complex background cases in smart city management systems. In the training process of SVM classification tree, it obtains an optimal two-class classification scheme in every node by means of DE, initially separates the classes that are easy to be separated and then the less easy ones, and finally adaptively generates the best classification tree. The simulation experiment proves that the proposed algorithm is effective when applied to smart city management systems.

INDEX TERMS Smart city management, support vector machine, image classification, differential evolution, K-nearest neighbor.

I. INTRODUCTION

Smart city is a new concept and mode for promoting the intellectualization of urban planning, construction, management, and service by using the new generation of information technology, such as Internet of things, cloud computing, big data, artificial intelligence, block chain, and spatial geographic information integration. The construction of a smart city mainly includes smart city management, grid social management, safe city, smart transportation, and other industrial application projects. The extraction and expression of image features are the fundamental and core technology of content-based image retrieval technique. In a broad sense, image features include two types, namely, text-based features (e.g., keyword and annotation) and visual features (e.g., color, texture, and shape) [1]. Content-based image retrieval automatically extracts and analyzes visual features, such as color, texture, and shape of an image, and then creates an index for quick inquiry. It directly analyzes image contents (i.e., color, texture, and shape of the image and the spatial relationship of objects) and then extracts these contents as feature vectors from which the index is based [2]. Support vector machine (SVM) originated from the optimization theory of statistical learning. It studies the manner in which a learning machine is formed and achieves pattern classification. The most prominent characteristic of SVM is that it constructs the decision hyperplane on the basis of the principle of Vapnik structural risk minimization (SRM) to maximize the margin between every class of data [3]. SVM is a classifier with a relatively strong classification capacity and excellent generalization performance and it is extensively applied. However, SVM still has many issues, such as selecting among different kernel functions and conventional training algorithms being slow in speed and having complex algorithms and computation [4].

The core idea of K-nearest neighbor (KNN) algorithm is that if the majority of k nearest neighbors of a sample belongs to a certain class in the feature space, then this sample also belongs to this class and has the attributes of the sample in this class [5]. KNN determines which class the sample belongs to by relying on limited neighbors instead of discriminating the class domains; thus, it is suitable for the set of samples with many crossed or overlapped class domains. Meanwhile, differential evolution (DE) is an evolutionary algorithm based on real-number encoding for optimizing the minimum value of the function. DE was proposed when solving Chebyshev polynomial and it is an evolutionary computation method

The associate editor coordinating the review of this manuscript and approving it for publication was Lu Liu.

based on population differences. Its overall structure resembles that of genetic algorithm and it has the operations of mutation, crossover, and selection [6].

The special contributions of this paper are as follows.

- This paper introduces image features and classification, semantic analysis and extraction of an image, and the theory of SVM. It analyzes the effects of SVM kernel parameters on the classification model.
- This paper investigates several methods for selecting kernel parameters and optimizes the model parameters with DE.
- Taking the image features extracted as the input of classifier, this paper achieves auto image classification by applying DE-SVM into image classification. It also analyzes the effects of optimization methods with different features and kernel parameters on classification performance.
- To solve the poor between-class separability in the lower-level nodes of the best (or better) classification tree, this paper combines KNN with SVM to discriminate which class the samples belong to during the classification.

The rest of this paper is organized as follows. Section II discusses related works, followed by the construction of the proposed SVM classifier in Section III. The proposed algorithm and classification process are discussed in Section IV. Section V shows the simulation experimental results. Finally, Section VI concludes the study with a summary and future research directions.

II. RELATED WORKS

Content-based image classification technique is interdisciplinary and evolves closely related to fields, such as computer vision and machine learning; the development of such method promotes the progress of content-based image classification technique. People always describe the visual features of an image from three perspectives, namely, color, texture and shape, and these features can be used in numerous fields, including content-based image retrieval, image classification, image recognition, and image tracking among others [7]. The content features of an image depict the semantic features contained in the image and they can be taken as abstract image representation. Image features can be divided into low-level features (e.g., color, texture, and shape) and high-level features (i.e., semantic feature). Content-based image classification is always used together with the features close to human visual perception; it classifies using the color, texture, and shape of the image and utilizes the spatial location features as image features. The study of image classifiers has made rapid progress with the emergence of machine learning methods [8]. In recent years, an increasing number of researchers have begun to study the applications of machine learning in image classification. The foundation and core of content-based image retrieval are to extract visual image features and how to retrieve using these features. The image features extracted include two types, namely, text features (e.g., keyword and annotation) and visual features (e.g., color, texture, shape, and appearance) [9].

The idea of SVM is to select a better Vapnik–Chervonenkis (VC) dimension to compromise the empirical risk and the confidence value; in this manner, the margin between every class of data can be maximized and the actual risk can be lowered on the premise of a proper number of samples [10]. In 1992, Vapnik et al. were the first to propose SVM, which was based on the concept of VC dimension. At present, the study of SVM has some limitations. Its performance largely depends on the selection of kernel functions; however, a better method for the selection of kernel functions for specific problems is unavailable. The existing SVM theory only discusses the situations with a fixed penalty coefficient; however, in reality, the losses caused by two misjudgments on positive and negative samples are different [11]. The current SVM study mainly focuses on two aspects as follows: one is the optimization of kernel, the applications of kernel trick, and the improvement of SVM, which are helpful supplements to standard SVM; and the other is the theoretical deepening and improvement of standard SVM and the combination with other optimization methods [12].

Cover and Hart proposed the first KNN in 1968, which is a classification algorithm. KNN belongs to instance-based learning and is a type of lazy learning; thus, KNN has no explicit learning process or training phase [13]. KNN data sets already have prior classification and feature values and they can be directly processed after receiving new samples. Under this circumstance, KNN corresponds with eager learning. The three basic elements of KNN are the distance measurement, the selection of k, and the classification decision rule. On the basis of the distance measurement selected (i.e., Manhattan or Euclidean distance), KNN can calculate the distance from the test samples and every sample in the training set [14].

In the process of feature extraction, the feature description in the high-dimensional space can be described by the feature description in the low-dimensional space through mapping or transformation [15]. Meanwhile the process of feature selection selects some of the most effective features from a set of features to reduce the dimension of feature space. Texture features are not based on pixel characteristics [16]. Such features need to be computed statistically in regions containing multiple pixels. In pattern matching, this regional feature has great advantages and it cannot be matched successfully because of local deviation.

At present, the shape-based retrieval method still lacks a relatively perfect mathematical model. If the target is deformed, then the retrieval results are often unreliable. Many shape features only describe the local nature of the target. Describing a target comprehensively often requires high computation time and storage capacity [17]. The shape information reflected by shape feature is not completely consistent with human's intuitive sense, or the similarity of feature space is different from that of the human visual system.

