Neurocomputing 136 (2014) 243–255

Contents lists available at ScienceDirect

Neurocomputing

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

Saliency-SVM: An automatic approach for image segmentation

Xuefei Bai [a], Wenjian Wang [a,b,*]

[a] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, PR China
[b] Key Laboratory of Computational Intelligence & Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, PR China

ARTICLE INFO

Article history:
Received 5 June 2013
Received in revised form
2 January 2014
Accepted 4 January 2014
Communicated by G.-B. Huang
Available online 23 January 2014

Keywords:
Image segmentation
Visual saliency detection
Support vector machine
Training set selection

ABSTRACT

Although there are some support vector machine (SVM) based methods for image segmentation, automatically and accurately segmenting objects that appeal to human perception is indeed a significant issue. One problem with these methods may be that the human visual attention is seldom taken into consideration. This paper proposes a novel visual saliency based SVM approach for automatic training data selection and object segmentation, namely Saliency-SVM. Firstly, a trimap of the given image is generated according to the saliency map in order to estimate the prominent object locations. Then, positive (salient object) and negative (background) training sets are automatically selected through histogram analysis on trimap for SVM training. Finally, the whole salient object is segmented using the trained SVM classifier. Experiment results on a benchmark dataset demonstrate the effectiveness of the proposed approach.

1. Introduction

Image segmentation is one of the most challenging problems in computer vision and image processing as it serves as a key and fundamental step to higher-level tasks such as image retrieval, object recognition and image understanding [1]. The goal of image segmentation is to partition an image into some no-overlapped meaningful regions, which have more similarity in the regions and less similarity between the regions. In recent years, there have been tremendous researches for image segmentation [1,2], such as threshold methods [3,4], region-based methods [5,6], edge detection methods [7,8], level set and active contour models [9,10], graph-based method [11], clustering methods [12,13], superpixel based method [14], and other hybrid techniques.

Threshold segmentation methods are widely used because of their simplicity and efficiency. However, traditional histogram-based threshold algorithms can only separate those areas which have distinct different gray levels. In addition, they cannot work well for images whose histograms are nearly unimodal. For edge-based methods, the most commonly used edge detection operators include Candy, Sobel, Prewitt and Laplacian. These operators are suitable for images which are simple and noise-free, and they focus on detecting pixels with abrupt grayscale changes on the object edges, therefore hard to yield closed contours or homogeneous regions. While region growing, splitting, merging and other region-based segmentation algorithms often deal with spatial repartition of image feature information to generate closed and homogeneous regions. But over-segmentation and under-segmentation are critical issues to be considered in such methods. And the task of level set-based image segmentation and active contour models are always formulated in energy minimization frameworks by selecting a set of suitable criteria, which are encoded as region or boundary functional in a cost function. If the coarse initial contours of the objects are provided appropriately by the user beforehand, this kind of methods can obtain promising segmentation results. Graph based image segmentation methods are modeled to divide a graph into several sub-graphs such that each of them represents a meaningful object in the image, but they always suffer from high computational complexity. And clustering methods, viewing an image as a large number of multidimensional data and classifying the image into different parts according to certain homogeneity criterion, can get much better segmentation results. But over-segmentation is the problem that must be solved and feature extraction is also an important factor for clustering. Superpixel or image segments can provide helpful grouping cues to guide segmentation and reduce the computational complexity. But sometime the segmentation performance depends on the superpixel generation approach. In conclusion, though much emphasis has been put on image segmentation and many approaches have been proposed in recent decades, there is no universal segmentation approach effective for all kinds of images.

As mentioned above, image segmentation can be viewed as a classification problem, which means labeling each pixel according to certain essential characteristics. Therefore lately, some popular

[*] Corresponding author at: School of Computer and Information Technology, Shanxi University, Taiyuan 030006, PR China. Tel.: +86 0351 7017566; fax: +86 0351 7017566.
E-mail addresses: baixuefei@sxu.edu.cn (X. Bai), wjwang@sxu.edu.cn (W. Wang).

0925-2312/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2014.01.008

classification methods have already been utilized successfully for image segmentation. Among them, support vector machine (SVM) proposed by Vapnik [15], is an excellent learning and classification method with the characteristics of high accuracy, fast computational speed, robustness and strong generalization ability. Moreover, SVM exhibits many unique advantages in solving classification problems with small size samples, nonlinear and high dimension, making it more suitable for image segmentation.

Considering segmenting a given image into foreground and background, this problem can be treated as a typical binary classification problem, therefore these advantages of SVM mentioned previously can be taken to solve this problem. Therefore, SVM classifier was used for some special image segmentation problems. Yu and Chang [16] presented an effective and efficient method for solving scenery image segmentation by applying SVM classifier. Mitra et al. [17] proposed a supervised pixel classifier for remote sensing image segmentation, and an active support vector learning algorithm was adopted to decrease the number of labeled points required to design the classifier. Cyganek [18] proposed an efficient color segmentation method which was based on the one-class SVM classifier, and the method has been developed especially for the road signs recognition system. Recently, some SVM based methods were also proposed for color image segmentation. Yu et al. [19] introduced a new SVM-based approach named Fast Support Vector Machine (FSVM), in which positive and negative training pixels were firstly marked by users as small rectangles in objects and background. Then a pruning strategy based on Gaussian model and a projection process were used to preserve support vectors while eliminating redundant training vectors. Finally, the remaining pixels viewed as the test set, was segmented into several regions by the trained SVM classifier. Experiment on some test images demonstrated that FSVM can significantly reduce the computational cost without losing classification performance. But manual intervention is required in FSVM because training samples are pre-specified by users. Furthermore, different training samples will affect the final segmentation performance of FSVM. More recently, a new approach for color image segmentation using SVM and Fuzzy c-means (FCM) was proposed [20], in which training samples of SVM were randomly selected from FCM clustering results. However, the number of clusters of FCM must be set in advance, and the random selection of training samples will also affect the final segmentation performance. An unsupervised method based on saliency maps and Fuzzy SVM was presented [21], in which saliency maps and corner points were used to produce a rectangle where the object locates. Then, Fuzzy SVM was used to segment single object in the rectangle. However, in the training process, the positive training samples in the rectangle must fall within the object. Otherwise they will affect the training results.

Though much emphasis has been put on SVM-based image segmentation and many approaches have been proposed, it is still a challenging task to automatically segment natural images due to their inherent complexity. And there are some unresolved issues in existing methods: (1) Similar to other SVM classification tasks, how to choose kernel function and its parameters still remains unresolved theoretically in SVM-based image segmentation and (2) SVM adopts a supervised learning mechanism, which makes use of some labeled training samples to learn the classifier, while these labeled training samples are not always available in practical tasks, especially for various image segmentation problems. And as far as we know, these two overlooked issues are rarely mentioned in the literature about SVM-based image segmentation. Restricted by the paper length, the first issue is beyond the scope of this paper. Therefore, how to effectively exploit multiple characteristics of image itself to automatically generate SVM training sets, as well as explore the excellent classification performance of SVM classifier for automatic color image segmentation is the main focus of this paper.

Recently, visual saliency detection, being closely related to how we perceive and process visual stimuli, is investigated by multiple disciplines including cognitive psychology, neurobiology, computer vision and image processing, and it has been successfully applied in combination with other methods for image segmentation [22–25]. In [22], a saliency map produced by spectral residual approach [26] was used to provide seeds for graph cuts implementation. And in [23], salient regions extracted by the saliency map and threshold method were viewed as the initialized regions for GrabCut segmentation method [27]. Similarly, the visual attention saliency map generated by three (color, intensity, and orientation) feature maps was used to guide region merging using a simple modified particle swarm optimization [24]. Later, a saliency-directed color image segmentation approach was presented in [25], in which a special facial saliency map was used to guide region merging method for the tracking based face segmentation. It follows that saliency maps generated using selective attention models can provide some useful cues for image segmentation.

Therefore to address the issues mentioned above, this paper proposes a novel and efficient approach integrating visual saliency detection and SVM classifier, namely Saliency-SVM, for automatic and adaptive color image segmentation. Unlike other SVM training dataset selection methods, salient region and background pre-segmented based on a saliency map extracted by visual saliency are explored to identify the positive and negative training datasets of SVM. And then, training pixels are automatically selected by a local homogeneity criterion for SVM training. Finally, the salient object is segmented from background using the trained SVM model.

The rest of this paper is organized as follows. In Section 2, we briefly introduce some related work about saliency detection method and SVM. The detailed process of the proposed Saliency-SVM is described step by step in Section 3. In Section 4, experiment results are analyzed and discussed. Finally, conclusions are addressed in last section.

2. Background knowledge about visual saliency detection and SVM

For the sake of readability of the following sections, we first briefly introduce some background knowledge about visual saliency detection mechanism and standard SVM.

2.1. Visual saliency detection mechanism

Human visual system has selective attention mechanism that directs human vision to the most interested parts of the received visual scene, which is often referred to as salient region, region of interest, or attention region. Using visual selective attention in a computer vision or image processing system can diminish the received visual data to some compact and relevant information [28]. In this way, saliency maps produced by visual selective attention are always used to find approximate object locations that are relatively meaningful and consistent with human perception. And the salient value of each pixel in a saliency map corresponds to how much attention may be focused on it. In other words, saliency describes what is prominent or noticeable. To find the attention region in a given image, many computational approaches have been proposed to model visual saliency maps in research fields of psychology, neurobiology, image processing and computer vision.

Based on a biologically plausible model proposed by Koch and Ullman [28], Itti and Koch proposed the most influential saliency model derived from “feature integration” theory to simulate the visual search process of human for rapid scene analysis [29]. Their system separately constructed intensity, color and orientation feature maps by a set of linear center-surround contrast using the difference between multiple scales. Saliency map was obtained as a combination of three feature maps, and determined a gaze point at the largest feature value. Later, Harel et al. combined activation maps derived from graph theory and other maps obtained by Itti's model to form a new graph-based saliency map [30].

