Image and Vision Computing 29 (2011) 29–40

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

A modified support vector machine and its application to image segmentation

Zhiwen Yu [a,*], Hau-San Wong [b], Guihua Wen [a]

[a] The School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[b] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 23 April 2009
Received in revised form 15 July 2010
Accepted 5 August 2010

Keywords:
Support vector machine
Image segmentation
Classification

ABSTRACT

Recently, researchers are focusing more on the study of support vector machine (SVM) due to its useful applications in a number of areas, such as pattern recognition, multimedia, image processing and bioinformatics. One of the main research issues is how to improve the efficiency of the original SVM model, while preventing any deterioration of the classification performance of the model. In this paper, we propose a modified SVM based on the properties of support vectors and a pruning strategy to preserve support vectors, while eliminating redundant training vectors at the same time. The experiments on real images show that (1) our proposed approach can reduce the number of input training vectors, while preserving the support vectors, which leads to a significant reduction in the computational cost while attaining similar levels of accuracy. (2)The approach also works well when applied to image segmentation.

1. Introduction

Image segmentation, the objective of which is to subdivide an image into several meaningful regions, is an important topic in image processing, computer vision and multimedia. The set of segmented regions not only corresponds to the underlying structure of the image, but is also related to the semantic meaning of the image. There are a number of methods to perform image segmentation based on different criteria and different properties of the images. In general, these approaches can be categorized into three types: Supervised learning based image segmentation methods, unsupervised learning based image segmentation methods, and level set based image segmentation methods. (i) Given an image, the supervised learning-based image segmentation method first selects a small set of pixels in different regions to serve as prior knowledge for training a classifier. Then, the remaining pixels in the image are considered as the test set and are partitioned into several meaningful regions by the trained classifier. Examples of the type of classifier used include Fisher Linear Discriminant [1], Support Vector Machine [2] and Bayesian classifier [3]. Many of the supervised learning-based image segmentation methods [4] are applied to face recognition, gait recognition and sign recognition. (ii) Given an image, the unsupervised learning-based image segmentation approaches first estimate the optimal number of regions (K) in the image by a cluster validity index. The cluster validity index used includes the class of internal indices (e.g., Dunn index [5], Davies-Bouldin index [6] and Silhouette index [7]) and the information-based criteria (e.g., Akaike Information Criterion (AIC) [8], Minimum Description Length (MDL) [9] and Bayesian Information Criterion (BIC) [10]). Next, different clustering algorithms, such as K-means, EM (the Expectation Maximization algorithm, [11]) and SOM (self organizing map, [12]), are adopted to partition the image into K regions. Most of the unsupervised learning-based image segmentation methods [13–16] are used for performing semantic image retrieval, automatic image annotation and multi-band image segmentation. (iii) Given an image, the problem of level set-based image segmentation is cast in a minimization framework by selecting a set of suitable criteria, which are encoded as region and boundary functionals in a cost function, to minimize the total cost [17,18]. A number of applications for the level set-based image segmentation methods have been proposed, such as natural image segmentation [19], medical image segmentation [20], leukocyte detection [21], and so on. Different image segmentation approaches are suitable under different conditions. If the coarse contours of the regions are provided by the user, the level set-based image segmentation method represents a good choice. If the user expects to perform image segmentation automatically, it is better to use the unsupervised learning-based image segmentation approaches. If the user specifies a small set of the training pixels, the supervised learning-based image segmentation approaches can first be considered.

In the paper, we focus on the supervised learning-based image segmentation methods. Specifically, we explore a new kind of SVM-based approach called Fast Support Vector Machine (FSVM) to perform image segmentation. In our approach, the user first specifies a small set of training pixels, such as a small part of an object and a small part of the background, as the clues. Then, FSVM is applied to train the classifiers based on the training pixels. Finally, the remaining image, which is viewed as the test set, is subdivided into several regions by the classifier.

* Corresponding author.
E-mail address: kevinyuzhiwen@hotmail.com (Z. Yu).

0262-8856/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2010.08.003

The remainder of the paper is organized as follows. Section 2 introduces the background of SVM. Section 3 describes the newly proposed fast support vector machine approach (FSVM) and its performance analysis. Section 4 applies our proposed approach for image segmentation. Section 5 is the conclusion and future work.

2. Background

Support vector machine is an important topic in many areas [3,22–38], such as pattern recognition, image processing, machine learning and bioinformatics owing to its broad applications. The objective of the SVM approach [22] is to find a hyperplane which maximizes the geometric margin and minimizes the classification error when given a two-class linearly separable problem. Most of the existing SVM approaches can be divided into two categories based on the algebraic view [22–30] and the geometric view [31–38]: (i) the approaches from the algebraic view include sequential minimal optimization (SMO) [24], SVM with soft margin [23], ν-SVM [26], kernel SVM [27], and support vector regression machine [30]. These approaches explore how to minimize the classification error and reduce the computational cost of SVM. (ii) The approaches from the geometric view include SVM with dual representation [32], the iterative nearest point algorithm [34], SVM based on convex hull [35,36], and SVM based on reduced convex hull (RCH) [31,37]. These approaches make use of the geometric properties of SVM to solve the classification problem.

Although researchers have studied a lot of problems related to SVM, most of them overlook one important issue: how to improve the efficiency of SVM while avoiding any deterioration of its classification performance in the scenario when the number of training vectors is large. On one hand, the support vectors, which are used to determine the hyperplane, are the most important training vectors in the SVM approaches and possess. There are two interesting properties: (i) most of the support vectors are located at the boundary of their respective classes, and (ii) the distances between support vectors and the hyperplane are smaller than the distances between other training vectors and the hyperplane. On the other hand, compared with the set of support vector, the contribution of the other training vectors in determining the classifier characteristics is limited. As a result, we design a new approach called fast support vector machine approach (FSVM) based on the properties of support vectors, which incorporates a Gaussian model and a projection process to remove the redundant training vectors for reducing the training time, while preserving the set of support vectors.

3. Fast support vector machine approach

Given a set of training vectors $F_{train} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_n\}$ with corresponding labels $L_{train} = \{l_1, l_2, ..., l_n\}$ ($l_i \in \{1,2\}$), the objective of the fast support vector machine approach (FSVM) is to (i) eliminate the redundant training vectors and (ii) train the classifier by the remaining training vectors. In other words, the difference between FSVM and the existing SVM approaches is that FSVM focuses on reducing the number of redundant training vectors. There are two assumptions which are essential to the FSVM formulation: (i) there exists a convex hull for the input training vectors in each class; (ii) the problem is separable.

Fig. 1 provides an overview of FSVM. Specifically, FSVM first eliminates the training vectors which are close to the center of the class using the Gaussian models. Then, it removes other training vectors by a projection process. Finally, FSVM performs sequential minimal optimization (SMO) on the remaining training vectors to obtain the classifier.

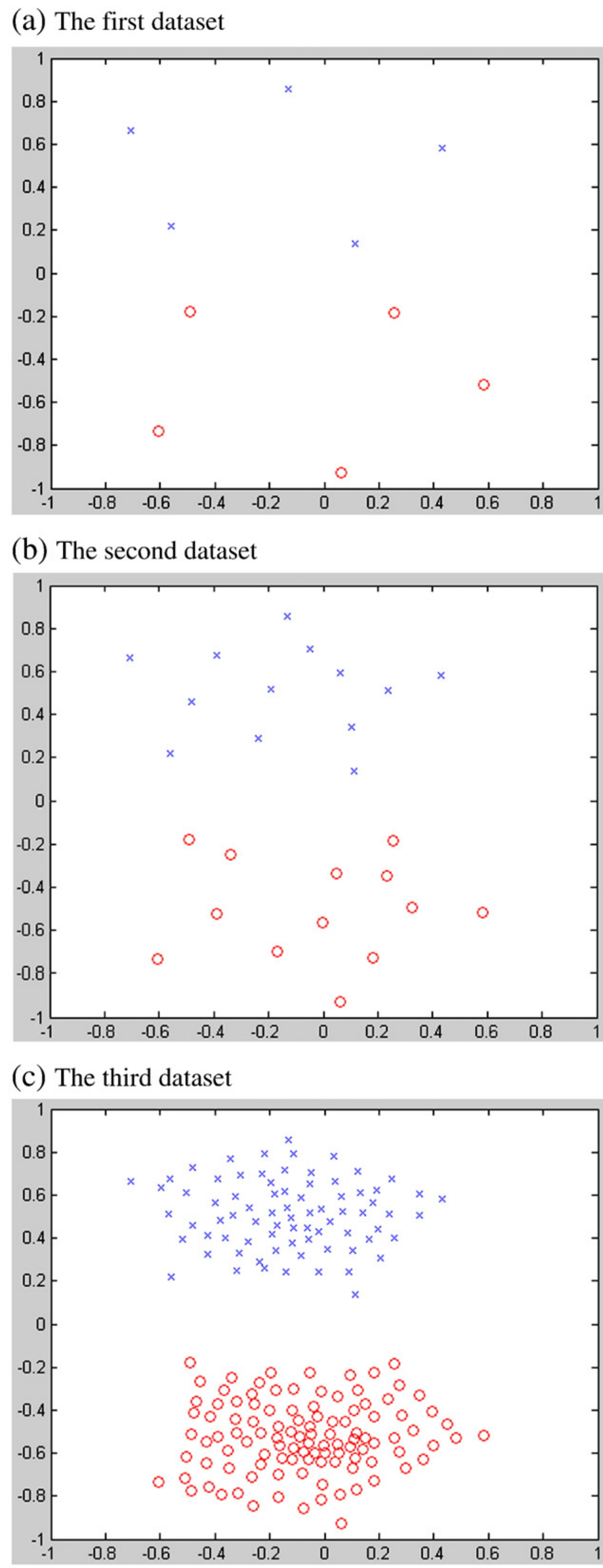

Fig. 2. Examples of k value estimation.