Semantic matching extracts the corresponding semantic features and then performs semantic retrieval of user advertisements [18]. The semantic features of users and advertisements are extracted separately to calculate relevance. For structured and semi-structured information, batch processing can be conducted by establishing mapping relationships between the original structure and semantic classes and attributes. For unstructured information, the common idea is to convert video and audio into the corresponding text and then extract it. Semantic extraction is the core issue of image semantic retrieval [19]. The ultimate goal of semantic extraction is to express each image in the image database with certain semantics. The two processes in semantic extraction are construction of image semantic classifier and semantic annotation. The former refers to the process of mapping or linking image or image visual features to image semantic representation [20].

Color features reflect the overall characteristics of a color image. An image can be approximated by its color characteristics [21]. Generally, color features are based on pixel characteristics. Texture features refer to the change of gray level or color of image pixels. They can obtain quantitative description and interpretation of image texture features and structure for image analysis, segmentation, and understanding. Texture features are values calculated from the image that quantify the features of gray level changes within the region [22]. The basic process of texture analysis is to extract some distinguishing features from the texture image from the pixels, which are used as texture primitives for detection; determine the information of the arrangement of texture primitives; establish the texture primitive model; and then use the texture primitive model to further segment, classify, or identify the texture image. The spatial-domain analysis within the region is directly in the spatial domain of the image without transformation, and the shape features are extracted from the region. Image classification and extraction are key tasks in many fields, including image retrieval, object detection in visual scene, network information filtering, and medical image application [23]–[28]. The two processes are accomplished in the SVM semantic classifier training module and the unknown image semantic indexing [11].

III. CONSTRUCTION OF SVM CLASSIFIER

First, SVM searches the optimal classification hyperplane for two-class samples in a linearly separable space. In case of linear inseparability, it adds the slack variable for analysis and maps the samples in the low-dimensional input space into the high-dimensional attribute space to make it linearly separable by means of nonlinear mapping; this approach makes it possible to analyze the nonlinearity of samples through linear algorithms and searches the optimal classification hyperplane in this feature space. Then, it constructs the optimal classification hyperplane in the attribute space with SRM to globally optimize the classifier and make the expected risk of the entire sample space satisfy a certain upper bound at a certain probability.

A. LINEAR SVM

Linear separability separates two classes with one straight line. Generally, it has two circumstances, that is, exactly correct separation and approximately correct separation. Many straight lines can separate two classes, and SVM aims to determine the optimal classification hyperplane and maximize the margin between them. Fig. 1 denotes the linearly separable SVM.

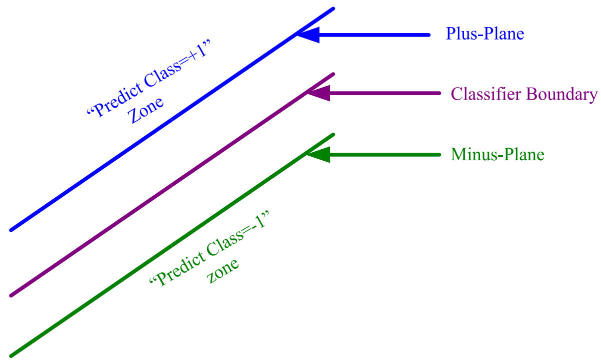


FIGURE 1. Linearly separable SVM.

The classifier boundary in Fig. 1 is $f(x)$. The red and blue lines (i.e., + and − planes, respectively) are the planes in which the support vectors are located, and the margin between the red and blue lines is the between-class margin to be maximized.

In object space, the solid-line formula is $w * x + b = 0$, marked as (w, b). It becomes the hyperplane when applied to the high-dimensional space. The distance formula from point x to the hyperplane is

$$d = \frac{|w * x + b|}{\|w\|} \tag{1}$$

Then,
$w \cdot x + b \geq +1, y = +1$ (positive); and
$w \cdot x + b \leq -1, y = -1$ (negative).

These formulas can be merged into
$y_i(w \cdot x_i + b) \geq 1, i = 1, 2, 3...$

The relationship between the distances from two-class data to the full-line separation is

$$\frac{|w \cdot x + b|}{\|w\|} \geq \frac{1}{\|w\|} \tag{2}$$

Then, the distance expression to separate two-class data is $2/\,||w||$.

SVM maximizes the distance of separation. The optimization equation is then obtained as follows:

$$\max_{w,b} \frac{2}{\|w\|}$$
$$s.t.y_i(w \cdot x_i + b) \geq 1.i = 1, 2, 3... \tag{3}$$

The negative label y is set as −1; hence, the boundary conditions can be merged into one formula.

It is translated into the following form, which is the final optimization equation:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$
$$s.t. y_i(w \cdot x_i + b) \geq 1. i = 1, 2, 3 \ldots \tag{4}$$

B. NONLINEAR SVM

Most classification problems are not linearly separable but nonlinearly separable, in which case separating two classes with a straight line is infeasible because the classification error will be relatively large. To solve the problem of nonlinear separability, nonlinear SVM has emerged [15].

Nonlinear SVM initially maps the input space into the high-dimensional feature space through the nonlinearity of kernel functions and then searches the optimal linear classification plane in the high-dimensional classification plane; this plane corresponds to the nonlinear classification plane in the input space. For linear inseparability, SVM introduces the slack variable and penalty factor to turn the object function into

$$\phi(w, \xi) = \frac{1}{2}(w \cdot w) + C \left[\sum_{i=1}^{N} \xi_i \right] \tag{5}$$

SVM transforms the input space into the high-dimensional space via nonlinear transformation SVM and then searches the optimal classification plane in the new space. The dot production operation under linear separability becomes $k(x, y) = \phi(x) \cdot \phi(y)$; in this manner, the final classification function obtained is

$$f(x) = sign \left[\sum_i \alpha_i y_i k(x_i \cdot x) + b \right] \tag{6}$$

The three types of frequently-used kernel functions are polynomial kernel function, radial basis function (RBF), and sigmoid kernel function.

1) POLYNOMIAL KERNEL FUNCTION

$$K(x, x_i) = [(x, x_i) + 1]^q \tag{7}$$

where q is the order of polynomial, and what is obtained is the q-order polynomial classifier.

2) GAUSSIAN RBF

$$K(x, x_i) = \exp\{-\frac{|x - x_i|^2}{\sigma^2}\} \tag{8}$$

The SVM obtained is a radial basis classifier; its fundamental difference from conventional RBF methods is that the center of every basis function here corresponds to a support vector. They and their output weight are automatically determined by the algorithm. The inner product function in the form of radial basis is similar to the human visual characteristics and is frequently used in practical applications. If different values of S are selected, then the corresponding classification plane will be considerably different [17].