Ma and Zhang proposed local contrast analysis to estimate saliency using a fuzzy growth model [31]. In addition, Liu et al. employed a set of features including multiscale contrast, center-surround histogram and color spatial distribution to describe a salient object, and a Conditional Random Field (CRF) was learned by combining these features to detect salient object [32]. Goferman et al. proposed a context-aware saliency to detect the image regions, which depended on the single scale and multiscale saliency detection [33]. Lately, Cheng et al. proposed a regional contrast based saliency extraction algorithm [34], which simultaneously evaluated global contrast differences and spatial coherence.

In addition to the contrast based methods mentioned above, saliency map can be computed by image frequency domain analysis. By analyzing the log-spectrum of natural images, Hou and Zhang generated the saliency map based on the spectral residual of the amplitude spectrum of an image's Fourier transform [26]. However in [35,36], the authors proposed and proved that it is the phase spectrum instead of amplitude spectrum of Fourier transform is the key to calculate the locations of salient regions. More recently, Achanta et al. applied a frequency tuned method to compute center-surround contrast using color differences from an image, in which saliency values were averaged within image segments produced by MeanShift pre-segmentation [37]. Then, the authors extended their work in [38] by varying the bandwidth of the center-surround filtering near image borders using symmetric surrounds. Generally, compared with methods based on image feature contrast, methods based on frequency domain analysis can be easily implemented since they have lower computational complexity and fewer parameters.

2.2. Support vector machine

Support vector machine (SVM) is a state-of-the-art machine learning technique whose foundations stem from statistical learning theory [15]. It adopts structural risk minimization principle which overcomes the conflict between over-fitting and under-fitting. And it has strong generalization to reduce the influence of the noises in training set under good accuracy. In addition, it overcomes the problems of dimension disaster through non-linear transform and dot matrix kernel function which is not to add computational complexity when mapping to higher dimension space.

Given a training dataset of l points $\{x_i, y_i\}_{i=1}^{l}$ with the input data $x_i \in \mathcal{R}_n$ and the corresponding target $y_i \in \{-1, +1\}$. In feature space, SVM takes the form:

$$\mathbf{Y}(\mathbf{X}) = \omega^{\mathbf{T}} \varphi(\mathbf{X}) + \mathbf{b} \tag{1}$$

where the non-linear mapping $\varphi(\cdot)$ maps the input vector into a so-called higher dimensional feature space, **b** is the bias and ω is a weight vector of the same dimension as the feature space.

SVM formulations start from the assumption that the linear separate case is

$$\begin{cases} \omega^T x_i + \mathbf{b} \geq +1 & \text{if } y_i = +1 \\ \omega^T x_i + \mathbf{b} \leq -1 & \text{if } y_i = -1 \end{cases} \tag{2}$$

For the non-separable case

$$\begin{cases} \omega^T \boldsymbol{\varphi}(x_i) + \mathbf{b} \geq +1 & \text{if } y_i = +1 \\ \omega^T \boldsymbol{\varphi}(x_i) + \mathbf{b} \leq -1 & \text{if } y_i = -1 \end{cases} \tag{3}$$

In this space, a linear decision surface is constructed with special properties that ensure high generalization ability of the network. By using a non-linear kernel function, it is possible to compute a separating hyper-plane with a maximum margin in a feature space.

We need to find an existing maximum margin $2/\|\omega\|$ between the classes among all hyper-planes separating the data. So, the classification problem is transformed into a quadratic programming problem:

$$\begin{aligned} &\min \quad \frac{1}{2}\omega^T \omega + \mathcal{C} \sum_{i=1}^{l} \zeta_i \\ &\mathbf{s.t} \quad y_i(\omega^T \boldsymbol{\varphi}(x_i) + \mathbf{b}) = 1 - \zeta_i, \quad \zeta_i \geq 0, \; i = 1, \ldots, l \end{aligned} \tag{4}$$

where $\mathcal{C}$ is the trade-off parameter between the error and margin.

The quadratic programming problem can be solved by using Lagrangian multipliers $\alpha_i \in \Re$. The solution satisfies the Karush–Kuhm–Tucker (KKT) conditions. And ω can be recovered by using $\omega = \sum_{i=1}^{l} \alpha_i y_i \boldsymbol{\varphi}(x_i)$, where α_i are non-zero values and x_i are support vectors (SV).

The decision boundary is determined only by the support vectors. Let $t_j (j = 1, \ldots, s)$ be the indices of the s support vectors. Then we can rewrite

$$\omega = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \boldsymbol{\varphi}(x_{t_j}) \tag{5}$$

The quadratic programming problem is solved by considering the dual problem

$$\begin{aligned} &\max_{\alpha} \quad Q(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ &\mathbf{s.t} \quad \begin{cases} 0 \leq \alpha_i \leq \mathcal{C} \\ \sum_{i=1}^{l} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{6}$$

With the kernel trick (Mercer Theorem)

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \tag{7}$$

Several types of kernels, such as linear, polynomial, splines, RBF, and MLP, can be used within the SVM. This finally results in the following:

$$\mathbf{y}(x) = \text{sign}\left(\sum \alpha_i y_i K(x, x_i) + b\right) \tag{8}$$

In addition to linear classification, SVM can be applied to nonlinear classification tasks, in which nonlinear mapping is used to generate the features from the original data space. The non-linearly separable data to be classified is mapped into a high-dimensional feature space, where the data can be linearly classified.

Although studies of visual attention have demonstrated that saliency map is sufficient to offer some useful information about salient objects and background [21–24], but as far as we know, the application of saliency map to automatically guide training data selection for SVM-based image segmentation is seldom reported.

3. Proposed saliency-SVM for image segmentation

In this paper, we attempt to obtain a solution to realize automatic selection of SVM training data and automatic image segmentation. The proposed Saliency-SVM method starts with saliency detection to find the prominent locations of salient object. And then a more discriminative representation, i.e., a trimap consisting of salient object, background and the residual region, is exploited to improve the result of saliency detection. Thus, positive (object) and negative (background) training datasets of SVM are automatically selected by means of histogram analysis on the trimap. To improve the SVM training speed, a local homogeneous criterion is used to select training pixels. Finally, the salient object is segmented from background using the trained SVM model. The whole procedure of proposed Saliency-SVM is illustrated as Fig. 1, which will be described in detail in the following subsections.

3.1. Pre-segmentation and trimap generation

Motivated by the visual attention model, which has been successfully extended to some image segmentation approaches [22,23], spectral analysis method [26,36] is employed to construct a saliency map indicating the potential salient regions due to its low computational cost and unsupervised manner.

For a given color image, at first, transform it to a gray level image $I(x,y)$, and its saliency map $SM(x,y)$ is calculated as follows:

$$f(x,y) = F(I(x,y)) \tag{9}$$

$$p(x,y) = P(f(x,y)) \tag{10}$$

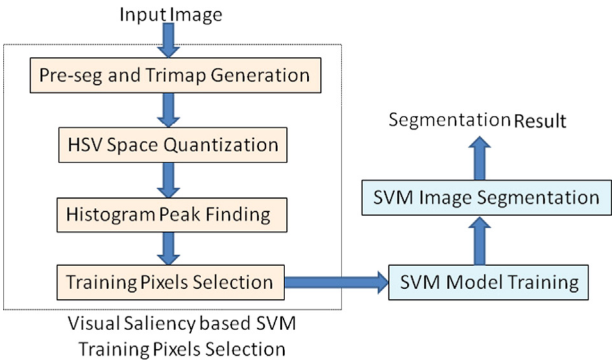

Fig. 1. Procedure of saliency-SVM.

$$SM(x,y) = g{*}\| F^{-1}[e^{i \cdot p(x,y)}] \|^2 \tag{11}$$

where F and F^{-1} refer to the Fourier Transform and Inverse Fourier Transform, respectively. $p(x,y)$ represents the phase spectrum of the Fourier transformed image, and $g{*}$ is a 2D Gaussian convolution with $\sigma = 8$ for a better visual effect as used in Refs. [26,36].

Saliency map generated in this way is a gray level image, which represents the salient value of each pixel. The saliency values range from 0 to 255. The larger the saliency value, the more likely the pixel attracts the observer's interest, as shown in Fig. 2(b). In order to further identify the location of a salient object, the object map $OM(x,y)$ is obtained by a binarization precessing for $SM(x,y)$:

$$OM(x,y) = \begin{cases} 0 & \text{if } SM(x,y) < t \\ 1 & \text{if } SM(x,y) \geq t \end{cases} \tag{12}$$

The binarization threshold t is set to be the value which maximizes the discrimination criterion (σ_B^2/σ_W^2) of two classes (the salient object and background), where σ_B^2 is the between-class variance and σ_W^2 is the within-class variance, respectively.

As illustrated in Fig. 2(c), the white area of $OM(x,y)$ represents the rough estimation of salient object R_o, while black area means the rough estimation of background R_b. And we have found that in general natural scene images, most of the salient object appear at or near the center of the image in order to attract user attention distinctly. So, the binary object map $OM(x,y)$ should be regularized using some morphological operators [39], to remove these uncertain pixels near the boundary of R_o. Here, the boundary of R_o is shrunk to form a more accurate salient object mask M_o, and then it is expanded to form the background mask:

$$\begin{aligned} M_o &= R_o \ominus E_{r_e} \\ M_b &= ((R_o \oplus D_{r_d}) - R_o) \cup R_b \end{aligned} \tag{13}$$

where $\ominus E_{r_e}$ is an erosion operator indicating shrinking region R_o for r_e pixels, and $\oplus D_{r_d}$ is a dilation operator denoting expanding region R_o for r_d pixels. A square structural element with the width of 10 pixels is used in erosion and dilation operators.