Algorithm FSVM (a set of training vectors F_{train})
1. Eliminate training vectors by the Gaussian models;
2. Eliminate training vectors by the projection process;
3. Perform SMO to obtain the binary SVM classifier;

Fig. 1. An overview of FSVM.

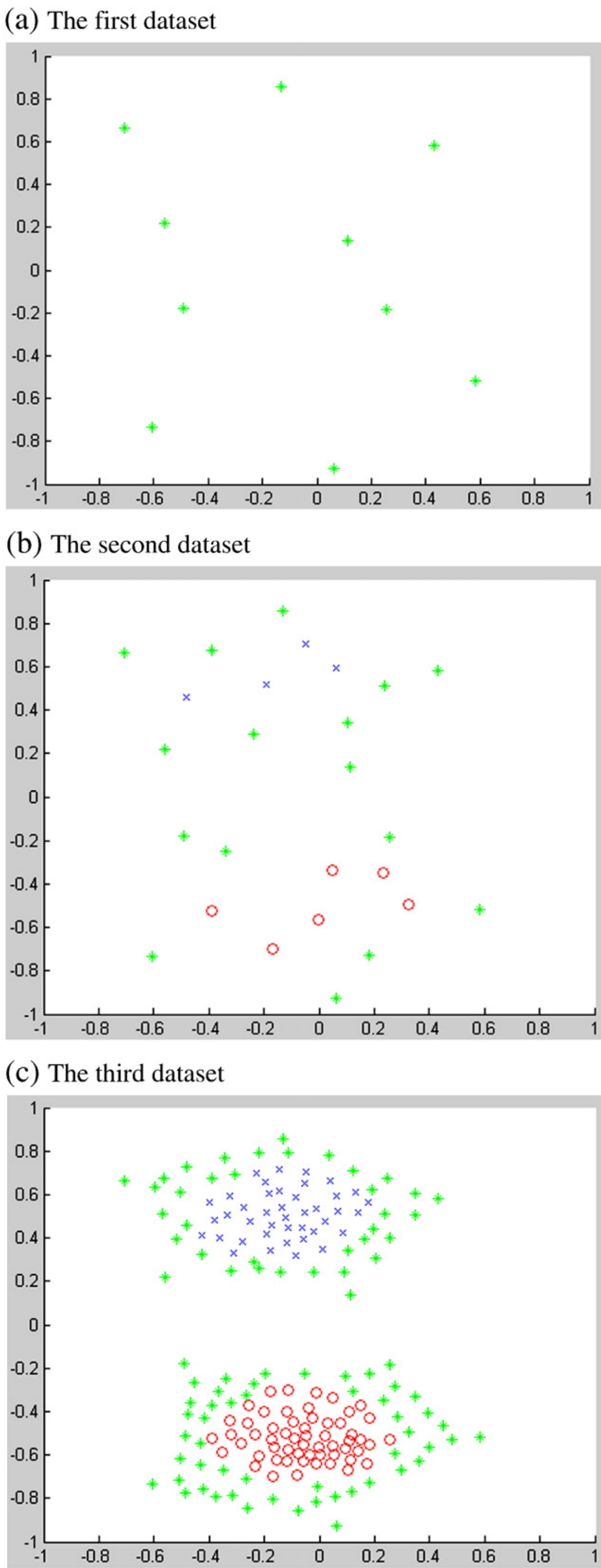

Fig. 3. Estimation of the *k* value.

3.1. Training vector elimination by the Gaussian model

FSVM first estimates the multivariate Gaussian distribution of the input training vectors of each class:

$$G = (\mu, \Sigma) \tag{1}$$

$$\mu = \frac{\sum_{i=1}^{n} \mathbf{f}_i}{n}, \quad \sigma^2 = \frac{\sum_{i=1}^{n} (\mathbf{f}_i - \mu)^2}{n} \tag{2}$$

where μ is the mean of the Gaussian model G, Σ is the $d \times d$ diagonal covariance matrix with σ^2 on its diagonal, and d is the number of dimensions. The value of the input training vector $\mathbf{f}$ with respect to the multivariate Gaussian probability distribution function can be calculated by

$$P(\mathbf{f}) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{|\Sigma|}} e^{\left(-\frac{1}{2}(\mathbf{f}-\mu)^T \Sigma^{-1} (\mathbf{f}-\mu)\right)} \tag{3}$$

One interesting observation of these probability values with respect to G is that the vectors which are close to the center of the Gaussian distribution have larger probability values, while the vectors which are close to the boundary have smaller probability values. FSVM selects k training vectors in each class with the smallest probability values in the next step.

The parameter k is the most important factor for FSVM, and the adoption of different k values will lead to different results. Fig. 2 shows three synthetic data sets. The training vectors in the same class (the blue crosses denote the first class and the red circles denote the second class) in different data sets have the same convex hull, while the numbers of training vectors in the convex hulls are significantly different.

To estimate the k value, FSVM has to determine the number of training vectors located at the boundary of the classes. The k value of the classes in Fig. 2(a) should be equal to 5, which is the number of training vectors in these classes, while the k value of the classes in Fig. 2(b) should be far smaller than the number of training vectors in these classes. Motivated by this observation, we design an estimation algorithm to evaluate the k value.

The algorithm first calculates the average probability value $\overline{P}$ for the training vectors in one of the classes I_{train} according to the following equation:

$$\overline{P} = \frac{\sum_{i=1}^{|I_{train}|} P(\mathbf{f}_i)}{|I_{train}|} \tag{4}$$

where I_{train} is the cardinality of the class.

Then, it computes the minimum average probability value $\overline{P}_{min}$ and the maximum average probability value $\overline{P}_{max}$:

$$\overline{P}_{min} = \frac{\sum_{j=1}^{|I_{min}|} P\left(\mathbf{f}_j\right)}{|I_{min}|}, \quad \overline{P}_{max} = \frac{\sum_{h=1}^{|I_{max}|} P(\mathbf{f}_h)}{|I_{max}|} \tag{5}$$

$$\mathbf{f}_j \in I_{min} = \{\mathbf{f}_i \mid P(\mathbf{f}_i) < \overline{P}, \mathbf{f}_i \in I_{train}\}$$
$$\mathbf{f}_h \in I_{max} = \{\mathbf{f}_i \mid P(\mathbf{f}_i) \geq \overline{P}, \mathbf{f}_i \in I_{train}\} \tag{6}$$

We further define the distribution ratio (R) as follows:

$$R = \frac{\overline{P}_{min}}{\overline{P}_{max}} \tag{7}$$

If most of the training vectors are located at the boundary of the class, $\overline{P}_{min}$ is close to $\overline{P}_{max}$ and R is close to 1. If R is small, most of the training vectors are close to the center of the class.

Next, the estimation algorithm determines the k value based on the distribution ratio (R). If $R \geq \tau$ (τ is a threshold and is set to 0.4 in this paper), k is equal to the number of the training vectors in the class. Otherwise, k is estimated by the following equation:

$$k = \left\lceil \frac{\overline{P} - \overline{P}_{min}}{\overline{P}_{max} - \overline{P}_{min}} \cdot |I_{train}| \right\rceil \tag{8}$$

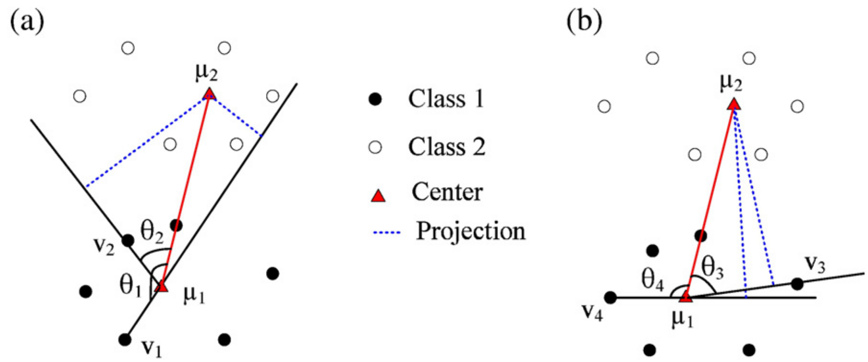

Fig. 4. The projection process.

where $|I_{train}|$ denotes the cardinality of the class I_{train}. By the same approach, we can estimate the k value for the other class. Fig. 3 illustrates the training vectors (red circles) selected by the estimation algorithm. It can be seen that all the training vectors (green asterisks) at the boundary of the classes in the different data sets are preserved, while most of the training vectors (blue crosses and red circles) close to the center of the classes are eliminated.

3.2. Training vector elimination by the projection process

Another interesting observation of the set of support vectors is that the support vectors of one of the classes are always located close to the other class, as illustrated in Fig. 5(b) and (d). Based on this observation, we propose a projection process to further eliminate the redundant training vectors as shown in Fig. 5(e) and (f).