3) SIGMOID KERNEL FUNCTION

$$K(x, x_i) = \tanh[v(x \cdot x_i) + c] \tag{9}$$

where $v > 0, c < 0$. SVM includes a multilayer perceptron network in the hidden layer. The weight and the number of nodes in the hidden layer are automatically determined by the algorithm instead of experienced as the conventional perceptron network. Moreover, this algorithm does not have the problem of local minimum point, which is encountered by the neural network. The effect of Gaussian kernel functions on data separation is shown in Fig. 2.

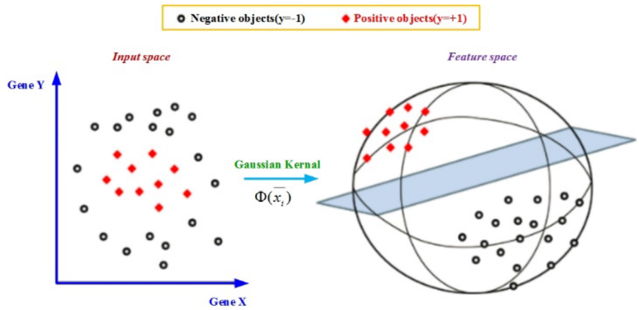


FIGURE 2. **Effect of gaussian kernel functions on data separation.**

When solving a linear inseparability problem, kernel functions use the computation in the low-dimensional feature space to avoid the large amount of computation in the inner product of vectors in the high-dimensional feature space. At this time, the SVM model can use the advantage of linear separability of data in the high-dimensional feature space and avoid introducing this high-dimensional feature.

Let $\phi(\cdot)$ be a mapping from the low-to the high-dimensional feature space. If function $K(X, z)$ exists, then the following equation exists for any low-dimensional feature vectors X and z:

$$K(x, z) = \phi(x) * \phi(z) \tag{10}$$

where $K(x, z)$ is the kernel function [18].

C. TRAINING OF SVM CLASSIFIER

For the common nonlinear separability problem, SVM transforms the nonlinearly separable input features in the low-dimensional space into the high-dimensional space through kernel functions and finds the optimal classification hyperplane in the high-dimensional space to achieve classification. With regard to the selection of kernel function, RBF has a broader convergence domain compared with other kernel functions and it is applicable for classification problems with a small number of samples. In terms of multiclassification realization, "one-to-one" classification combination has a fast training speed and it avoids the problem of inseparability. This paper selects RBF as the kernel function and constructs an image classifier with "one-to-one" combination [19].

For the problem of linear inseparability, separating negative samples from positive ones with a hyperplane is impossible. However, changing the hyperplane into a hypersurface can correctly classify positive and negative samples, as shown in Fig. 3.

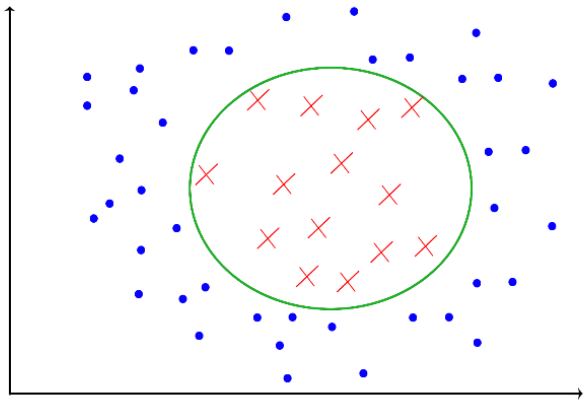


FIGURE 3. Positive and negative objects separated by hypersurface.

The surface equation is

$$k_1x_1^2 + k_2x_2^2 + k_3x_1 + k_4x_2 + k_5x_1x_2 + k_6 = 0 \qquad (11)$$

The new coordinate is mapped as follows:

$$z_1 = x_1^2, \quad z_2 = x_2^2, \; z_3 = \sqrt{2}x_1,$$
$$z_4 = \sqrt{2}x_2, \quad z_5 = \sqrt{2}x_1x_2 \qquad (12)$$

The hypersurface is represented into the hyperplane in the new coordinate as follows:

$$k_1'z_1 + k_2'z_2 + k_3'z_3 + k_4'z_4 + k_5'z_5 + k_6' = 0 \qquad (13)$$

That is, the hyperspace transforms the linear inseparability problem in two-dimensional space (x_1, x_2) into a linear separability problem in five-dimensional space $(z_1, z_2, z_3, z_4, z_5)$.

The mapped inner product is obtained in the new coordinate as follows:

$$(\varphi(p)\cdot\varphi(q)) = p_1^2q_1^2 + p_2^2q_2^2 + 2p_1q_1 + 2p_2q_2 + 2p_1p_2q_1q_2 \qquad (14)$$

The following kernel functions are used:

$$k(p, q) = ((p\cdot q) + 1)^2 (p_1q_1 + p_2q_2 + 1)^2;$$
$$k(p, q) = p_1^2q_1^2 + p_2^2q_2^2 + 2p_1q_1 + 2p_2q_2 + 2p_1p_2q_1q_2 + 1.$$

Then,

$$k(p, q) = (\varphi(p)\cdot\varphi(q)) + 1 \qquad (15)$$

The kernel functions complete the inner product operation after being mapped from the low- into the high-dimensional space. With the kernel functions, mapping variables individually into the high-dimensional space before computing the inner product is unnecessary. This operation can be simply performed in the low-dimensional space using kernel functions. The object function is

$$\max_\alpha \min_{w,b,\varepsilon} L(w, b, \varepsilon, \alpha, \mu) = \min_\alpha \frac{1}{2}\sum\sum \alpha_i\alpha_j y_i y_j (x_i\cdot x_j) - \sum \alpha_i \qquad (16)$$

This object function contains the inner product operation with independent variables. Meanwhile, the inner product operation can be directly computed in the low-dimensional space via kernel functions after being mapped into the high-dimensional space. In this manner, kernel functions originate. The commonly used kernel function is the Gaussian kernel, which can map the low- into infinite-dimensional space. After the kernel functions are used, the object function and constraint conditions of the optimization problem become

$$\min_\alpha \frac{1}{2}\sum\sum \alpha_i\alpha_j y_i y_j K(x_i3, x_j) - \sum \alpha_i s.t. \sum \alpha_i y_i = 0$$
$$0 \le \alpha_i \le C \qquad (17)$$

At the beginning, an SVM classifier is trained to display the SVM classes. Fig. 4 shows the result of the abnormal data of iris outlier detection by using the SVM classifier and one-class learning. Figs. 5 and 6 show the scatter plot of iris measurements and the iris classification regions, respectively.

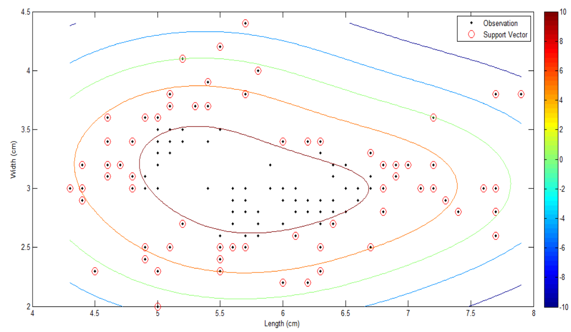


FIGURE 4. Iris outlier detection.