Therefore, salient object and background in the given image can be pre-segmented with the masks M_o and M_b, as shown in

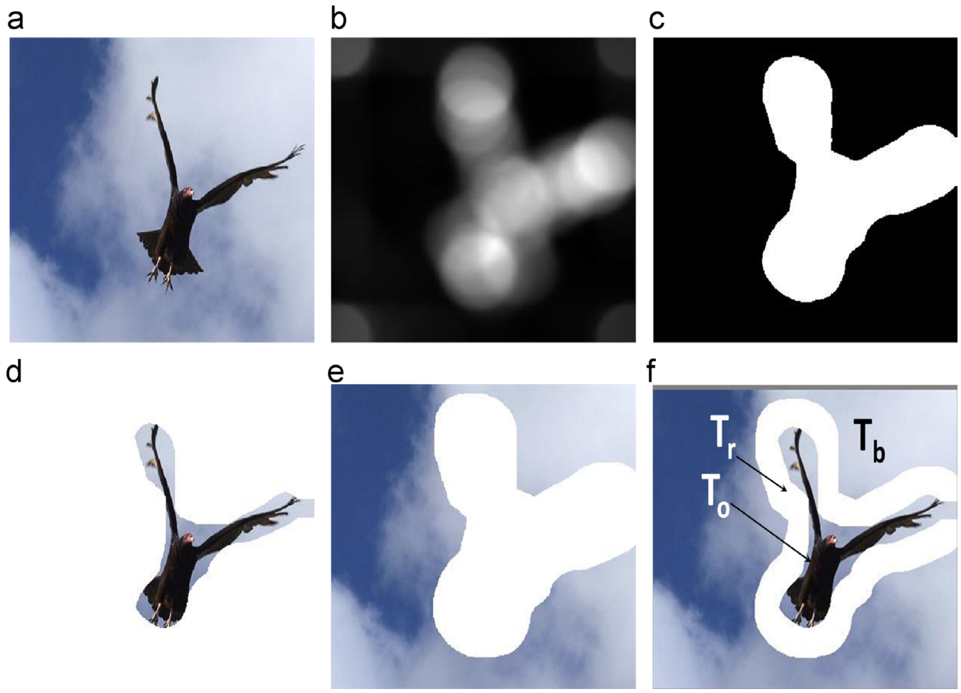

Fig. 2. (a) Original image; (b) saliency maps; (c) binary mask; (d) salient region; (e) background; (f) trimap.

Fig. 2(d) and (e). Finally, a coarse trimap $\{T_o, T_r, T_b\}$ is produced, in which T_o means salient object pre-segmented, T_b represents background, and T_r is the uncertain pixel set in residual region, as shown in Fig. 2(f).

As can be seen from Fig. 2, salient object and background pre-segmented in this stage are only approximate representations, but this pre-processing stage driven by visual saliency is simple and fast, and the results obtained can provide some essential cues for the subsequent SVM-based training and segmentation.

3.2. SVM training dataset generation

As described previously, we strive to achieve an automatic SVM image segmentation method purely based on the characteristics of image itself. And it can be seen that pixels in T_o and T_b regions have obvious distinguishing characteristics, such as color feature and spatial location. Therefore, in order to accurately segment out salient object from background, all pixels in T_o and T_b can be used to train the SVM classifier. However, all these pixels are redundant as training samples, and the total number of these pixels makes a large-scale training dataset for SVM training. Obviously, in salient object and background, there are always some pixels that can represent certain characteristics of homogeneous region where they locate, referred to as "representative pixels" in this paper. So to solve the problem mentioned above, in this stage, we aim at performing SVM training on a small-scale training dataset consisting with representative pixels in T_o and T_b regions to speed up the process.

The primary difference between salient object and background is often reflected in color feature and spatial location. In order to select a set of representative pixels from region T_o and T_b, we use pixel color and spatial features as the distinguishing property. So, pixels that satisfy the following two-fold criteria are selected as representative ones to constitute the SVM training datasets:

Spatial criterion: Pixel that locates in T_o or T_b region
Color criterion: Pixel that with dominant color of T_o or T_b

The first spatial criterion of representative pixel is easy to obtain, that is the two-dimensional horizontal and vertical coordinates. And the dominant colors of T_o and T_b in second criterion, which can represent the distinguishing color distribution characteristic of each region, are determined by the following scheme.

3.2.1. HSV color space quantization

There are many ways can be used to express dominant colors. However, the most commonly used RGB color space contains 256^3 possible colors, which is too computationally expensive in the feature extraction process even for small sized images. On the other hand, HSV (Hue: [0, 360°], Saturation: [0, 1], and Value: [0, 1]), which is capable of emphasizing human visual perception, is shown to have better results for image segmentation than RGB color space [40]. Thus, in order to determine the dominant color of region T_o and T_b, a quantization operation in HSV color space is firstly implemented to reduce the computational complexity. That is to say, each channel of HSV color space is quantized to different values, and then a one-dimensional histogram is generated.

Because the human visual system is more sensitive to hue than to saturation and intensity so that the hue channel should be quantized finer than saturation and intensity. And it is well known that the color distribution (red, orange, yellow, green, cyan, blue and purple) of the hue channel is not uniform, therefore a non-uniform quantization scheme similar to [41] is applied. As a result, the hue channel is quantized to 7 non-uniform bins represented from 0 to 6, and each indicates a major color, as shown in Fig. 3.

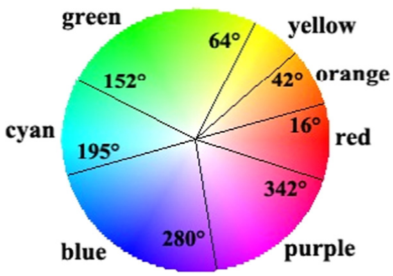

Fig. 3. Hue channel quantization. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

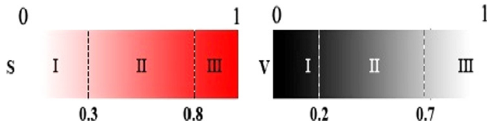

Fig. 4. SV channels quantization. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

And the saturation and intensity channels are quantized non-uniformly to three bins in the same way. Taking red color for the example, S channel from white to red with different saturation and V channel from black to white with different intensity are shown in Fig. 4. When S value is large enough, for instance S is greater than 0.8, region-III can be regarded as pure red color. And when V value is small enough, for instance V is smaller than 0.2, region-I can be perceived as a pure black area. Therefore, three non-uniform bins expressed from 0 to 2 are enough to represent the saturation and intensity information.

The quantization scheme can also be summarized as follows:

$$H=\begin{cases}0 & \text{if } h\in(342,16]\\ 1 & \text{if } h\in(16,42]\\ 2 & \text{if } h\in(42,64]\\ 3 & \text{if } h\in(64,152]\\ 4 & \text{if } h\in(152,195]\\ 5 & \text{if } h\in(195,280]\\ 6 & \text{if } h\in(280,342]\end{cases}\qquad \begin{matrix} S=\begin{cases}0 & \text{if } s\in[0,0.3)\\ 1 & \text{if } s\in[0.3,0.8)\\ 2 & \text{if } s\in[0.8,1]\end{cases}\\ V=\begin{cases}0 & \text{if } v\in[0,0.2)\\ 1 & \text{if } v\in[0.2,0.7)\\ 2 & \text{if } v\in[0.7,1]\end{cases}\end{matrix} \tag{14}$$

According to the above quantization scheme, one-dimensional feature vector is constructed by three channel values as follows:

$$L=Q_sQ_vH+Q_vS+V \tag{15}$$

where Q_s and Q_v are quantization coefficients of saturation and intensity channels, respectively. As most quantization methods used, quantization coefficients are set as $Q_s=Q_v=3$, hence,

$$L=9H+3S+V \tag{16}$$

Thus, three channels (hue, saturation and intensity) can be distributed in one-dimensional vector L and $L\in\{0,1,\ldots,62\}$. Because the quantization result has only 63 bins, the computational complexity will be decreased tremendously. Furthermore, by considering the non-uniform character in three channels, the quantization result is more similar to the human vision mode. So the color value of each pixel in T_o and T_b regions can be quantized to one of 63 colors, and then the HSV histogram with 63 bins of T_o and T_b will be calculated subsequently to determine their dominant colors by peak selection in next step, respectively.

3.2.2. Histogram peak selection

Peaks in histogram can indicate the distribution information of image colors. For a color image, dominant colors could be identified by peaks in its global histogram. While there are obvious color differences between salient object T_o and background T_b extracted previously, so dominant colors of these regions can be found by histogram peak selection. Hence, after the HSV color space quantization, an adaptive histogram peak selection method is subsequently presented to reduce the number of colors in quantized HSV color space and ignore less frequently occurring colors, so as to select the dominant colors of T_o and T_b regions.

Considering that the colors in a natural image typically cover only a small portion of the full color space, and statistical result on 1000 natural images shows that no more than three or four dominant colors exist in salient object or background in above 85% images. So, it is assumed that no more than three dominant colors are necessary to describe T_o and T_b regions. Taking salient regions T_o for the example, the main steps to adaptively select histogram peaks are briefly stated as follows:

Step 1: Calculate the global histogram of region T_o after HSV color space quantization:

$$H^o = \frac{Num(f(x,y)=l_i)}{Num(T_o)}, \quad (x,y) \in T_o, \; l_i \in \{0,1,\ldots,62\};$$

where $Num(T_o)$ means the total number of pixels in region T_o, and $Num(f(x,y)=l_i)$ is the number of pixels with color level l_i in T_o.