We formulate the projection process as follows: The training vectors are initially divided into two classes I_{train} and J_{train}:

$$\begin{aligned} F_{train} &= I_{train} \cup J_{train} \\ I_{train} &= \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_I\} \\ J_{train} &= \left\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_J\right\} \end{aligned} \tag{9}$$

FSVM first considers the class I_{train} by translating the origin of the coordinate system to the center μ_I of this class:

$$\mathbf{f}_i' = \mathbf{f}_i - \mu_I, \;\; i \in [1, I] \tag{10}$$

Then, the vector $\mu_J\mu_I$ which is a vector joining the centers of the two classes is obtained by the following equation:

$$\mu_J' = \mu_J - \mu_I \tag{11}$$

Next, FSVM projects all the vectors $\mathbf{f}'_i$ ($i \in [1,I]$) on to the vector μ'_J as follows:

$$|\mu_J'|\, cos\theta_i = \frac{\mathbf{f}_i' \cdot \mu_J'}{|\mathbf{f}_i|} \tag{12}$$

where θ_i is the angle between the input vector $\mathbf{f}_i'$ and the vector μ'_J. The following equation is obtained by substituting $(\mathbf{f}_i - \mu_I)$ and $(\mu_J - \mu_I)$ for $\mathbf{f}_i'$ and μ'_J respectively:

$$\left|\left(\mu_J - \mu_I\right)\right| cos\theta_i = \frac{(\mathbf{f}_i - \mu_I) \cdot \left(\mu_J - \mu_I\right)}{|\mathbf{f}_i - \mu_I|} \tag{13}$$

$$\delta\left(\left|\left(\mu_J - \mu_I\right)\right| cos\theta_i\right) = \begin{cases} 1 \;\text{if}\; \left|\left(\mu_J - \mu_I\right)\right| \cos\theta_i \geq 0 \\ 0 \;\text{Otherwise} \end{cases} \tag{14}$$

$\delta(|(\mathbf{f}_i - \mu_I)|\, cos\theta_i) = 1$, the training vector will be preserved. Fig. 4 (a) provides an example of the projection process. The training vector $\mathbf{f}_2$ in Fig. 4(a) will be preserved, since $cos\theta_2 > 0$.

FSVM also preserves two training vectors $\mathbf{f}_{i_1^*}$ and $\mathbf{f}_{i_2^*}$ which satisfy one of the following conditions:

$$\begin{aligned} i_1^* &= arg\;\; min_{i \in [1,I] \;\; and \;\; cos\theta_i \geq 0} \;\; cos\;\; \theta_i \\ i_2^* &= arg\;\; min_{i \in [1,I] \;\; and \;\; cos\theta_i < 0} - cos\;\; \theta_i \end{aligned} \tag{15}$$

The training vector $\mathbf{f}_3$ in Fig. 4(b) will be preserved since $\mathbf{f}_3$ satisfies the first condition, while the training vector $\mathbf{f}_4$ in Fig. 4(b) will be preserved due to its satisfaction of the second condition.

Next, FSVM eliminates all the training vectors which satisfy $\delta(|(\mathbf{f}_i - \mu_I)|\, cos\theta_i) = 0$ but do not satisfy one of the above conditions. Accordingly, the training vector $\mathbf{f}_1$ in Fig. 4(b) is removed from the training set.

By the same approach, a subset of the training vectors in J_{train} can be eliminated from the training set.

3.3. Training the classifier

Finally, FSVM performs sequential minimal optimization (SMO) to compute the discriminant function ($\phi(\mathbf{f}) = \omega^T\mathbf{f} + b$) of the binary SVM classifier using the remaining training vectors F_{remain}. To estimate the optimal parameters (ω^*, b^*) of SVM, SMO needs to solve the following quadratic problem: In order to estimate the optimal parameters (ω^*, b^*) of SVM, SMO needs to solve the following quadratic task:

$$(\omega^*, b^*) = argmin_{\omega, b} \;\; \frac{1}{2}||\omega||^2 \tag{16}$$

subject to

$$l_i\left(\omega^T\mathbf{f}_i + b\right) \geq 1, \mathbf{f}_i \in F_{remain}, l_i \in \{1, -1\} \tag{17}$$

In order to train the parameters of the SVM, solving the above quadratic task should satisfy the Karush–Kuhn–Tucker (KKT) conditions [39]:

$$\frac{\partial}{\partial \omega}\chi(\omega, b, \tau) = 0 \tag{18}$$

$$\frac{\partial}{\partial b}\chi(\omega, b, \tau) = 0 \tag{19}$$

$$\tau_i\left[l_i\left(\omega^T\mathbf{f}_i + b\right)\right] - 1 = 0 \tag{20}$$

where $\tau = \{\tau_1, \tau_2, \ldots, \tau_{n'}\}$ is the vector of the Lagrange multipliers, $i \in \{1, \ldots, n'\}$, n' is the cardinality of the training set F_{remain}, and $\chi(\omega, b, \tau)$ is the Lagrange function which is defined as

$$\chi(\omega, b, \tau) = \frac{1}{2}\omega^T\omega - \sum_{i=1}^{n'} \tau_i\left[l_i\left(\omega^T\mathbf{f}_i + b\right)\right] - 1 \tag{21}$$

In this way, FSVM obtains a binary classifier from the remaining training vectors.

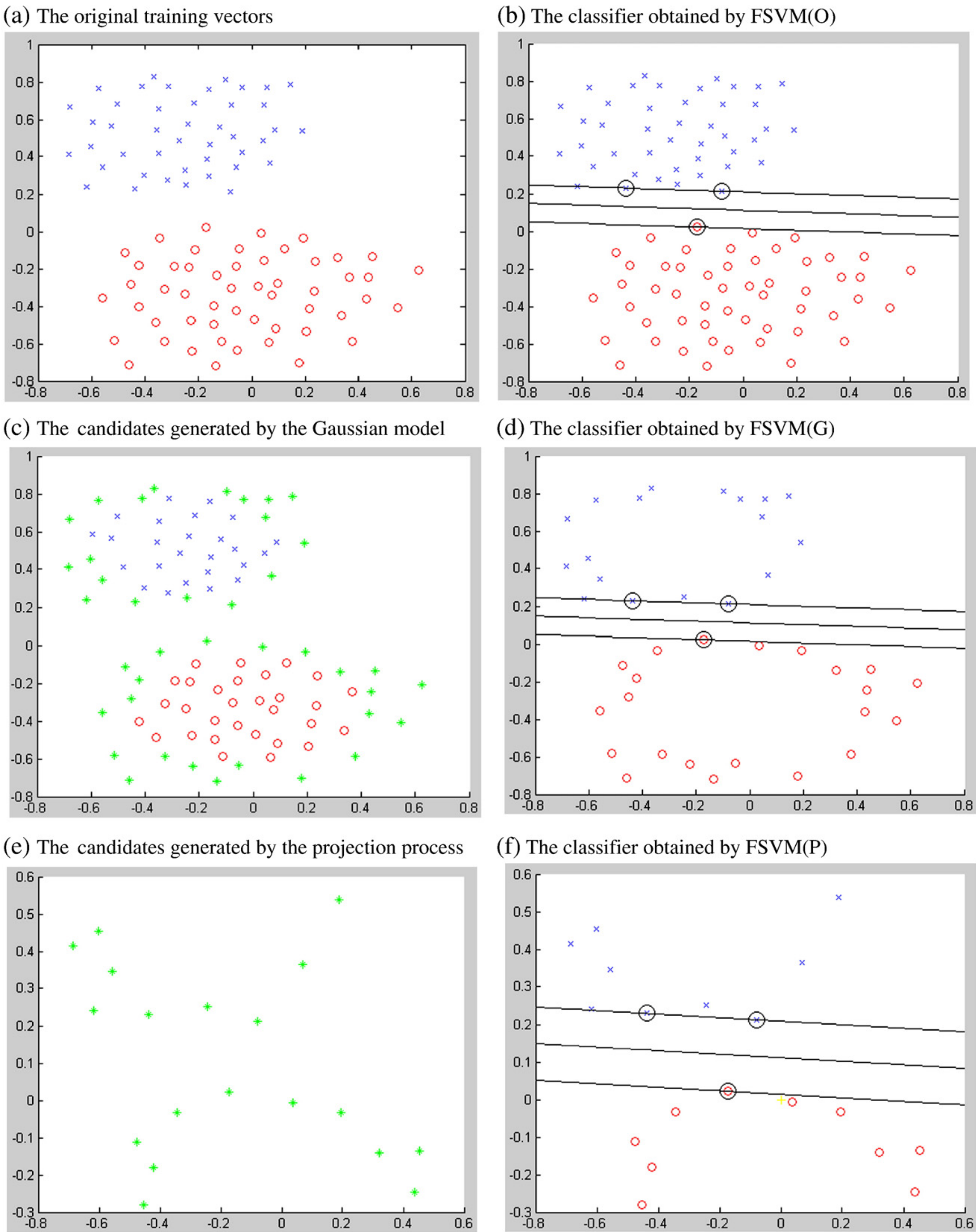

Fig. 5. An example of fast support vector machine approach.