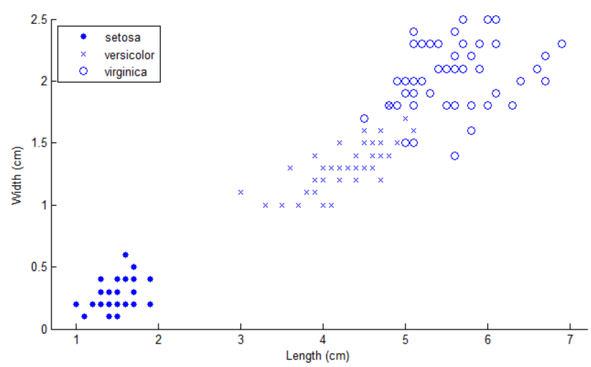


FIGURE 5. Scatter plot of iris measurements.

The SVM classifier is equivalent to the 1-Nearest Neighbor (1NN) classifier, which only takes one representative point in every class of support vectors. The nearest neighbor method is one of the most important nonparametric pattern recognition methods. 1NN takes all training samples as representative points; thus, in the classification, 1NN needs to calculate the distance from sample x to be recognized to all training samples, and the result is the class to which the nearest training samples to x belong. KNN is the expansion of 1NN.

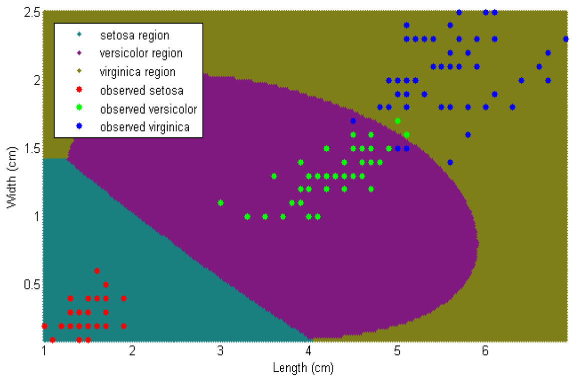


FIGURE 6. Iris classification regions.

Thus, it selects k nearest neighbors of x in the classification and places x into the class to which most of k neighbors belong [17], [18].

IV. PROPOSED ALGORITHM AND CLASSIFICATION PROCESS

A. IMAGE CLASSIFICATION BASED ON COLOR FEATURES

Color usually plays an important role in human perception to the environment and objects. In many cases (especially for natural scenes), color is the most convenient and effective feature to describe an image. In comparison with other low-level features, color feature has stronger robustness against image scaling, rotation, occlusion, and other shape change. The strengths of color feature make it the most widely used low-level feature adopted in content-based image retrieval.

Recognition algorithms based on color feature are diverse. Among them, color statistical histogram is the most frequently used. For a given image $(f_{xy})_{M\times N}$, $M\times N$ is the size of the image, f_{xy} is the color value of different pixels, and C are all colors contained in the image. Then, the color histogram of the image is defined as

$$hc = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\delta(f_{ij}-c) \quad \forall c\in C \tag{18}$$

In color histogram, the most important idea is as follows. First, the color space is discretized, and the frequency of every type of color appearing in the image is then calculated by means of statistics. Color histogram can extract image features and compute the similarity of them handily; in this way, the impact on the change of image size is quite small.

The color distribution in the image is represented by the use of moment, where no quantitative processing shall be conducted on the colors; hence, the image processing is considerably simplified. The image colors only need to be distributed in the low-order moments (i.e., first-, second-, and third-order), which can be expressed as follows:

$$\mu_i = \frac{1}{N}\sum_{j-1}^{N} p_{ij} \tag{19}$$

$$\sigma_i = (\frac{1}{N}\sum_{j-1}^{N}(p_{ij}-\mu_i)^2)^{\frac{1}{2}} \tag{20}$$

$$s_i = (\frac{1}{N}\sum_{j-1}^{N}(p_{ij}-\mu_i)^3)^{\frac{1}{2}} \tag{21}$$

Let p_{ij} be the explicit probability value of the pixel with a grayscale numerical value of in the ijth color component of the image in the process of image recognition. N represents the grayscale technology. In the same image, nine components constitute the color moment related to recognition.

B. SVM MULTICLASS CLASSIFICATION ALGORITHM BASED ON DE AND KNN

This paper divides the multiclass training samples into two classes in every node by means of DE to make the separability of these classes as strong as possible. It constructs the rational classification tree and finally generates the best (or better) classification tree. DE is designed as follows.

Let the number of evolutionary generations be t, the population size be NP, the number of spatial dimensions be D, the current population be $X(t) = \{x_1^t, x_2^t, \cdots, x_{NP}^t\}$, and the ith individual in the population be $x_i^t = (x_{i1}^t, x_{i2}^t, \cdots, x_{iD}^t)^T$. In the evolution, three operations are successively performed on every individual x_i^t.

For every individual x_i^t, the mutated individual $v_i^t = (v_{i1}^t, v_{i2}^t, \cdots, v_{iD}^t)^T$ is generated according to the following formula:

$$v_{ij}^t = x_{r_1j}^t + F(x_{r_2j}^t - x_{r_3j}^t) \quad j = 1, 2, \cdots, D \tag{22}$$

where $x_{r_1}^t = (x_{r_11}^t, x_{r_12}^t, \cdots, x_{r_1D}^t)^T$, $x_{r_2}^t = (x_{r_21}^t, x_{r_22}^t, \cdots, x_{r_2D}^t)^T$, and $x_{r_3}^t = (x_{r_31}^t, x_{r_32}^t, \cdots, x_{r_3D}^t)^T$ are three individuals randomly selected from the population, and $r_1 \neq r_2 \neq r_3 \neq i$; $x_{r_1j}^t$, $x_{r_2j}^t$, and $x_{r_3j}^t$ are the j-dimensional component of individuals r_1, r_2, and r_3 respectively; and F is the mutagenic factor, the value of which is usually within the scope of [0, 2]. In this manner, the mutated individual v_i^t is obtained.

The experimental individual $u_i^t = (u_{i1}^t, u_{i2}^t, \cdots, u_{iD}^t)^T$ is obtained from the mutated individual v_i^t and the parent individual $\mathbf{x_i^t}$. Then,

$$u_{ij}^t = \begin{cases} v_{ij}^t & \text{if } rand[01] \le CR \text{ or } j = j\_rand \\ x_{ij}^t & \text{if } rand[01] > CR \text{ and } j \neq j\_rand \end{cases} \tag{23}$$

where $rand[0, 1]$ is a random number within the scope of [0, 1]; and CR is a constant within the scope of [0, 1] and is called the crossover factor. A large CR indicates a high crossover probability. j_rand is an integer randomly selected within the scope of $[1, D]$; it guarantees that the experiment individual u_i^t obtains at least one element from the mutated individual v_i^t.