Step 2: Identify all peaks $P_{ko}: P_{l_1}, P_{l_2}, \ldots, P_{l_k}$, l_i is the color level index of the ith peak, and $l_1 < l_2 < \cdots < l_k$.

Step 3: Compute the max and min peak values of H^o. $P_{max} = \max\{P_{l_1}, P_{l_2}, \ldots, P_{l_k}\}$, $P_{min} = \min\{P_{l_1}, P_{l_2}, \ldots, P_{l_k}\}$, the mean value $\mu_m = (P_{max}+P_{min})/2$ and the standard deviation $\sigma_m = \sqrt{\sum_{i=1}^{k}(P_{l_i}-\mu_m)^2/k}$. The height threshold in T_o is set as $T_{ho} = \mu_m - \sigma_m$. Some lower peaks are removed based on T_{ho}, and new peaks $P_{ho}: P_{l_1}, P_{l_2}, \ldots, P_{l_h}$ are generated.

Step 4: Remove some peaks according to width threshold T_{wo}. The threshold $T_{wo} = 20$ is set based on the assumption that there should be no more than three dominant colors in T_o object. For two adjacent peaks P_{l_i} and P_{l_j}, if $(l_j - l_i) < T_{wo}$, then keep the peak with greater value and remove another peak from P_{ho}.

Step 5: Output the final peak sequence P_{no}, and dominant colors of T_o are determined as $C_o: l_1, l_2, \ldots, l_n$.

Similar to the manner of salient region T_o, the dominant colors C_b of background T_b can be obtained too. After the above processing, representative pixels that locate in salient region T_o and with dominant color C_o, as well as those locate in background T_b and with dominant color C_b, can be selected. Meanwhile, SVM training dataset (positive training set TS_p and negative training set TS_n) consisting of these representative pixels, can be constructed as

$$\begin{aligned} TS_p &= \{(x,y) | f(x,y) = i, (x,y) \in T_o, i \in C_o\} \\ TS_n &= \{(x,y) | f(x,y) = i, (x,y) \in T_b, i \in C_b\} \end{aligned} \tag{17}$$

Due to the global saliency information, spatial location and local color feature are all considered, training datasets derived from this method have some advantages over existed methods [19–21]: without human intervention, fully representing image characteristic distribution, strong robustness and computational efficiency.

3.3. Saliency-SVM training and segmentation

Generally, the total number of SVM training dataset is too large to be used for training directly. Therefore, training sample selection is one of the major factors determining to what degree the SVM classification rules can be generalized to unseen samples. A previous study showed that this factor could be more important for obtaining accurate classifications than the selection of classification algorithms. For SVM based image segmentation, training pixels can be selected in many ways. A commonly used sampling method is to identify and label small patches of homogeneous pixels in an image, as described in [19]. However, this manner may have some disadvantages: (1) Adjacent pixels tend to be spatially correlated or have similar values. Training samples collected this way underestimate the spatial variability of each class and are likely to give degraded classification, (2) segmentation results heavily depend on the training sample selection, which is a very skillful task. A freshman often fails to provide effective ones and more interactions are required for re-correcting. (3) In some cases, human interactions are not always feasible. A simple method to minimize the effect of spatial correlation is random sampling, which results in instability of classification.

Furthermore, when mapped to a higher feature space, training data with the same color value in a small local area may be redundant to learn the separating hyperplane, so selecting central pixel to replace the surrounding area is a way to reduce redundancy and improve the learning efficiency. Therefore, a local neighborhood homogeneity criterion is adopted to select small part of pixels in TS_p and TS_n as training samples for SVM training.

For a pixel $p(i,j)$ in training set TS_p or TS_n, its local homogeneity in $n \times n$ neighborhood is measured as

$$M_p = D_p^{n \times n} = \Sigma_{q \in N_p^{n \times n}} d(p,q) \tag{18}$$

where $d(p,q)$ is the Euclidean color distance between pixel p and q in quantized HSV color space, $N_p^{n \times n}$ is the pixel set of adjacent neighbors of pixel p.

As the color difference at lower level can indicate more intuitive local homogeneity, pixels that meet $M_p \le T_{lh}$ (T_{lh} is the local homogeneity threshold) in TS_p and TS_n will be selected as training samples of SVM. The greater the threshold value, the more pixels ultimately selected to train SVM classifier, and vice versa.

Because global saliency information and local color feature are all considered, training pixels selected from the previous stage can reduce the influence caused by random sampling. After the training pixels are selected, next we should specify which features extracted from training pixels should be provided as input vectors to train the SVM classifier. In this paper, for each training pixel, nine features are extracted to create the input vectors, include (1) four color features: r, g, b values in RGB color space and intensity i; (2) two texture features: Gabor filter is adopted as [20], e denotes the maximum of the six coefficients of a pixel and g denotes the maximum of six gradient magnitudes; (3) two spatial features: since neighboring pixels always possess the similar class label, so two-dimensional coordinates x and y of the training pixel are used; (4) the saliency feature s of training pixel. And each feature has been linearly scaled to the range [0, 1.0] to avoid features in bigger numeric ranges dominating those in smaller numeric ranges.

For the sake of convenience, LibSVM toolbox [42] is applied when Saliency-SVM is trained. Finally, all pixels in the given image are viewed as test set and classified with a label using the trained SVM model. As a result, the proposed Saliency-SVM method is summarized as follows:

Step 1: Detect and extract salient region T_o and background T_b as described in Section 3.1.

Step 2: Generate training datasets TS_p and TS_n of two regions T_o and T_b, respectively, by HSV quantization and histogram peak selection as detailed in Section 3.2.

Step 3: Select training samples from training datasets TS_p and TS_n according to the neighborhood homogeneity threshold, and extract feature vectors to train SVM model as described in Section 3.3.

Step 4: Segment out the whole salient object from the given image by the trained SVM model.

4. Experiment and discussion

The proposed Saliency-SVM aims to generate training dataset automatically for image segmentation, so to evaluate its performance, comprehensive experiments were conducted to compare it with other methods from different aspects. Test images in all experiments are selected from a benchmark dataset proposed in [37], which consists of 1000 images and ground truth segmentation results containing the most salient object for each image are provided too, where white area means the salient object and black area means background. The sizes of test images is all in 400 × 300 pixels. Parameters of the proposed Saliency-SVM method in all experiments are set as follows: window size for homogeneity $n=3$ and local homogeneity threshold $T_{lh}=0$.

Four evaluation metrics used to quantitatively assess the segmentation performance were the following:

(1) The segmentation Error Rate (ER) is defined as

$$ER=\frac{(N_f+N_m)}{N_t}\times 100\% \qquad (19)$$

where N_f is the number of false-segmented image pixels, N_m denotes the number of miss-segmented image pixels, and N_t is the total number of image pixels.

(2) Global Consistency Error (GCE) [43] measures the extent to which one segmentation can be viewed as a refinement of the other. Segmentations which are related in this manner are considered to be consistent, since they could represent the same natural image segmented at different scales. This measure allows for refinement, but suffers from degeneracy. Let $R(S,p_i)$ be the set of pixels in segmentations S that contains pixel p_i, the local refinement error is defined as

$$E(S_1,S_2,p_i)=\frac{|R(S_1,p_i)\backslash R(S_2,p_i)|}{|R(S_1,p_i)|} \qquad (20)$$

This error is not symmetric w.r.t. the compared segmentations, and takes the value 0 when S_1 is a refinement of S_2 at pixel p_i, then GCE is defined as

$$GCE(S_1,S_2)=\frac{1}{n}\min\left\{\sum_i E(S_1,S_2,p_i),\sum_i E(S_2,S_1,p_i)\right\} \qquad (21)$$

(3) Probabilistic Rand Index (PRI) [44] is commonly used to measure the similarity between two clusterings. In our experiments, it is employed to count the fraction of pairs of pixels whose labels are consistent between the compared segmentation S_c and the ground truth segmentation S_g. PRI is defined as

$$PRI(S_c,S_g)=\frac{1}{\binom{N}{2}}\sum_{i,j}[c_{ij}p_{ij}+(1-c_{ij})(1-p_{ij})] \qquad (22)$$

where N is the number of pixels, and p_{ij} is the ground truth probability that $\prod(l_i=l_j)$, and $c_{ij}=\prod(l_i^{S_c}=l_j^{S_c})$. The PRI has a value in the interval [0, 1], with 0 indicates that the two segmentations do not agree on any pair of pixels and 1 indicates that the compared segmentation S_c is exactly the same as the ground truth segmentation S_g.

(4) Variation of Information (VI) introduced in [45] measures the distance between two clusterings in terms of the information difference between them. As image segmentation can be seen as a clustering problem, the VI metric is defined as the distance between two segmentations as the average conditional entropy of one segmentation given the another. It can roughly measures the amount of randomness in one segmentation which cannot be explained by the other.

$$VI(S_c,S_g)=H(S_c)+H(S_g)-2I(S_c,S_g) \qquad (23)$$

where H and I represent the entropies and the mutual information between the compared segmentation S_c and ground truth S_g, respectively.

4.1. Comparison with other SVM based methods

In the first experiment, our objective was to compare image segmentation performance of various methods based on SVM model. Because the original codes of methods described in [19,20] are not available on the internet, we implemented the author's codes according to the algorithms described in their papers as accurately as possible. In addition, we also implemented a traditional SVM based image segmentation method using human interaction. So the proposed Saliency-SVM was compared with three SVM based methods: interactive SVM based method, FSVM described in [19] and FCM-SVM method described in [20] (referred to as SSVM, ISVM, FSVM and FCM-SVM respectively hereinafter). For fair comparison, four compared methods are all implemented in Matlab, and LibSVM toolbox [42] is used in order to facilitate the calculation. For ISVM and FSVM approaches, positive and negative training pixels are specified by the user, but the main difference is training pixels of ISVM method are some pixels at random positions selected by user mouse, while that of FSVM are pixels in two specified rectangles. And the initial cluster number of FCM algorithm in FCM-SVM method is set to 2.