3.4. An example

Fig. 5 shows an example of the FSVM algorithm. Fig. 5(a) shows the original input training vectors, Fig. 5(c) shows the training vectors after pruning by the Gaussian model, and Fig. 5(e) shows the training vectors after pruning by the projection process. The corresponding binary classifiers obtained by FSVM are shown in Fig. 5(b), (d), and (f), respectively. Although the classifiers are obtained from different training sets with different numbers of training vectors, the classifiers in Fig. 5(b), (d), and (f) are the same due to the use of the same support vectors which results in the same margin as shown in Fig. 6 (d). The pruning process based on the Gaussian model and the

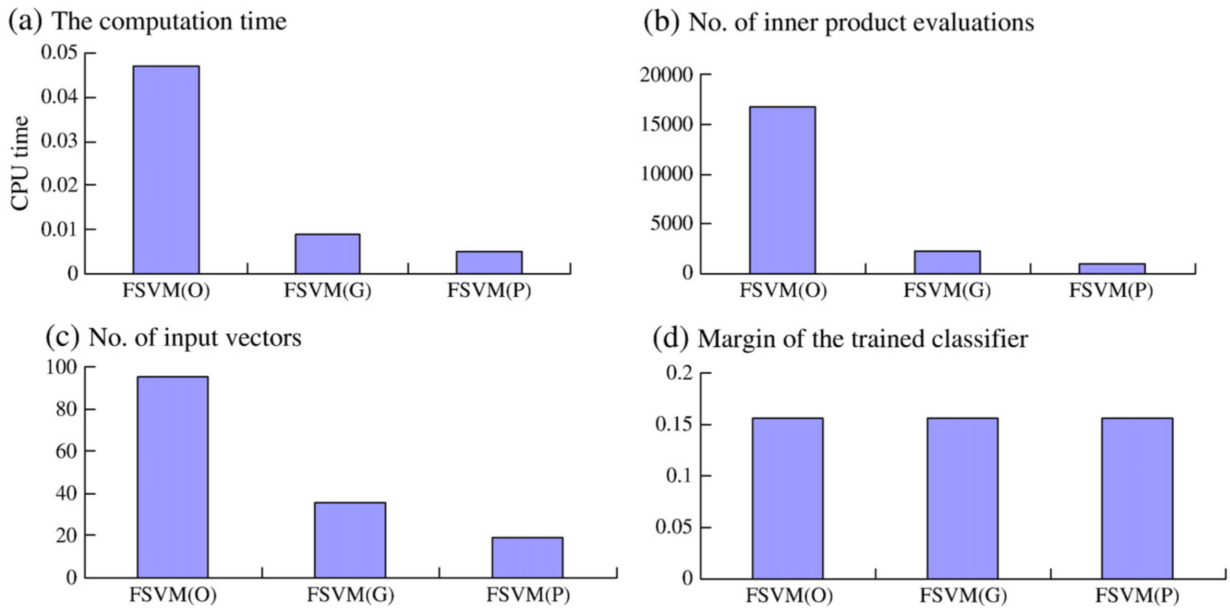

Fig. 6. The performance of the different approaches.

projection step removes most of the training vectors as illustrated in Fig. 6(c), which leads to a significant reduction of the computational cost as shown in Fig. 6(a) and the number of inner product evaluations as shown in Fig. 6(b) (where *FSVM(O)*, *FSVM(G)* and *FSVM(P)* correspond to the cases of no pruning, pruning using the Gaussian model, and pruning using both the Gaussian model and projection process respectively).

3.5. Performance analysis

To analysis the performance of FSVM, we assume (i) the number of input training vectors is n; (ii) the number of selected training vectors by the Gaussian model is k; (iii) the number of remaining vectors after pruning by the projection process is k', where $k' < k \leq n$.

The time complexity of estimating the parameters of the Gaussian model in the first step of FSVM is $O(n)$. In the second step, selecting k vectors from each class takes $O(n + klogk)$ time . The projection in the third step takes up $O(k)$ time. The time complexity of SMO is $O(k' logk')$. As a result, the time complexity of FSVM is $O(n + klogk)$. If k is constant, the time complexity of FSVM will be $O(n)$, which is the lower bound for FSVM. If k is proportional to n, the time complexity of FSVM will be $O(nlogn)$, which corresponds to the upper bound.

4. Experiments

In our experiments, we compare three approaches: FSVM(O) which performs SMO on the original training data set, FSVM(G) which performs SMO on the training data set after pruning by the Gaussian model, and FSVM(P) which performs SMO on the training data set after pruning by the projection process and the Gaussian model. The default setting of the threshold τ for k value estimation is 0.4.

4.1. Experiments on image data sets

The image training data set consists of a set of training vectors with R, G, B values. The pixels in the red rectangle in Fig. 7 are the training vectors, which serve as the prior knowledge for image segmentation. The remaining pixels are considered as the test data set. Our objective is to perform image segmentation based on the prior knowledge which is incorporated into the SVM classifier through training.

Fig. 8 shows the total computation time for image segmentation for the different images. The total computation time consists of two components: the time for training the classifier based on the training data set, and the time for performing image segmentation on the test data set. It can be seen from Fig. 8(a1)–(d1) that the computation time

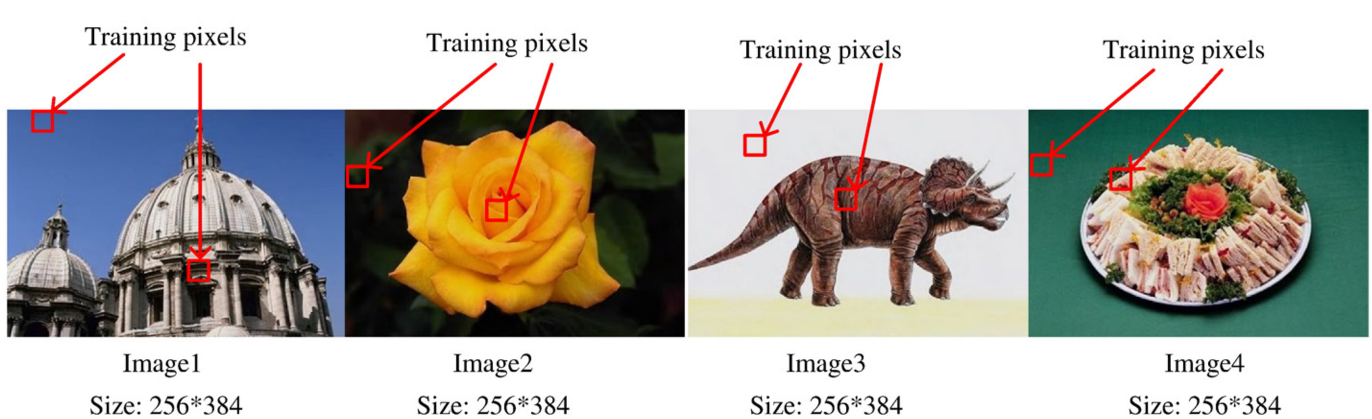

Fig. 7. The training pixels.

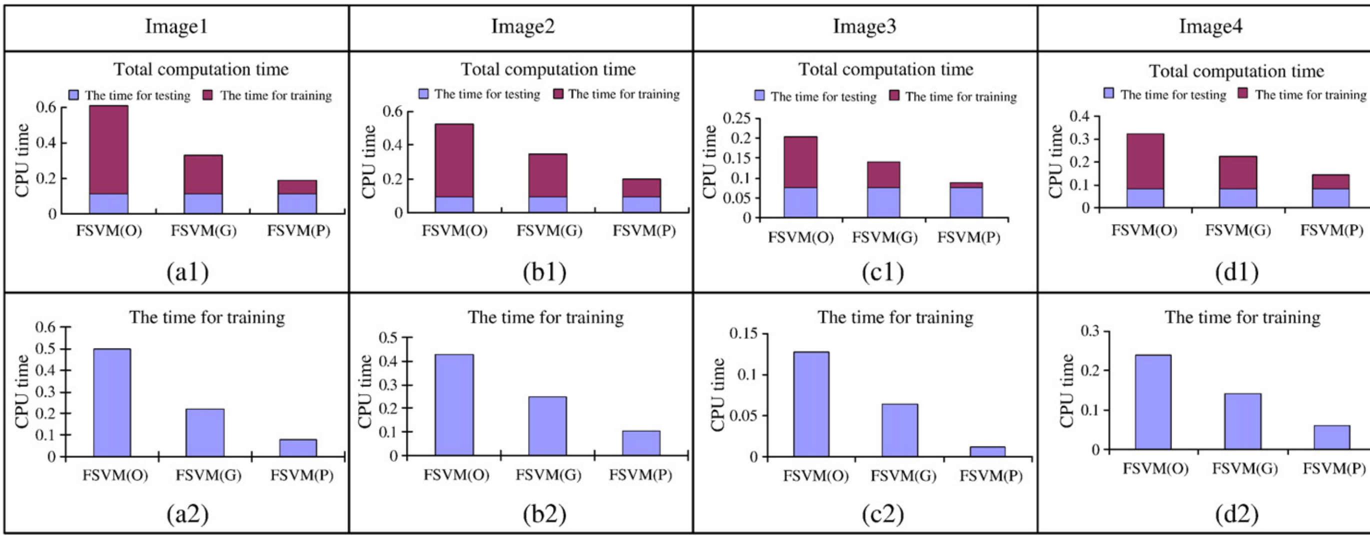

Fig. 8. The total computation time of image segmentation for different images.

for image segmentation on the test data set is a constant, once the classifier is obtained from the training data set. The computation time for training the classifier is sensitive to the size of the training data set and the particular training algorithm used. If the training algorithm is fixed, the computation time for training is only related to the size of the training data set. Fig. 9(a1)–(d1) show the sizes of the training data sets, while Fig. 8(a2)–(d2) show the corresponding training times. If the number of training vectors in the training data set is large, the number of inner product evaluations (Fig. 9(a2)–(d2)) in the process of estimating the parameters of the classifiers will also be large. This leads to a high computational cost for the training process.