DE uses "greedy" selection strategy to select the individual with the best fitness from the parent individual x_i^t and the experiment individual u_i^t as the individual $\mathbf{x_i^{t+1}}$ in the next generation. The selection operation is

$$x_i^{t+1} = \begin{cases} x_i^t & \text{if } fitness\ (x_i^t) < fitness(u_i^t) \\ u_i^t & \text{otherwise} \end{cases} \tag{24}$$

where $fitness(\cdot)$ is the fitness function. Generally, the object function to be optimized is the fitness function.

First, it calculates the distance between the test data and every training data. Then, it sorts on the basis of the progressive ascending relationship of the distances, selects the K points with the minimum distance, ad determines the appearance frequency of the class to which the top K points belong. Thereafter, it takes the class with the highest appearance frequency of the top K points as the predicted class of the test data.

The distance between the two samples in the feature space reflects the similarity of two samples. The feature space of KNN is usually the n-dimensional real-number vector space R^n. The distance it uses is the Euclidean distance or another type of distance, such as the L_p or Minkowski distance.

Let the feature space X be the n-dimensional real-number vector space R^n, $x_i, x_j \in X$, $x_i = (x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(n)})^T$, $x_j = (x_j^{(1)}, x_j^{(2)}, \cdots, x_j^{(n)})^T$, and the L_p distance between x_i, x_j is defined as

$$L_p(x_i, x_j) = (\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p)^{\frac{1}{p}} \tag{25}$$

where $p \geq 1$.

When $p = 1$, the distance is called the Manhattan distance, that is,

$$L_1(x_i, x_j) = \sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}| \tag{26}$$

When $p = 2$, the distance is called the Euclidean distance, that is,

$$L_2(x_i, x_j) = (\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^2)^{\frac{1}{2}} \tag{27}$$

When $p = \infty$, the distance is the maximum of every coordinate distance, that is,

$$L_\infty(x_i, x_j) = \max_l(x_i^{(l)} - x_j^{(l)}) \tag{28}$$

C. NOVEL IMAGE CLASSIFICATION ALGORITHM

The main steps of the image classification algorithm designed in this paper are as follows.

Input: Training image set $D = \{(B_1, y_1), \ldots, (B_N, y_N)\}$ (where $y_i \in \{-1, +1\}$ represents the concept label) and K value.

Output: SVM classifier $(a*, b*)$ and projection space R.

Let M be the set of classification samples, k be the number of KNN, set S be the set of classifiers corresponding to the classes with poor separability and low classification recognition rate obtained by training, and ε be the classification threshold. ε is usually set from 0.8 to 1.2.

Step 1: Generate the visual semantics and construct the projection space. Combine the visual features x_{ij} corresponding to all segmentation regions of image B_i in D and mark them as $Inset = \{x_t | t = 1, 2, \ldots, T\}$, where $T = \sum_{i=1}^{N} n_i$ represents the total number of visual features.

Step 2: Cluster all feature vectors in S into K classes and refer to cluster center v_k as the visual semantics and the set composed by K visual semantics as the projection space, marked as $P_r = \{v_k | k = 1, 2, \ldots, K\}$.

Step 3: If x traverses the classifier $S_i \notin S$ in the classification, then directly determine which class x belongs to according to the discrimination formula and turn to Step 6; otherwise proceed to Step 4.

Step 4: Establish $g(x) = \sum_i y_i \alpha_i^* K(x_i, x) + b^*$. If $|g(x)| \geq \varepsilon$, then directly determine which class x belongs to according to the discrimination formula $f(x) = sgn\left\{\sum_i y_i \alpha_i^* K(x_i, x) + b^*\right\}$ and turn to Step 6; otherwise, proceed to Step 5.

Step 5: Take the set S_i^{sv} of support vectors of classifier S_i as the set of representative points of KNN. Calculate the distance between the samples and every support vector. Sort d_i from small to large and take the top k neighbors of x.

x belongs to the class to which most of k neighbors belong. Then, proceed to Step 6.

Step 6: Some samples in the sample space are selected as training samples using the constructed SVM model for training. The trained SVM model and the KNN algorithm are used as membership functions to construct the SVM model.

Step 7: For the trained SVM model, the image in the space is segmented by the sample space constructed in Steps 2–5 and its membership degree.

Step 8: Determine whether the termination condition is satisfied (i.e., the maximum number of iterations or the minimum fitness threshold). If so, then output the optimal solution; otherwise, return to Step 3.

V. EXPERIMENTAL RESULTS AND ANALYSIS

Several different experiments are designed in this section to test the performance of the proposed algorithm.

On the basis of the analysis of the error rate of KNN, it is lower than the nearest neighbor method when $N \rightarrow \infty$. The upper and lower bounds of the nearest neighbor method and KNN are within one- and two-fold error rate of the Bayesian decision method, as shown in Fig. 7. Fig. 8 shows the classification demonstration of the nearest neighbor and KNN methods.

A severe problem of the nearest neighbor is that it requires to store all training samples and it has heavy distance computation. Two improved methods are available; one is to organize and sort out the sample set, divide them into different groups and layers, compress the computation to a small range close to the neighbor of the test sample as possible, and avoid blindly computing the distance with every sample in the set;

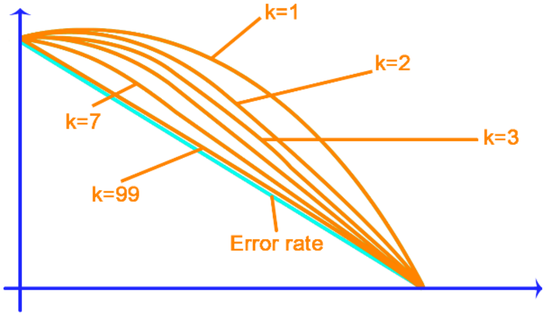
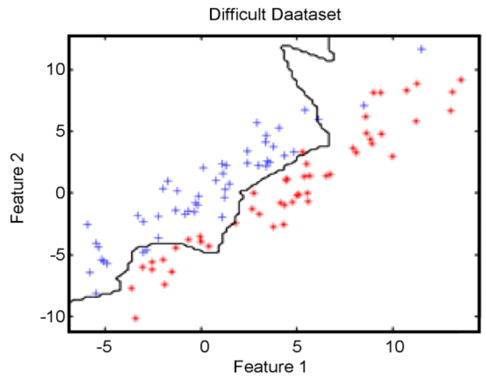
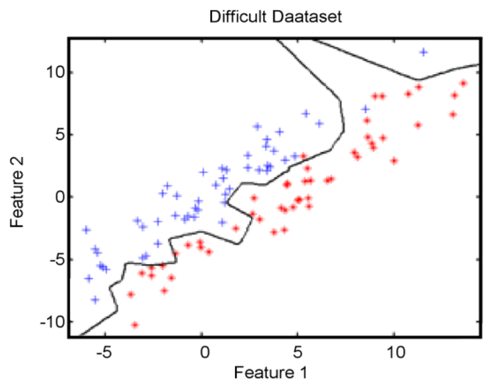


FIGURE 7. Analysis of error rate of KNN.