Some visual comparison of salient object segmentation results using SSVM, ISVM, FSVM and FCM-SVM methods are shown in Fig. 5. It can be seen that SSVM outperforms the other three methods and achieves the best performance. Most salient objects in five test images (img1–img5) can be segmented effectively from background by SSVM method in Fig. 5(e) compared with ISVM in Fig. 5(b), FSVM in Fig. 5(c) and FCM-SVM in Fig. 5(d). In general, segmentation results of SSVM method are the closest to the ground truth segmentation results in most cases (see Fig. 5(e) and (f)). Moreover, some detailed shape information of salient objects can also be extracted by SSVM method, such as the black center and the stem of the flower in the second test image. Additionally, small salient object as in the first test image, the bottle besides the people can be segmented completely by SSVM method. The detailed information of segmented objects can be helpful for succeeding task such as object recognition and scene understanding. While segmentation results of ISVM, FSVM and FCM-SVM methods are not quite correct compared with that of ground truth. There are some pixels false-segmented or miss-segmented in object and background, which lead to the lower segmentation accuracy and insufficient visual effect.

In addition, the influence of different numbers of training pixels to the final segmentation result was also tested in our experiment. For FCM-SVM, $N_j/10$ image pixels were selected as training pixels as set in [20], where N_j means the number of pixels in the jth FCM clustering result. For FSVM method, 800 training pixels in two specified rectangles are selected in each test images at first, then after the Gaussian model prune and a projection process, the sizes of the reduced training set are 140, 132, 170, 148 and 156 for img1–img5 respectively. For ISVM method, the number of training pixels was tuned in a series of experiments, and we found that when there are 30 positive training pixels and 30 negative training pixels specified uniformly by the user, the segmentation result of ISVM method achieves the relatively optimal effect. And segmentation results comparison of ISVM using different number of training pixels are presented in Fig. 6, of which training pixels of salient object (marked with red cross) and background (marked with green circle) are in the

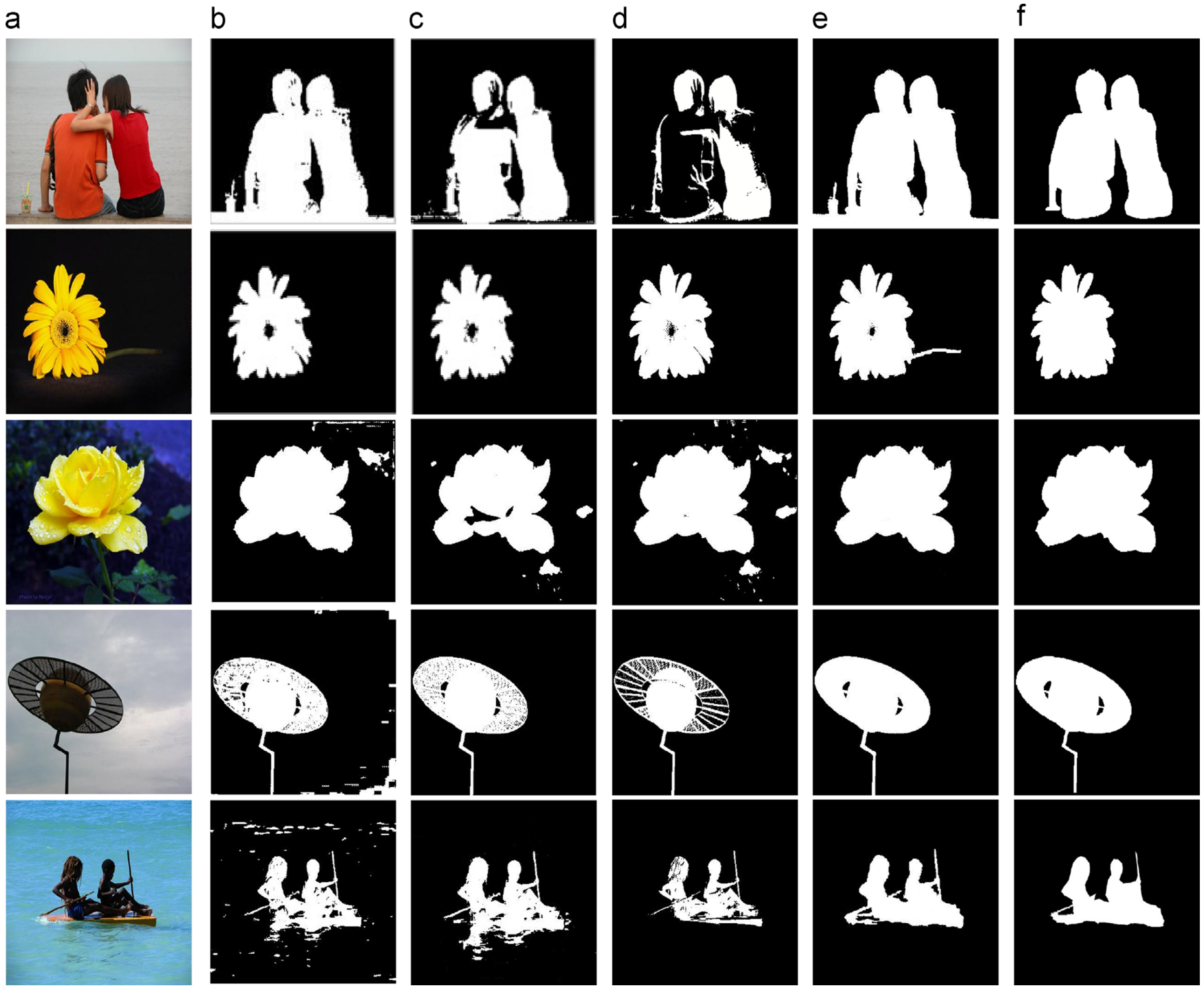

Fig. 5. Segmentation result comparison of ISVM, FSVM, FCM-SVM and SSVM for test img1–img5. (a) input images; (b) segmentation results of ISVM (using 60 training pixels); (c) segmentation results of FSVM; (d) segmentation results of FCM-SVM; (e) segmentation results of SSVM; (f) ground truth segmentations.

top row and corresponding segmentation results are in the bottom row. As can be seen that, only exactly right specified training pixels of salient object and background can lead to a complete salient object segmentation in Fig. 6(a), and fewer right training pixels produce segmentation results with noises as shown in Fig. 6(b). While training pixels specified failed to represent the object and background features lead to bad segmentation results such as object lost and background lost (see Fig. 6(c) and (d)). Likewise, segmentation performance of FSVM method also suffers from the same drawback mentioned above, i.e., ISVM and FSVM methods may fail to work when the training rectangles are specified wrongly, especially for objects with heterogeneous color features. For example in test img1, only a group of pixels in the people's head and body were all specified as positive training samples, the segmentation results of ISVM and FSVM methods would be correct. It follows that the segmentation performance of this kind of SVM based method heavily depends on the number and distribution of training pixels. While training pixels of SSVM are identified automatically via local homogeneity criterion and trimap stemmed from visual saliency detection, they can properly represent the feature distribution of salient object and background in the test images, thus effectively reduce the influence of number and distribution of training pixels to the ultimate segmentation result.

In conclusion, segmentation results of ISVM, FSVM and FCM-SVM are inferior to that of the proposed SSVM method visually, and quantitative evaluation metrics (ER, GCE, PRI and VI) also support this conclusion, as listed in Table 1, in which the black values indicate the best results. And ↑ means the larger the metric, the better the segmentation result, and vice versa. It can be seen that SSVM method is better than the other three methods in most cases. The mean ER, GCE, PRI and VI values of five test images are 2.80%, 0.02, 0.95 and 0.20, respectively. We also evaluate the segmentation performance of four compared methods on other images of the database, and comparison results are consistent with the above conclusion in a large part.

Furthermore, Table 2 gives the comparative results in classification accuracy (CA), the number of support vectors (Num), CPU time for training and segmenting of four methods. It can be seen that CPU time of SSVM is significantly less than the time required by the algorithms of ISVM, FSVM and FCM-SVM. Meanwhile, fewer support vectors are used for training process but yield lower generalization error.

4.2. Comparison with binary segmentation

As SVM is a popular binary classification algorithm, and SVM-based image segmentation task can also be viewed as a binary classification problem, so in the second experiment, we evaluated the performance of SSVM method and a typical adaptive threshold binary segmentation method [46].