Among the three approaches, FSVM(O) results in a high computational cost as shown in Fig. 8(a2)–(d2) since it considers all the training vectors (800 training vectors for each image) in the training set. The performance FSVM(G) is better than that of FSVM(O), since the Gaussian model prunes a large number of training vectors from the training set. The number of training vectors which are eliminated from the training sets is 345, 374, 332 and 304 for Image1, Image2, Image3 and Image4, respectively. Due to

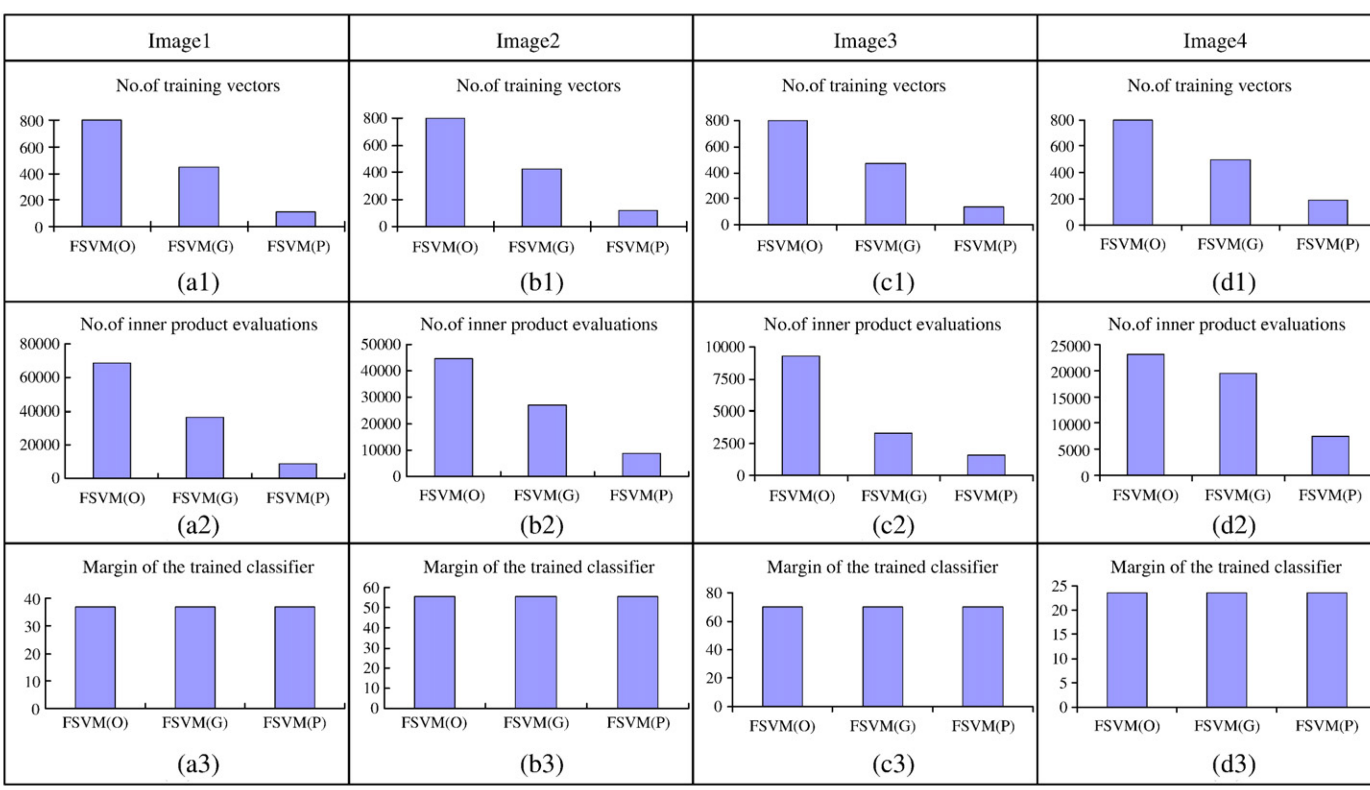

Fig. 9. The performance of the different algorithms.

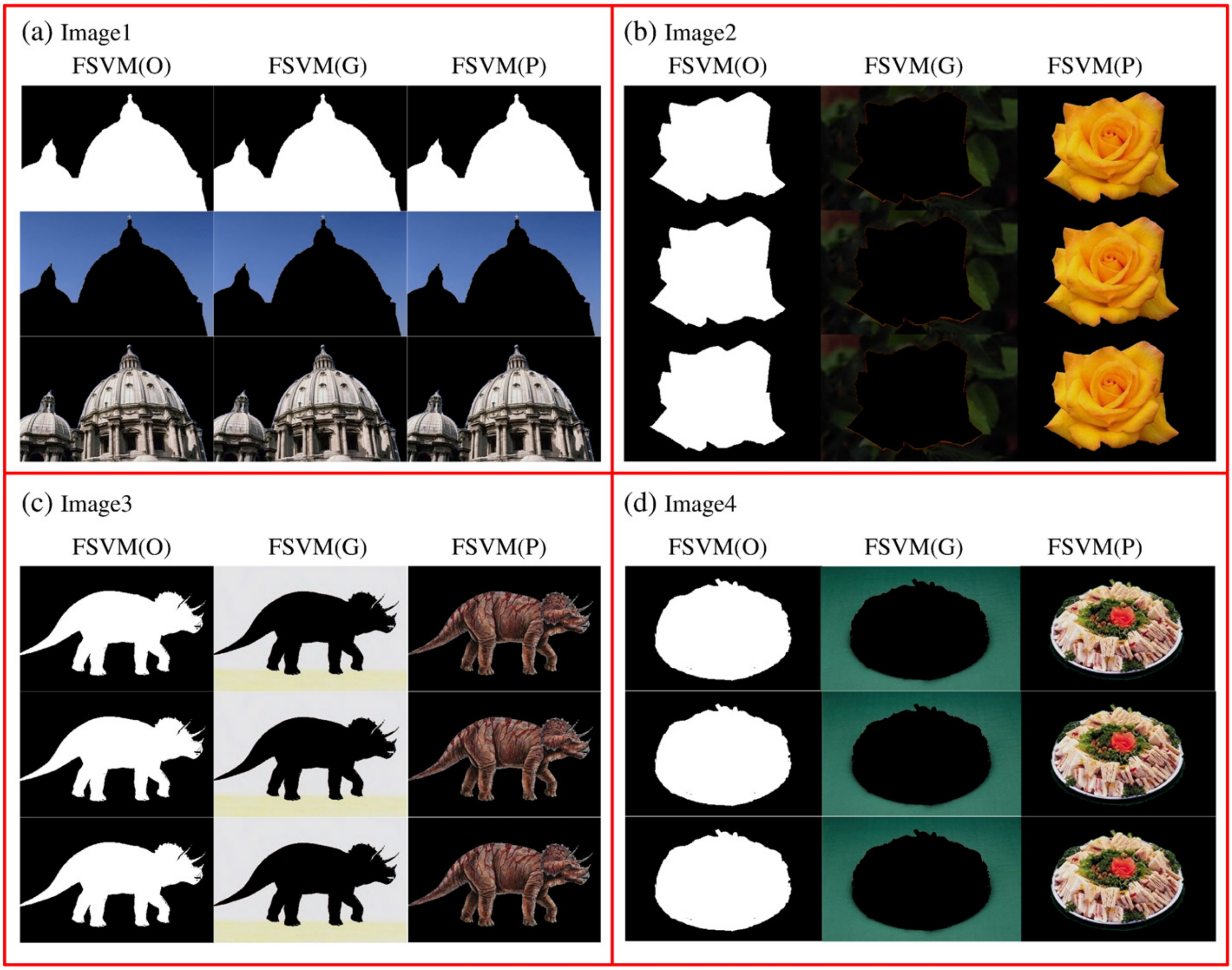

Fig. 10. The segmentation results of the images by the three approaches.

the reduction of the size of the training set, the computational cost of FSVM(G) is lower than that of FSVM(O).