(a) 3-NN



(b) Nearest neighbor

FIGURE 8. Classification demonstration of nearest neighbor and KNN.

and the other is to select the in-force samples for classification computation from the original sample set to rationally reduce the total number of samples, thereby cutting the computation cost and save storage amount.

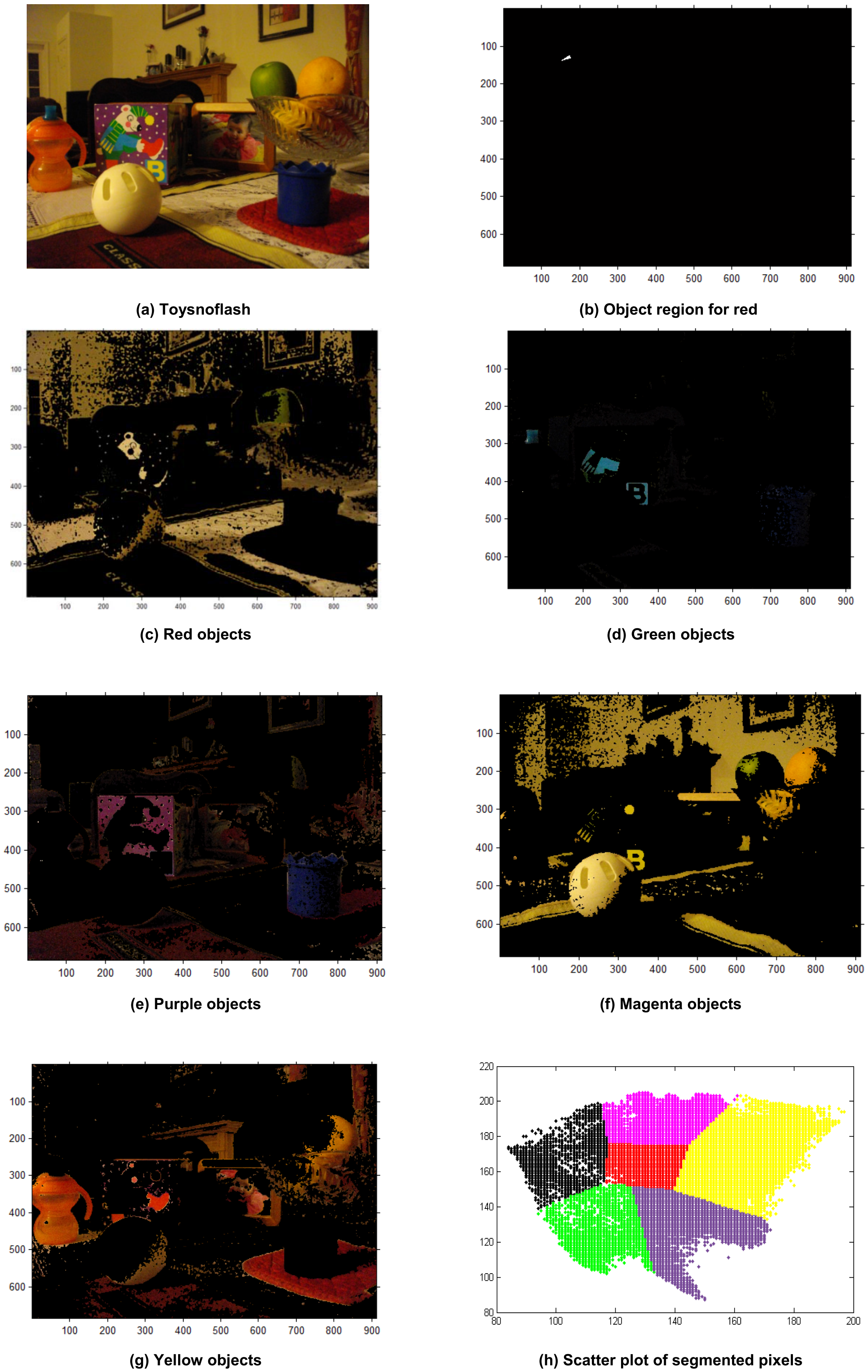
For the linearly inseparable data, the inseparable data in the low-dimensional space are mapped into the high-dimensional space, and SVM increases the generalization capability of the learning machine on the basis of SRM. Thus, a small error is obtained from the limited samples in the training set while maintaining a small error for the independent test set. The multiclass classification strategy adopted in this paper is a one-to-many strategy. For K-class classification, a one-to-many strategy needs to train one SVM for every class, that is, training the K sub-SVM. Hence, to determine the class of a certain object, the objects are classified individually with the trained K sub-SVM; and the samples in the ith class are trained in the ith classifier as the positive samples, whereas the rest are taken as negative samples. In this manner, the output classification result is obtained, and the class corresponding to the SVM with the maximum output value is selected as the class of this object.

This paper tests using test standard images "Toysnoflash" and "Office_4". First, six color regions are selected as the color center on the basis of the prior knowledge. Then, the gray difference between the color value of every pixel and different color centers is calculated. The color value of red, green, and blue (RGB value) of every pixel is the coordinate in the three-dimensional spatial coordinate system, and the nearest Euclidean distance in the three-dimensional space is the discrimination basis. Every RGB value is divided into different color centers, and the corresponding region of every color center is finally accurately segmented. Figs. 9 and 10 show the color classification results of images Toysnoflash and Office_4, respectively.

The experimental results show that the proposed algorithm can distinguish different colors expediently, and the color classification achieves good results. For the input sample, if the classification can be conducted in the upper level of the classification tree, then the discrimination function is directly calculated and the class of the input samples is obtained; otherwise, the input samples are substituted into the lower-level classifier of the classification tree. However, the classification accuracy is rather low due to the poor between-class separability in the lower-level of the classification tree. Then, before using these classifiers in the discrimination, a classification threshold is provided. If the absolute value of the classification discrimination function is larger than the classification threshold, then the class of the input samples is obtained directly through the discrimination function; otherwise, KNN is used to determine the class.