Fig. 7 illustrates some test image segmentation results comparison using SSVM and threshold method (img6–img9). It can be

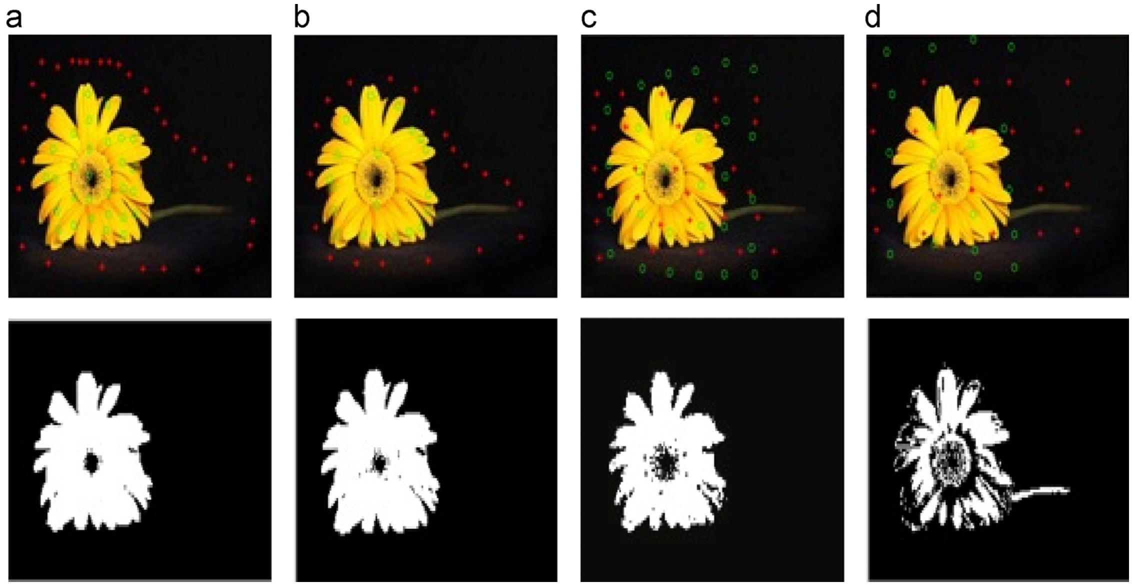

Fig. 6. Segmentation results comparison of different number training pixels by ISVM method. (a) 60 right specified training pixels; (b) 40 right specified training pixels; (c) 60 wrong specified training pixels; (d) 40 wrong specified training pixels.

Table 1
Quantitative comparison (ER, GCE, PRI and VI) of ISVM, FSVM, FCM-SVM and SSVM for test img1–img5.

	ER (%) ↓				GCE ↓				PRI ↑				VI ↓			
	ISVM	FSVM	FCM-SVM	SSVM	ISVM	FSVM	FCM-SVM	SSVM	ISVM	FSVM	FCM-SVM	SSVM	ISVM	FSVM	FCM-SVM	SSVM
img1	9.5	9.7	24.6	**5.3**	0.07	0.08	0.25	**0.02**	0.89	0.83	0.76	**0.95**	0.29	0.31	0.40	**0.26**
img2	3.9	4.0	4.1	**3.4**	0.04	0.05	0.07	**0.03**	0.93	0.93	0.94	**0.95**	0.37	0.36	0.36	**0.29**
img3	5.7	7.6	7.3	**0.2**	0.26	0.30	0.29	**0.01**	0.90	0.88	0.85	**0.96**	0.08	0.13	0.12	**0.04**
img4	8.2	8.7	11.6	**0.4**	0.20	0.21	0.22	**0.02**	0.91	0.90	0.89	**0.92**	0.19	0.20	0.23	**0.14**
img5	5.9	9.1	5.4	**4.7**	0.07	0.09	0.06	**0.03**	0.92	0.89	0.91	**0.97**	0.21	0.24	0.20	**0.18**

Table 2
Quantitative comparison (CA, Num and CPU time) of ISVM, FSVM, FCM-SVM and SSVM for test img1–img5.

	CA (%) ↑				Num ↓				CPU ↓			
	ISVM	FSVM	FCM-SVM	SSVM	ISVM	FSVM	FCM-SVM	SSVM	ISVM	FSVM	FCM-SVM	SSVM
img1	90.5	90.7	75.4	**94.7**	42	15	34	**9**	9.0	7.5	8.9	**3.5**
img2	96.1	96.0	95.9	**96.4**	32	16	30	**12**	8.5	6.8	9.1	**2.7**
img3	94.3	92.4	92.7	**99.8**	36	10	28	**6**	7.9	7.2	7.5	**3.1**
img4	91.8	91.3	88.4	**99.6**	37	18	39	**10**	8.1	7.3	8.3	**3.4**
img5	94.1	90.9	94.6	**95.3**	54	20	45	**15**	5.0	6.7	8.9	**3.3**

seen that segmentation results of SSVM are much better than that of the threshold method. Only those images who have obvious difference between object and background, threshold method can obtain better segmentation effect. Most objects segmented by threshold method are with noises or incomplete edges. Corresponding quantitative evaluation metrics are listed in Table 3.

So in brief, the proposed SSVM method using traditional SVM model combined with global and local image cues derived from visual saliency detection is much better than the naive binary classification method for image segmentation.

4.3. Comparison with automatic segmentation methods

One advantage of the proposed Saliency-SVM method is that neither the prior knowledge of the image nor any human intervention is needed. In other words, each step in this algorithm is automatic, so in the third experiment, we compared the proposed SSVM method with other two popular automatic methods: NCuts [11] and MeanShift [47].

Fig. 8 shows some visual results comparison obtained by three methods for four test images (img10–img13). As can be seen that segmentation results of Ncuts method are not accurate enough, and usually accompanied with information loss in salient object. And Meanshift method produces lower quality segmentation results with noises and discontinuous edges of salient objects. While the proposed SSVM method can obtain good segmentation results closest to the ground truth results in all test images. And corresponding quantitative evaluation metrics are listed in Table 4.

4.4. Performance analysis of other factors

Finally in our experiments, we evaluated the segmentation performance of the proposed SSVM method from other aspects, such as kernel function, model parameters and the number of training pixel.

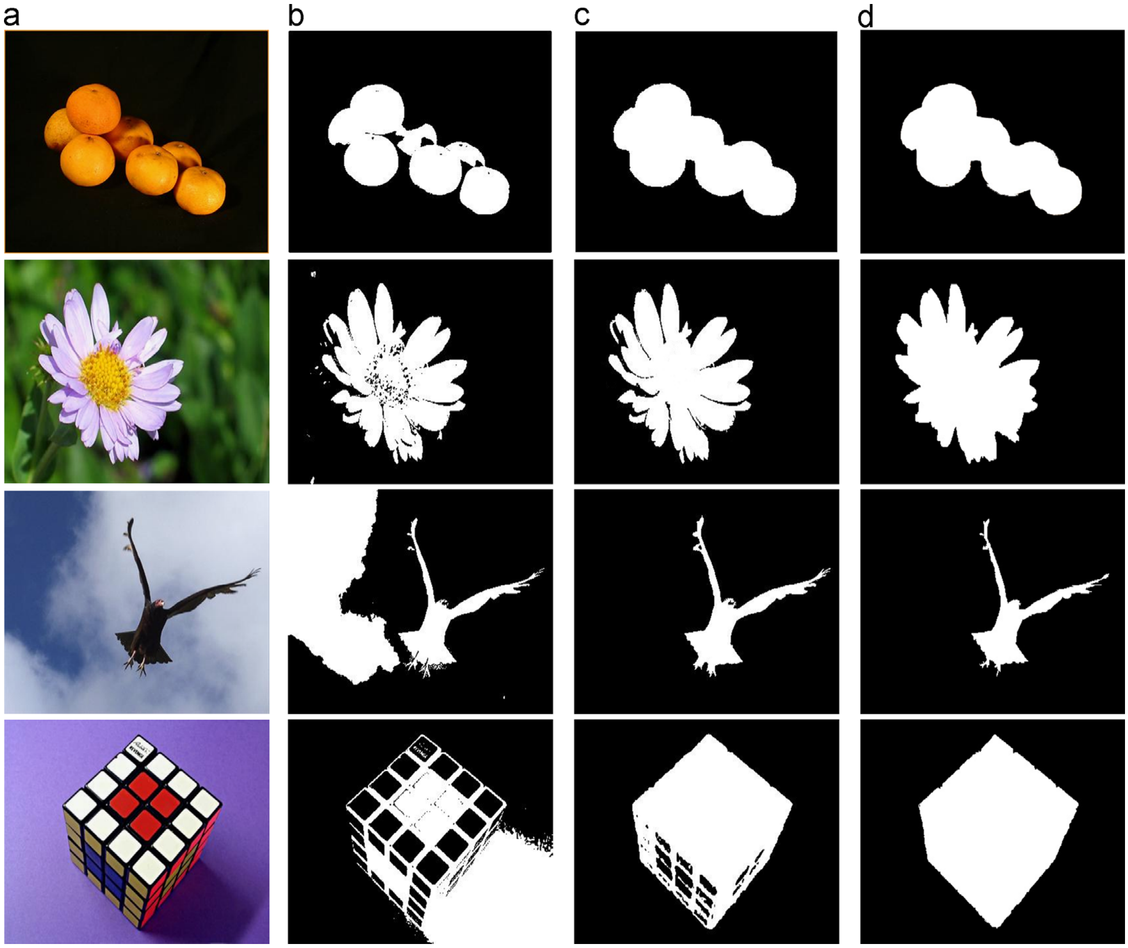

Fig. 7. Segmentation results comparison of SSVM and threshold method for test img6–img9. (a) input images; (b) segmentation results of threshold method; (c) segmentation results of SSVM; (d) ground truth segmentations.

Table 3
Quantitative comparison (ER, GCE, PRI and VI) of threshold method and SSVM for test img6–img9.

	ER (%) ↓		GCE ↓		PRI ↑		VI ↓	
	Threshold	SSVM	Threshold	SSVM	Threshold	SSVM	Threshold	SSVM
img6	2.5	**0.2**	0.05	**0.01**	0.94	**0.99**	0.07	**0.03**
img7	3.1	**1.4**	0.06	**0.08**	0.92	**0.95**	0.09	**0.06**
img8	29.7	**0.3**	0.38	**0.02**	0.81	**0.98**	0.21	**0.04**
img9	27.6	**3.5**	0.36	**0.11**	0.82	**0.91**	0.22	**0.08**

The kernel function in SVM classifier plays an important role of implicitly mapping the input vector into a high-dimensional feature space. But there are no perfect approaches available to “learn” the form of kernel. Common choices of kernel function are the linear kernel, polynomial kernels and Gaussian RBFs. To optimize these parameters, a 5-fold cross validation procedure is used for training and testing the SSVM classifier with various models and parametric setting on test img2 with ground truth segmentation result. Table 5 gives the comparative results in classification accuracy of different kernels at different regularization parameter C. It can be seen that the kernel function and model parameters have a certain influence on the segmentation performance, and for test img2, the segmentation performance is the best by using RBF kernel with $\sigma^2=0.25$ at $C=1$.