FSVM(P) outperforms FSVM(O) and FSVM(G) and achieves the best performance as shown in Fig. 8(a2)–(d2). The training set of FSVM(P) has the smallest number of training vectors when compared with those of FSVM(G) and FSVM(O) as shown in Fig. 9(a1)–(d1). The projection process further reduces the redundant training vectors from the training sets after these sets are processed by the Gaussian model. The number of

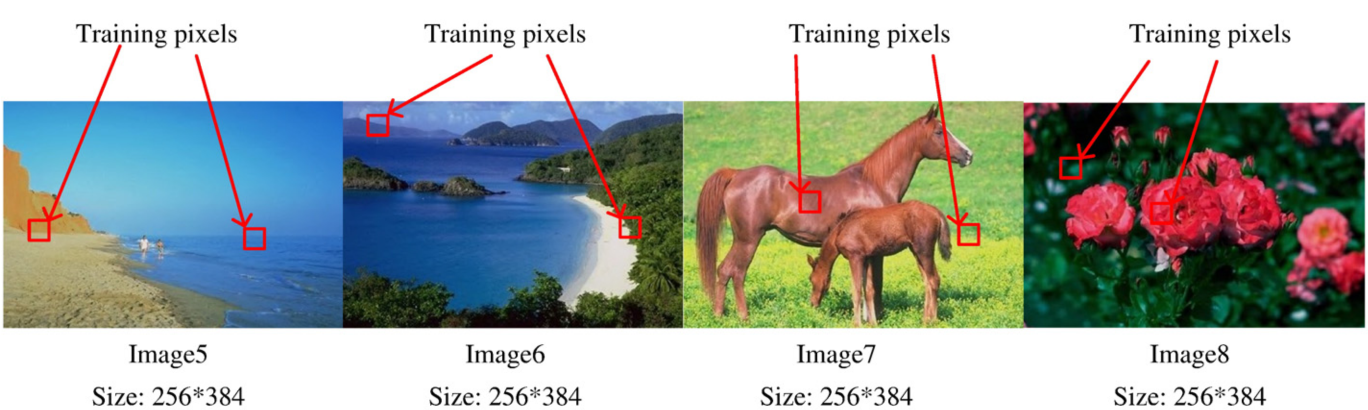

Fig. 11. The training pixels.

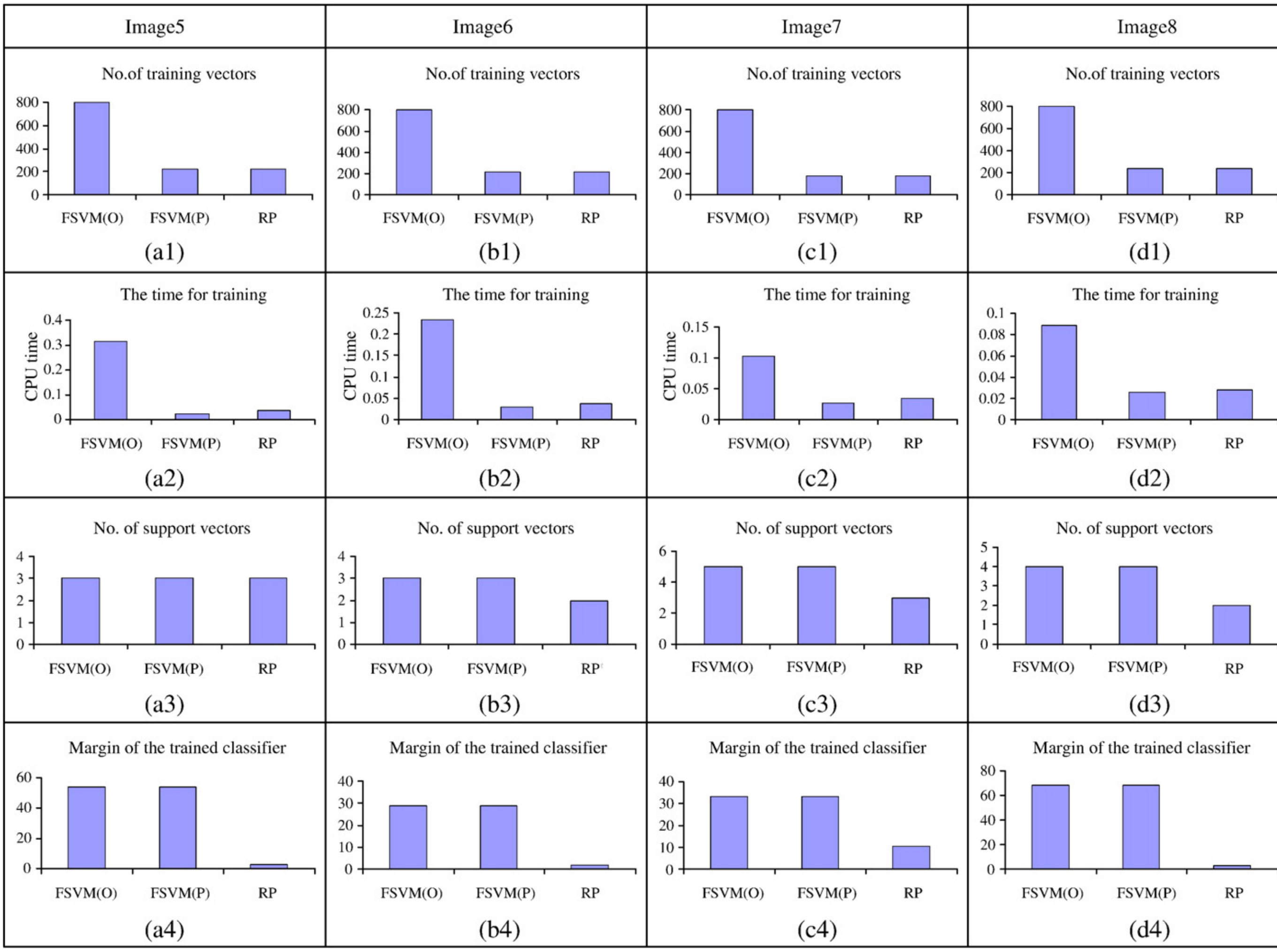

Fig. 12. The image segmentation performance of the different algorithms.

redundant training vectors which are eliminated by the projection process is 346, 308, 328 and 306 for Image1, Image2, Image3 and Image4, respectively. As a result, the sizes of the reduced training set are 109, 118, 140 and 190 for Image1, Image2, Image3 and Image4, respectively.

Fig. 9(a3)–(d3) show the margin of the trained classifier. The trained classifiers obtained by FSVM(O), FSVM(G) and FSVM(P) have the same margins with respect to the same images. The set of support vectors of the classifiers corresponding to the same images is also the same, and the number of support vectors is 9, 6, 2 and 3 for Image1, Image2, Image3 and Image4, respectively. This indicates that the eliminated training vectors by the Gaussian model and the projection process are the redundant training vectors, while all the support vectors are preserved.

Fig. 10 shows the segmentation results on the test data set. Each column of Fig. 10(a)–(d) presents the segmentation results obtained by the same classifier. The first row denotes the results represented in the form of black and white regions. The second and third rows denote each of the regions with its original color pixel values. The segmentation results obtained by the three classifiers corresponding to FSVM(O), FSVM(G) and FSVM(P) are indistinguishable, while FSVM (P) requires the lowest computational cost by comparing with the other two approaches, as shown in Fig. 8(a1)–(d1).

4.2. Comparison with random pruning

In the following, we compare our new approach FSVM(P), which performs SMO on the training data set after pruning by the projection process and the Gaussian model, with a random pruning approach (RP). The random pruning approach randomly eliminates a training vector from the training set until the number of training vectors is equal to a pre-specified value.

Fig. 11 illustrates four selected images. The pixels in the red rectangle in these images are the training vectors for each image, while the remaining pixels are considered as the test data set.

The segmentation results obtained by the approach FSVM(O) are regarded as the ground truth. In order to perform a fair comparison, the same number of remaining training vectors as in FSVM(P) is used in RP as shown in Fig. 12(a1)–(d1). Although the training times of RP for the four images are comparable to those of FSVM(P) as shown in Fig. 12(a2)–(d2), the trained classifiers by RP and FSVM(P) for the same image are significantly different: First, the support vectors are different. FSVM(P) adopts the same set of support vectors as those of FSVM(O), while RP eliminates most of these support vectors as shown in Fig. 12(a3)–(d3). Second, the margins of the trained classifiers by FSVM(O) and RP are also different. The margin of the trained classifier by FSVM(P) is the same as that of FSVM(O), while the margin of the trained classifier by RP is much smaller than that of FSVM(O) as shown in Fig. 12(a4)–(d4). Since the training vectors are eliminated randomly from the training set by the random pruning approach (RP), there is a high probability that the support vectors will be removed from the training set, which leads to the case that the margin of the classifier cannot be maximized in the training process.