KNN determines the class to which the samples to be classified belong on the basis of the nearest sample or samples. In the classification decision, KNN is only related to few neighbors. Generally, it automatically generates the best (or better) classification tree and then separates the classes that can be easily separated, and followed by those that are less easily separated. However, in the classification using this method, the closer to the nodes in the lowest level of the classification tree, the worst between-class separability and the lowest classification recognition rate will be. To solve this problem, this paper uses the classification tree structure obtained from training and performs classification by combining SVM and KNN in the nodes of the classification tree with poor between-class separability.

The proposed method does not need prior knowledge of specific problems. With limited training samples, it can control well the promotion capability of learning machine; essentially, it solves convex planning problems or the quadratic



(a) Toysnoflash

(b) Object region for red

(c) Red objects

(d) Green objects

(e) Purple objects

(f) Magenta objects

(g) Yellow objects

(h) Scatter plot of segmented pixels

FIGURE 9. Color classification result of image toysnoflash.

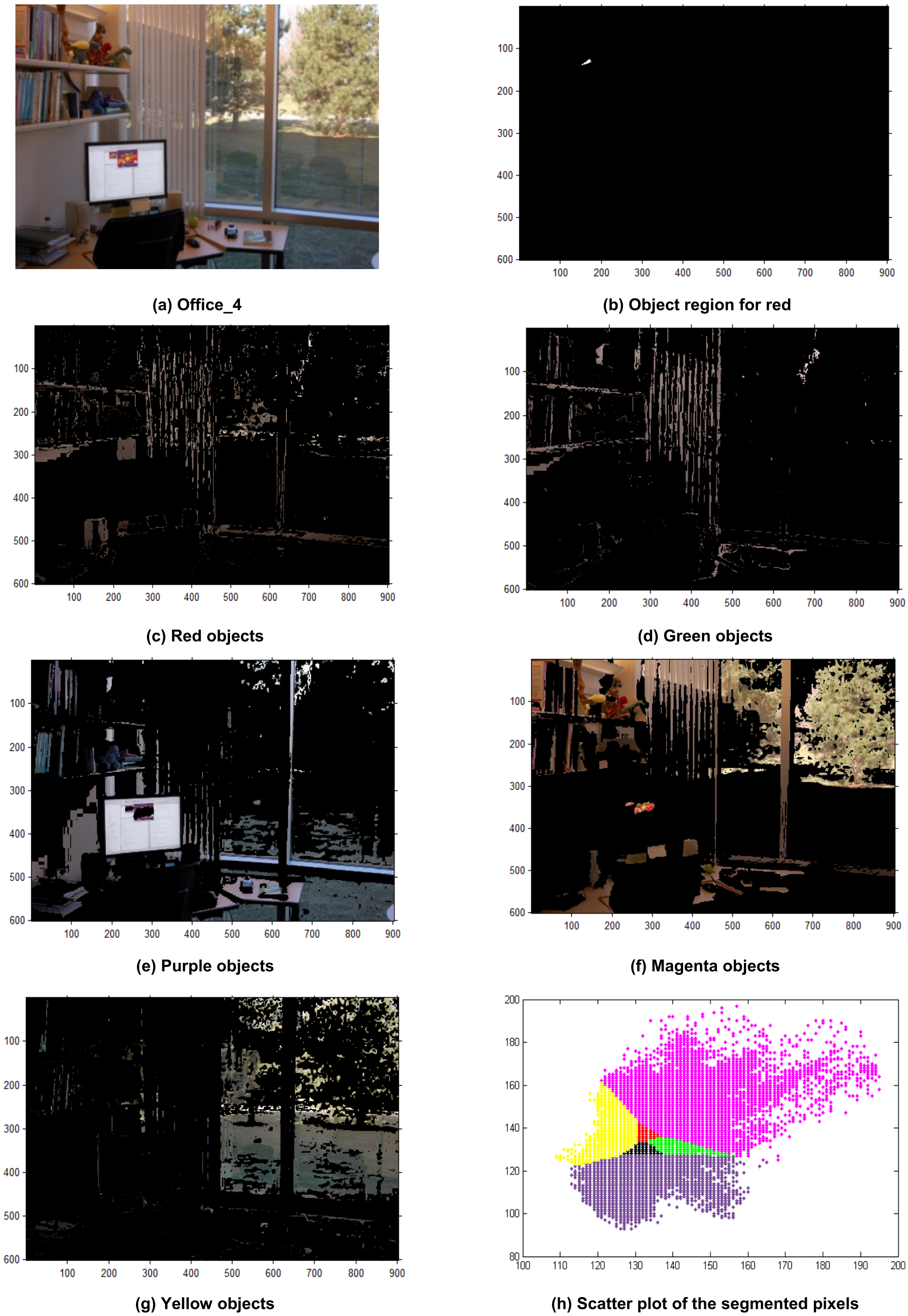


FIGURE 10. Color classification result of image office_4.

programming of dual problems. For the most fundamental two-class classification problem, two types of SVM can be used, namely, linearly separable SVM and linearly inseparable SVM. The SVM maps the samples in the input space into a feature space, which may be high-dimensional, through certain nonlinear function relationship to make two-class samples linearly separable in this feature space and (i.e., can be expanded to multiclass samples). It also searches the optimal linear classification hyperplane in this feature space for the samples.

VI. CONCLUSION

This paper investigates the basic theory and implementation approaches of SVM. SVM is a convex optimization problem; thus, its local optimal solution must be its global optimal solution, which is beyond the capability of other learning algorithms. SVM is the optimal method for two-class problems with small samples and for multiclass problems. It also constructs multiple two-class SVM classifiers. At present, the construction of SVM multiclass classifier has become a research hotspot. This paper optimizes the SVM tree structure through DE, generates the optimal tree structure, and performs image classification. In the optimization of SVM classification model parameters based on DE, the image features extracted are taken as the input of the classifier, image classification is realized, and the influence of the optimization method with different features and kernel parameters on classification performance is analyzed. The simulation result proves that the proposed method works effectively.

REFERENCES

[1] Z. Ma, Q. Liu, K. Sun, and S. Zha, "A syncretic representation for image classification and face recognition," *CAAI Trans. Intell. Technol.*, vol. 1, no. 2, pp. 173–178, Apr. 2016.

[2] K. Cai, R. Yang, L. Li, S. Ou, Y. Chen, and J. Dou, "A semi-automatic coronary artery segmentation framework using mechanical simulation," *J. Med. Syst.*, vol. 39, no. 10, p. 129, Oct. 2015.

[3] R. F. Murray, "Classification images in a very general decision model," *Vis. Res.*, vol. 123, pp. 26–32, Jun. 2016.

[4] C. M. Alaíz and J. A. K. Suykens, "Modified Frank–Wolfe algorithm for enhanced sparsity in support vector machine classifiers," *Neurocomputing*, vol. 320, pp. 47–59, Dec. 2018.

[5] R. Panja and N. R. Pal, "MS-SVM: Minimally spanned support vector machine," *Appl. Soft Comput.*, vol. 64, pp. 356–365, Mar. 2018.