On the other hand, the classification performance of SVM classifier heavily depends on the number of training examples. In order to evaluate the relation between the segmentation performance of SSVM and the number of training pixels, a series of the number of training pixels are set to train the SSVM and its segmentation performance is evaluated. Training pixels of SSVM method are selected from training set TS_p and TS_n based on the local homogeneity threshold t_{lh}, and the smaller the t_{lh} value, the fewer pixels selected for SVM training, and vice versa.

Table 6 gives the comparative results in classification accuracy, the number of support vectors, CPU time of different number of training pixels based on different local homogeneity thresholds. It can be seen that although the number of support vectors, the training time and the segmenting time are changed to varying degrees as the number of training pixels increases, the classification accuracy is slightly changed. When the threshold value t_{lh} is very high, it means that the pixels in training sets TS_p and TS_n are all selected as training pixels, which leads to more training time. Conversely, when the threshold t_{lh} is set to zero, only those with the highest homogeneity are selected as training pixels. Thus, the training time is rapidly reduced, but the classification accuracy has little effect. Therefore, the homogeneity criterion for selecting reprehensive pixels in local region is effective in terms of

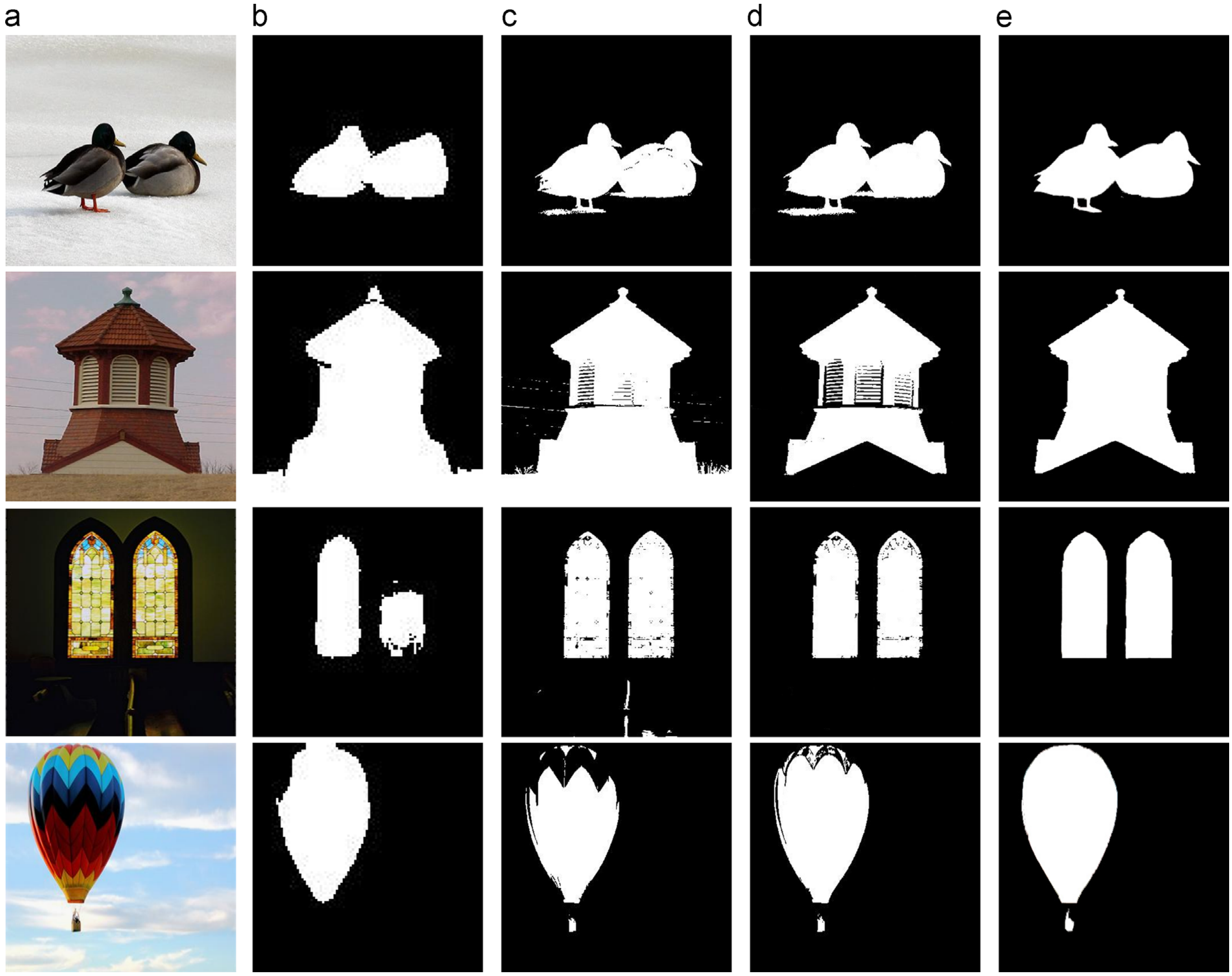

Fig. 8. Segmentation results comparison of NCuts, MeanShift and SSVM for test img10–img13. (a) Input images; (b) segmentation results of NCuts; (c) segmentation results of MeanShift; (d) segmentation results of SSVM; (e) ground truth segmentations.

Table 4
Quantitative comparison (ER, GCE, PRI and VI) of NCuts, MeanShift and SSVM for test img10–img13.

	ER (%) ↓			GCE ↓			PRI ↑			VI ↓		
Test image	NCuts	MeanShift	SSVM	NCuts	MeanShift	SSVM	NCuts	MeanShift	SSVM	NCuts	MeanShift	SSVM
img10	4.9	2.1	**2.0**	0.08	0.02	**0.01**	0.90	0.97	**0.96**	0.49	0.19	**0.18**
img11	23.3	20.1	**4.1**	0.29	0.27	**0.07**	0.64	0.91	**0.92**	1.20	1.07	**0.46**
img12	9.6	5.3	**3.1**	0.15	0.08	**0.04**	0.83	0.75	**0.95**	0.83	0.35	**0.29**
img13	4.6	8.6	**2.1**	0.08	0.13	**0.04**	0.91	0.83	**0.96**	0.50	0.73	**0.28**

segmenting speed and accuracy, especially for real-time image segmentation tasks.

From above experiments and analysis, it can be concluded that the performance of the proposed SSVM method is superior to that of ISVM, FSVM, FCM-SVM, threshold, Meanshift and Ncuts. Segmentation results of SSVM are the closest to the ground truth results in most cases. Experiment results on the whole test dataset support this conclusion but not listed due to the limit of paper length. And quantitatively comparison of average values of 5 metrics (ER, GCE, PRI, VI and CPU time) on the whole test dataset are almost consistent with the visual comparison, as shown in Table 7. Obviously, the proposed SSVM method is better than other methods in 4 evaluation metrics (ER, GCE, PRI, and VI), but the average CPU time of SSVM is slightly inferior to that of threshold method. Because each pixel is taken as a node of a graph in Ncuts method, and Meanshift method needs a iterative process, the computational cost of these two methods are relatively high. Above experiment results demonstrate that the proposed SSVM method can automatically and effectively select training pixels and segment out salient object in relatively less times. But it is worth mentioning that because saliency map was not originally proposed for the purpose of generic object segmentation, SSVM method is somewhat limited especially for images without specific target or object segmentation with cluttered background.

5. Conclusions and further work

In this paper, we present a novel SVM method based on visual saliency for automatic training data selection and image segmentation.

Table 5
Comparative results in classification accuracy of different kernels with different *C* and model parameters.

Kernel	Parameter	*C*				
		0.1	1	10	100	1000
Linear		90.1	90.4	93.5	92.1	89.6
Polynomial	$d=1$	90.3	89.6	89.0	88.6	87.4
	$d=2$	90.8	88.6	87.2	88.7	86.1
	$d=3$	89.6	88.5	85.7	83.9	83.1
RBF	$\sigma^2=0.125$	92.1	92.6	93.3	92.6	91.5
	$\sigma^2=0.25$	92.3	96.6	94.9	93.5	90.0
	$\sigma^2=0.5$	91.7	92.8	91.4	90.7	87.5
	$\sigma^2=1$	90.2	91.3	90.6	88.4	85.6
	$\sigma^2=2$	88.3	87.9	86.2	85.5	84.7

Table 6
Comparative results in classification accuracy, the number of support vectors, CPU time of different number of training pixels.

Value of t_{th}	0	10	20	50	100
classification accuracy (%)	96.6	95.8	96.8	96.9	96.9
Number of support vectors	12	36	59	138	246
CPU time	2.7	4.2	5.8	8.3	12.5

Table 7
Quantitative comparison of five evaluation metrics of seven methods on the whole database.