Fig. 13 shows the segmentation results based on the classifiers obtained by FSVM(O), FSVM(P) and RP. Each column of Fig. 13(a)–(d) presents the segmentation results obtained by the same classifier. The

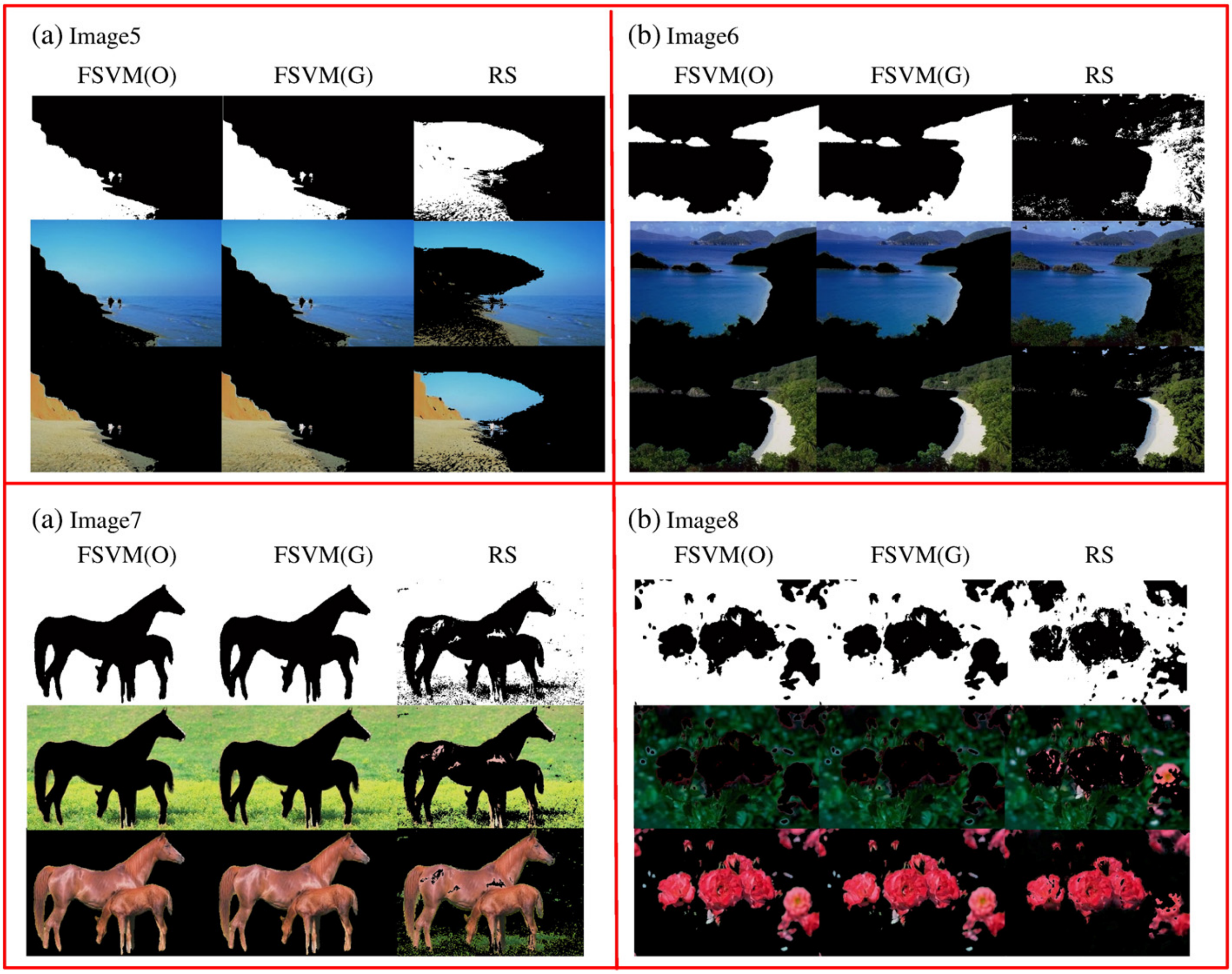

Fig. 13. The segmentation results of the images by the three approaches.

first row denotes the results represented in the form of black and white regions. The second and third rows denote each of the regions with its original color pixel values. The segmentation results obtained by FSVM(O) are regarded as the ground truth. It is obvious that the results based on FSVM(P) and FSVM(O) are indistinguishable, while the segmentation results obtained by RP are noisier when compared with the results of FSVM(O). To quantify the difference of the segmentation results, we use the following accuracy (AC) measure to characterize the discrepancy between the ground-truth segmentation results and the corresponding results obtained by FSVM(P) and RP:

$$AC = \frac{L^{correct}}{L^{total}} \times 100\% \tag{22}$$

where $L^{correct}$ denotes the number of pixels in the test set which are correctly labeled, and L^{total} denotes the total number of pixels in the test set. Table 1 shows the comparison of the accuracies obtained by FSVM(P) and RP. FSVM(P) achieves a higher accuracy due to its ability to preserve the support vectors and obtain the same classifier as that of FSVM(O). Since RP randomly prunes the training vector from the training set, many of the support vectors are removed, which leads to the low accuracy of RP.

Table 1
The accuracy of FSVM(P) and RP.

	Image5	Image6	Image7	Image8
FSVM(P)	100%	100%	100%	100%
RP	69.5%	85.4%	93.7%	97.5%

4.3. Further exploration

We perform further studies on the important cases of multiple classes and non-separable classes through the image segmentation application. The images in Fig. 14 not only contain multiple regions, but also include regions that are difficult to be distinguished. For example, it is difficult to distinguish the sky region and the building region in the left image of Fig. 14, and the sky region and the elephant region in the right image of Fig. 14, since these regions have similar colors. The pixels in the red rectangles in these images are the training vectors for each image respectively, while the remaining pixels are used as the test data set.

Table 2 shows the total computation time of FSVM(O), FSVM(G) and FSVM(P), while Table 3 lists the number of training vectors in FSVM(O), FSVM(G) and FSVM(P). FSVM(P) outperforms its competitors in terms of the total computation time, since it uses a smaller set of training vectors as shown in Table 3. This also indicates that the pruning strategy of FSVM(P) works well in the process of eliminating redundant training vectors. Fig. 15 illustrates the segmentation results based on the classifiers

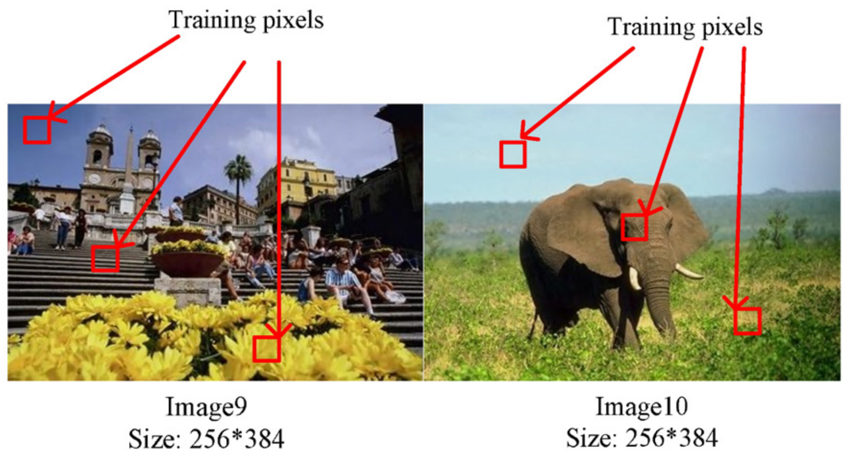

Fig. 14. The example of training pixels in the images.

obtained using FSVM(O), FSVM(G) and FSVM(P). All the segmentation results are satisfactory. Specifically, FSVM(O), FSVM(G) and FSVM(P) successfully subdivide the left image of Fig. 15 into three regions (the sky, the flower and the building), and they also correctly identify the elephant region from the sky region and the grass region. As a result, FSVM(O), FSVM(G) and FSVM(P) perform well in the important cases of multiple classes and non-separable classes. In particular, FSVM(P) is the best choice among all these approaches if we take into consideration both the efficiency and the segmentation quality.

4.4. Discussion

In summary, our proposed approach is based on the theory of SVM, the properties of support vectors and a probabilistic formulation. As is well known, the objective of SVM is to find a hyperplane which provides the maximum margin for both classes. The margin between both classes is measured by the distances between support vectors of each class and the hyperplane. If a support vector s is not located close to the boundary of its associated class, we can find another training vector s' along the direction toward the hyperplane, and this vector s' is closer to the hyperplane than the support vector s, which means that s is not the support vector. As a result, one of the interesting properties of support vectors is that they are located at the boundary of their classes. According to this property, we propose to adopt the pruning strategy based on the Gaussian model to eliminate the training vectors which are close to the center of their associated classes. In addition, another interesting property of support vectors is that they are located close to the hyperplane. As a result, the training vectors which are far away from the hyperplane can be eliminated by the pruning strategy based on the projection process.

In summary, the contributions of this work are as follows: First, the modified SVM (FSVM(G)) with the pruning strategy based on the Gaussian model is proposed to eliminate the redundant training vectors. The advantage of this is observed in Figs. 7(c2) and (d2) and 11(a1)–(d1), in which the number of training vectors is reduced when compared with traditional SVM (FSVM(O)). The reduced number of training vectors will in turn lead to the lower computational cost of FSVM(G). As shown in Figs. 8(a) and (b) and 10(a2)–(d2), the computation time of FSVM(G) is significantly smaller than that of FSVM(O).

Second, the modified SVM(FSVM(P)) with the pruning strategy based on the projection process is designed to improve the efficiency of FSVM(G). The effectiveness of this can be seen in Figs. 7(c3) and (d3) and 11(a1)–(d1), in which we observe that, through the incorporation of the pruning strategy based on the projection process, both the computation time and the number of training vectors are the lowest when compared with other approaches.

Third, the modified SVM (FSVM(P)) improves the efficiency of conventional SVM (FSVM(O)), while preventing any deterioration in the classification performance. The advantage of this is observed in Fig. 12 and 17, in which we observe that the segmentation results of the images obtained by FSVM(P) and FSVM(O) are indistinguishable, while FSVM(P) requires a lower computational cost when compared with FSVM(O), as shown in Fig. 10(a1)–(d1) and Table 2. As a result, FSVM(P) is the better choice when compared with traditional SVM if we consider the efficiency and the segmentation quality together.