[6] M. D. de Lima, N. L. Costa, and R. Barbosa, "Improvements on least squares twin multi-class classification support vector machine," *Neurocomputing*, vol. 313, no. 3, pp. 196–205, Nov. 2018.

[7] Z. Huang, J. Tang, G. Shan, J. Ni, Y. Chen, and C. Wang, "An efficient passenger-hunting recommendation framework with multi-task deep learning," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2019.2901759.

[8] Y. Wang, E. Zio, X. Wei, D. Zhang, and B. Wu, "A resilience perspective on water transport systems: The case of eastern star," *Int. J. Disaster Risk Reduction*, vol. 33, pp. 343–354, Feb. 2019.

[9] K. Cai, R. Q. Yang, H. Z. Chen, L. H. Li, J. Zhou, S. X. Ou, and F. Liu, "A framework combining window width-level adjustment and Gaussian filter-based multi-resolution for automatic whole heart segmentation," *Neurocomputing*, vol. 220, pp. 138–150, Jan. 2017.

[10] H. Chen, H. Qiao, L. Xu, Q. Feng, and K. Cai, "A fuzzy optimization strategy for the implementation of RBF LSSVR model in vis-NIR analysis of pomelo maturity," *IEEE Trans. Ind. Informat.*, to be published. doi: 10.1109/TII.2019.2933582.

[11] X. Li, M. Fang, H. Wang, and J.-J. Zhang, "Supervised transfer kernel sparse coding for image classification," *Pattern Recognit. Lett.*, vol. 68, no. 12, pp. 27–33, Dec. 2015.

[12] S. Maldonado, J. Merigó, and J. Miranda, "Redefining support vector machines with the ordered weighted average," *Knowl.-Based Syst.*, vol. 148, pp. 41–46, May 2018.

[13] X. Shen, L. Niu, Z. Qi, and Y. Tian, "Support vector machine classifier with truncated pinball loss," *Pattern Recognit.*, vol. 68, no. 8, pp. 199–210, Aug. 2017.

[14] L. Liu, L. Shao, and X. Li, "Evolutionary compact embedding for large-scale image classification," *Inf. Sci.*, vol. 316, no. 9, pp. 567–581, Sep. 2015.

[15] M. L. D. Dias and A. R. R. Neto, "Training soft margin support vector machines by simulated annealing: A dual approach," *Expert Syst. Appl.*, vol. 87, pp. 157–169, Nov. 2017.

[16] M. Zhang, D. Zhang, F. Goerlandt, X. Yan, and P. Kujala, "Use of HFACS and fault tree model for collision risk factors analysis of icebreaker assistance in ice-covered waters," *Saf. Sci.*, vol. 111, pp. 128–143, Jan. 2019.

[17] M. R. Phangtriastu, J. Harefa, and D. F. Tanoto, "Comparison between neural network and support vector machine in optical character recognition," *Procedia Comput. Sci.*, vol. 116, pp. 351–357, Oct. 2017.

[18] M. Rasmussen, J. Rieger, and K. N. Webster, "Approximation of reachable sets using optimal control and support vector machines," *J. Comput. Appl. Math.*, vol. 311, no. 2, pp. 68–83, Feb. 2017.

[19] B. B. Damodaran and R. R. Nidamanuri, "Assessment of the impact of dimensionality reduction methods on information classes and classifiers for hyperspectral image classification by multiple classifier system," *Adv. Space Res.*, vol. 53, no. 12, pp. 1720–1734, Jun. 2014.

[20] F.-J. González-Serrano, Á. Navia-Vázquez, and A. Amor-Martín, "Training support vector machines with privacy-protected data," *Pattern Recognit.*, vol. 72, no. 12, pp. 93–107, Dec. 2017.

[21] S. Uhlmann and S. Kiranyaz, "Classification of dual- and single polarized SAR images by incorporating visual features," *ISPRS J. Photogramm. Remote Sens.*, vol. 90, no. 4, pp. 10–22, Apr. 2014.

[22] B. Wu, L. Zong, X. Yan, and C. G. Soares, "Incorporating evidential reasoning and TOPSIS into group decision-making under uncertainty for handling ship without command," *Ocean Eng.*, vol. 164, pp. 590–603, Sep. 2018.

[23] J. Wang, J. Luo, X. Liu, Y. Li, S. Liu, R. Zhu, and A. Anjum, "Improved Kalman filter based differentially private streaming data release in cognitive computing," *Future Gener. Comput. Syst.*, vol. 98, no. 9, pp. 541–549, Sep. 2019.

[24] K. Cai, Y. Rongqian, L. Li, Z. Xie, S. Ou, Y. Chen, and J. Dou, "Tetralogy of fallot cardiac function evaluation and intelligent diagnosis based on dual-source computed tomography cardiac images," *Artif. Organs*, vol. 40, no. 5, pp. 459–469, May 2016.

[25] B. Jalaeian, R. Zhu, H. Samani, and M. Motani, "An optimal cross-layer framework for cognitive radio network under interference temperature model," *IEEE Syst. J.*, vol. 10, no. 1, pp. 293–301, Mar. 2016.

[26] K. Cai, R. Yang, H. Yue, L. Li, S. Ou, and F. Liu, "Dynamic updating atlas for heart segmentation with a nonlinear field-based model," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 13, no. 3, Sep. 2017, Art. no. e1785.

[27] R. Zhu, X. Zhang, X. Liu, W. Shu, T. Mao, and B. Jalaian, "ERDT: Energy-efficient reliable decision transmission for cooperative spectrum sensing in industrial IoT," *IEEE Access*, vol. 3, pp. 2366–2378, 2015.

[28] Q. Ke, J. Zhang, M. Wozniak, and W. Wei, "The phase and shift-invariant feature by adaptive independent subspace analysis for cortical complex cells," *Inf. Technol. Control*, vol. 48, no. 1, pp. 58–70, Mar. 2019.

WANNENG SHU (M'81) received the B.E. degree in computer science and technology from Jianghan University, Wuhan, China, in 2003, the M.E. degree in computer application technology from Central China Normal University, Wuhan, in 2007, and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, in 2013.

He is currently an Associate Professor with South-Central University for Nationalities, Wuhan. His research interest includes the wireless communication, smart city, green cloud computing, machine learning, and intelligent computing.

KEN CAI received the B.S. degree in measurement control technology and instruments and the M.S. degree in measuring and testing technologies and instruments from the Guangdong University of Technology, Guangzhou, China, in 2003 and 2006, respectively, and the Ph.D. degree in biomedical engineering from the South China University of Technology, Guangzhou, in 2011.

He is currently an Associate Professor with the College of Automation, Zhongkai University of Agriculture and Engineering, Guangzhou. His main research interests include machine vision and biomedical instrument.