Test image	ER(%) ↓	GCE ↓	PRI ↑	VI ↓	CPU ↓
ISVM	6.13	0.14	0.91	0.26	3.53
FSVM	7.25	0.17	0.88	0.25	4.21
FCM-SVM	8.11	0.22	0.87	0.31	2.89
Threshold	12.1	0.13	0.81	0.74	**1.98**
Ncuts	8.87	0.13	0.85	0.71	8.52
Meanshift	5.37	0.11	0.89	0.58	9.62
SSVM	**4.26**	**0.07**	**0.92**	**0.21**	2.27

The advantages of the proposed Saliency-SVM include: (1) it exploits a visual saliency detection method that is independent of image features such as intensity, shape, texture or other prior knowledge of the given image; (2) the whole salient object with homogeneous features can be extracted without human intervention; (3) saliency-SVM for image segmentation is automatic, avoiding the processing of manual intervention for selecting training data. In addition to these advantages, the proposed Saliency-SVM has its limitation too as mentioned previously, due to the training data selection relies on the result of visual saliency detection. In further research on this topic, we will consider more high-level image characteristics for training data selection, as well as to extend the application of the proposed Saliency-SVM to content-based image retrieval and face recognition.

Acknowledgements

The authors thank the helpful comments and suggestions form the anonymous reviewers as well as the Associated Editor, for their effort to improve the paper. The work described in this paper was partially supported by the National Natural Science Foundation of China (No. 61273291, 61303091), Research Project Supported by Shanxi Scholarship Council of China (No. 2012-008), Scientific and Technological Project of Shanxi Province (No. 20120321027-01), Open Project Foundation of Information Technology Research Base of Civil Aviation Administration of China (No. CAAC-ITRB-201305).

References

[1] H.D. Cheng, X.H. Jiang, Y. Sun, J.L. Wang, Color image segmentation: advances and prospects, Pattern Recogn. 34 (12) (2001) 2259–2281.
[2] J.E. Francisco, D.J. Allan, Benchmarking image segmentation algorithms, Int. J. Comput. Vis. 85 (2) (2009) 167–181.
[3] K. Hammouche, M. Diaf, P. Siarry, A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation, Comput. Vis. Image Underst. 109 (2) (2008) 163–175.
[4] M.H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, Expert Syst. Appl. 38 (1) (2011) 13785–13791.
[5] E. Saber, S.R. Vantaram, V. Amuso, M. Shaw, R. Bhaskar, Automatic image segmentation by dynamic region growth and multiresolution merging, IEEE Trans. Image Process. 18 (10) (2009) 2275–2288.
[6] J. Ye, G. Xu, A geometric flow approach for region-based image segmentation, IEEE Trans. Image Process. 21 (12) (2012) 4735–4745.
[7] H. Wang, J. Oliensis, Generalizing edge detection to contour detection for image segmentation, Comput. Vis. Image Underst. 114 (7) (2010) 731–744.
[8] A. Pablo, M. Michael, F. Charless, M. Jitendra, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 898–916.
[9] L.A. Vese, T.F. Chan, A multiphase level set framework for image segmentation using the Mumford and Shah model, Int. J. Comput. Vis. 50 (3) (2002) 271–293.
[10] S. Liu, Y. Peng, A local region-based Chan–Vese model for image segmentation, Pattern Recogn. 45 (7) (2012) 2769–2779.
[11] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[12] Z.M. Wang, Y.C. Soh, Q. Song, K. Sim, Adaptive spatial information-theoretic clustering for image segmentation, Pattern Recogn. 42 (9) (2009) 2029–2044.
[13] Z. Yu, O.C. Au, R. Zou, W. Yu, J. Tian, An adaptive unsupervised approach toward pixel clustering and color image segmentation, Pattern Recogn. 43 (5) (2010) 1889–1906.
[14] Z. Li, X.M. Wu, S.F. Chang, Segmentation using superpixels: a bipartite graph partitioning approach, in: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Providence, USA, June 2012, pp. 789–796.
[15] V. Vapnik, The Nature of Statistical Learning Theory, Spring-Verlag, New York, 2000.
[16] Y.H. Yu, C.C. Chang, Scenery image segmentation using support vector machines, Fund. Inform. 61 (2004) 379–388.
[17] P. Mitra, B.U. Shankar, S.K. Pal, Segmentation of multispectral remote sensing images using active support vector machines, Pattern Recogn. Lett. 25 (9) (2004) 1067–1074.
[18] B. Cyganek, Color image segmentation with support vector machines: applications to road signs detection, Int. J. Neural Syst. 18 (4) (2008) 339–345.
[19] Z. Yu, H.S. Wong, G. Wen, A modified support vector machine and its application to image segmentation, Image Vis. Comput. 29 (1) (2011) 29–40.
[20] X.Y. Wang, T. Wang, J. Bu, Color image segmentation using automatic pixel classification with support vector machine, Neurocomputing 74 (18) (2011) 3898–3911.
[21] Q. Zhao, Y. Hu, J. Cao, Automatic image segmentation based on saliency maps and Fuzzy SVM, in: Proceeding of IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009), December 2009, Shanghai, China, pp. 121–124.
[22] Y. Fu, J. Cheng, Z. Li, H. Lu, Saliency cuts: an automatic approach to object segmentation, in: Proceeding of 19th International Conference on Pattern Recognition, Tampa, USA, December 2008, pp. 1–4.
[23] Q. Li, Y. Zhou, J. Yang, Saliency based image segmentation, in: Proceeding of International Conference on Multimedia Technology, Hangzhou, China, July 2011, pp. 5068–5071.
[24] C.Y. Lee, J.J. Leou, H.H. Hsiao, Saliency-directed color image segmentation using modified particle swarm optimization, Signal Process. 92 (1) (2012) 1–18.
[25] H. Li, K.N. Ngan, Saliency model-based face segmentation and tracking in head-and-shoulder video sequences, J. Vis. Commun. Image Represent. 19 (5) (2008) 320–333.
[26] X. Hou, L. Zhang, Saliency detection: a spectral residual approach, in: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA, June 2007, pp. 1–8.
[27] C. Rother, V. Kolmogorov, A. Blake, "GrabCut": interactive foreground extraction using iterated graph cuts, ACM Trans. Graph. 23 (3) (2004) 309–314.
[28] C. Koch, S. Ullman, Shifts in selective visual attention: towards the underlying neural circuitry, Hum. Neurobiol. 4 (4) (1985) 219–227.
[29] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, IEEE Trans. Pattern Anal. Mach. Intell. 20 (11) (1998) 1254–1259.
[30] J. Harel, C. Koch, P. Perona, Graph-based visual saliency, in: Proceeding of 21st Annual Conference on Neural Information Processing Systems, Vancouver, Canada, December 2007, pp. 545–552.
[31] Y.F. Ma, H.J. Zhang, Contrast-based image attention analysis by using fuzzy growing, in: Proceeding of ACM International Conference on Multimedia, Berkeley, USA, November 2003, pp. 374–381.
[32] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, H.Y. Shum, Learning to detect a salient object, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2) (2011) 353–367.

[33] S. Goferman, L. Zelnik-Manor, A. Tal, Context-aware saliency detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (10) (2012) 1915–1926.

[34] M.M. Cheng, G.X. Zhang, N.J. Mitra, X. Huang, S.M. Hu, Global contrast based salient region detection, in: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Providence, USA, June 2011, pp. 21–23.

[35] C. Guo, Q. Ma, L. Zhang, Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform, in: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA, June 2008, pp. 23–28.

[36] C. Guo, L. Zhang, A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression, IEEE Trans. Image Process. 19 (1) (2010) 185–198.

[37] R. Achanta, S. Hemami, F. Esgtrada, S. Süsstrunk, Frequency-tuned salient region detection, in: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Miami, USA, June 2009, pp. 1597–1604.

[38] R. Achanta, S. Süsstrunk, Saliency detection using maximum symmetric surround, in: Proceeding of 17th IEEE International Conference on Image Processing, Hong Kong, China, September 2010, pp. 2653–2656.

[39] P. Brigger, J.R. Casas, M. Pardas, Morphological operators for image and video compression, IEEE Trans. Image Process. 5 (6) (1996) 881–898.

[40] T.W. Chen, Y.L. Chen, S.Y. Chien, Fast image segmentation based on K-Means clustering with histograms in HSV color space, in: Proceeding of IEEE Workshop on Multimedia Signal Processing, Cairns, Australia, October 2008, pp. 322–325.

[41] L. Zhang, F.Z. Lin, B. Zhang, A CBIR method based on color-spatial feature, in: TENCON 99. Proceedings of the IEEE Region 10 Conference, Cheju Island, Korea, 1999, pp. 166–169.

[42] LIBSVM, A library for support vector machines, Available from: ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm/⟩.

[43] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceeding of International Conference on Computer Vision, Vancouver, USA, July 2001, pp. 416–423.

[44] R. Unnikrishnan, C. Pantofaru, M. Hebert, Toward objective evaluation of image segmentation algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 29 (6) (2007) 929–943.

[45] M. Meila, Comparing clusterings – an information based distance, J. Multivar. Anal. 98 (5) (2007) 873–895.

[46] N. Otsu, A threshold selection method from gray level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.

[47] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intell. 24 (5) (2002) 603–619.

Xuefei Bai received the B.S. degree in computer science from Taiyuan Normal University, China, in 2002, the M.S. degree in computer science from Northwest University, China, in 2005, and is currently a Ph.D. Candidate in School of Computer and Information Technology at Shanxi University. Her current research interests include image processing, computer vision, virtual reality and machine learning.

Wenjian Wang received the B.S. degree in computer science from Shanxi University, China, in 1990, the M.S. degree in computer science from Hebei Polytechnic University, China, in 1993, and Ph.D. degree in applied mathematics from Xi'an JiaoTong University, China, in 2004. She worked as a research assistant at the Department of Building and Construction, The City University of Hong Kong from May 2001 to May 2002. She has been with the Department of Computer Science at Shanxi University since 1993, where she was promoted as Associate Professor in 2000 and as Full Professor in 2004, and now serves as a Ph.D. supervisor in Computer Application Technology and System Engineering. She has published more than 40 academic papers on machine learning, computational intelligence, and data mining. Her current research interests include neural networks, support vector machines, machine learning theory and environmental computations.