5. Conclusion and future work

This paper investigates the problem of eliminating the redundant data from the training set in classifier learning. Although there exist a large number of techniques to improve the original SVM formulation, few of them consider how to reduce the number of training vectors while avoiding deterioration of the classification performance. Our major contribution is a new approach based on the Gaussian model and a projection process to eliminate the redundant training vectors. Based on these strategies, we further propose an algorithm to estimate the number of training vectors (the k value) at the boundary of the classes. Our experiments on the image data set demonstrate that the new approach can significantly reduce the computational cost without any change in the classification performance. In the future, we shall explore how to extend our approach to solve more difficult pattern classification problems in different application areas.

Acknowledgment

The work described in this paper was partially supported by grants from National Natural Science Foundation of China (Project No. 61003174 and 60973083), a grant from Natural Science Foundation of Guangdong Province, China (Project No. 10451064101004233), a grant from the

Table 2
The total computation time of different approaches (seconds).

	FSVM(O)	FSVM(G)	FSVM(P)
Image9	1.634	1.265	0.847
Image10	1.659	1.292	0.861

Table 3
The number of training vectors for the different approaches.

	FSVM(O)	FSVM(G)	FSVM(P)
Image9	1200	814	302
Image10	1200	822	313

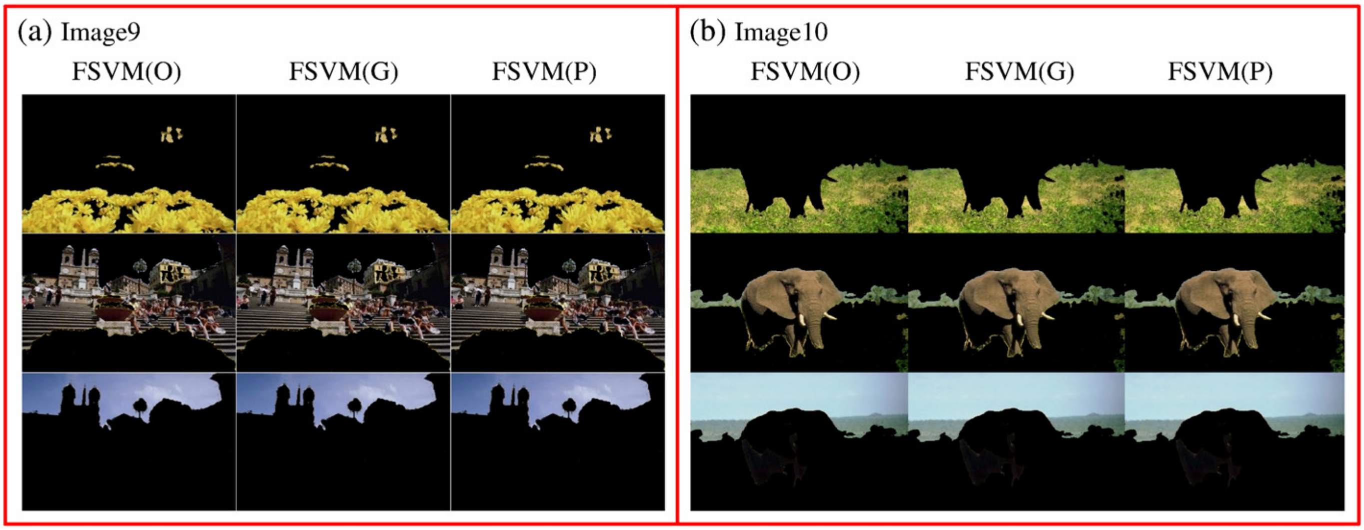

Fig. 15. The segmentation results of the images by the three approaches.

Fundamental Research Funds for the Central Universities (Project No. 2009ZM0255), a grant from the Research Grants Council of Hong Kong Special Administrative Region, China (Project No. CityU 121607) and a grant from the City University of Hong Kong (Project No. 7002374).

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd editionJohn Wiley & Sons, 2001.
[2] N. Cristianini, J. Shawe-Taylor, Support Vector Machines, Cambridge University Press, 2000.
[3] M.I. Schlesinger, V. Hlaváč, Ten lectures on statistical and structural pattern recognition, Kluwer Academic Publishers, 2002.
[4] C. Xiang, X.A. Fan, T.H. Lee, Face recognition using recursive Fisher linear discriminant, IEEE Trans. Image Process. 15 (8) (Aug. 2006) 2097–2105.
[5] J.C. Dunn, Well separated clusters and optimal fuzzy partitions, J. Cybernetics 4 (1974) 95–104.
[6] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Analysis. Machine Intell. 1 (4) (1979) 224–227.
[7] L. Konopnicki, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, New York, 1990.
[8] H. Akaike, Prediction and entropy, in: A.C. Atkinson, S.E. Fienberg (Eds.), Celebration of Statistics, Sprieger-Verlag, New York, 1985, pp. 1–24.
[9] J. Rissanen, Modeling by shortest data description, Automatica 14 (1978) 465–471.
[10] G. Schwarz, Estimating the dimension of a model, Ann. Stat. 6 (1976) 461–464.
[11] Arthur Dempster, Nan Laird, Donald Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B 39 (1) (1977) 1–38.
[12] T. Kohonen, Self-Organizing Maps, 2nd edition, Springer-Verlag, Heidelberg, 1997.
[13] Q.M. Luo, T.M. Khoshgoftaar, Unsupervised multiscale color image segmentation based on MDL principle, IEEE Trans. Image Process. 15 (9) (2006) 2755–2761.
[14] S.C. Zhu, A.L. Yuille, Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 18 (9) (Sep. 1996) 884–900.
[15] J. Li, J.Z. Wang, Automatic linguistic indexing of pictures by a statistical modeling approach, IEEE Trans. Pattern Anal. Mach. Intell. 25 (9) (2003) 1075–1088.
[16] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using expectation-maximization and its application to image querying, IEEE Trans. Pattern Anal. Mach. Intell. 24 (8) (2002) 1026–1038.
[17] J. Lie, M. Lysaker, X.C. Tai, A binary level set model and some applications to Mumford-Shah image segmentation, IEEE Trans. Image Process. 15 (5) (May 2006) 1171–1181.
[18] S. Gao, T.D. Bui, Image segmentation and selective smoothing by using Mumford-Shah model, IEEE Trans. Image Process. 14 (10) (Oct. 2005) 1537–1549.
[19] T. Brox, J. Weickert, Level set segmentation with multiple regions, IEEE Trans. Image Process. 15 (10) (Oct. 2006) 3213–3218.
[20] M.H. Gazit, R. Kimmel, N. Peled, D. Goldsher, Segmentation of thin structures in volumetric medical images, IEEE Trans. Image Process. 15 (2) (Feb. 2006) 354–363.
[21] D.P. Mukherjee, N. Ray, S.T. Acton, Level set analysis for leukocyte detection and tracking, IEEE Trans. Image Process. 13 (4) (April 2004) 562–572.
[22] V.N. Vapnik, Statistical Learning Theory, John Wiley & Sons, 1998.
[23] Corinna Cortes, Vladimir Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[24] John C. Platt, Using analytic QP and sparseness to speed training of support vector machines, NIPS (1998) 557–563.
[25] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, John C. Platt, Support vector method for novelty detection, NIPS (1999) 582–588.
[26] C.C. Chang, C.J. Lin, Training ν-support vector classifiers: theory and algorithms, Neural Comput. 13 (9) (2001) 2119–2147.
[27] Bernhard Schölkopf, Patrice Simard, Alex J. Smola, Vladimir Vapnik, Prior knowledge in support vector kernels, NIPS (1997) 640–646.
[28] Thorsten Joachims, Estimating the generalization performance of an SVM efficiently, ICML (2000) 431–438.
[29] Thorsten Joachims, A support vector method for multivariate performance measures, ICML (2005) 377–384.
[30] Harris Drucker, Christopher J.C. Burges, Linda Kaufman, Alex J. Smola, Vladimir Vapnik, Support vector regression machines, NIPS (1996) 155–161.
[31] Michael E. Mavroforakis, Sergios Theodoridis, A geometric approach to support vector machine (SVM) classification, IEEE Trans. Neural Netw. 17 (3) (2006) 671–682.
[32] Kristin P. Bennett, Erin J. Bredensteiner, Duality and geometry in SVM classifiers, ICML (2000) 57–64.
[33] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Trans. Neural Netw. 11 (1) (2000) 124–136.
[34] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, Tech. Rep. TR-ISL-99-03, Dept. CSA, IISc, Bangalore, Karnataka, India, 1999.
[35] V. Franc, V. Hlaváč, An iterative algorithm learning the maximal margin classifier, Pattern Recognit. 36 (9) (2003) 1985–1996.
[36] T.T. Friess, R. Harisson, Support vector neural networks: the kernel adatron with bias and soft margin, Univ. Sheffield, Dept. ACSE, 1998 Tech. Rep. ACSE-TR-752.
[37] Michael E. Mavroforakis, Margaritis Sdralis, Sergios Theodoridis, A novel SVM geometric algorithm based on reduced convex hulls, ICPR (2) (2006) 564–568.
[38] J.X. Dong, A. Krzyzak, C.Y. Suen, Fast SVM training algorithm with decomposition on very large data sets, IEEE Trans. Pattern Anal. Mach. Intell. 27 (4) (2005) 603–618.
[39] Sergios Theodoridis and Konstantinos koutroumbas, Pattern Recognition (Third edition), Academic Press, Elsevier, 2006.