Chapter 2
Learning and Recognition Methods for Image Search and Video Retrieval

Ajit Puthenputhussery, Shuo Chen, Joyoung Lee, Lazar Spasovic and Chengjun Liu

Abstract Effective learning and recognition methods play an important role in intelligent image search and video retrieval. This chapter therefore reviews some popular learning and recognition methods that are broadly applied for image search and video retrieval. First some popular deep learning methods are discussed, such as the feed-forward deep neural networks, the deep autoencoders, the convolutional neural networks, and the Deep Boltzmann Machine (DBM). Second, Support Vector Machine (SVM), which is one of the popular machine learning methods, is reviewed. In particular, the linear support vector machine, the soft-margin support vector machine, the non-linear support vector machine, the simplified support vector machine, the efficient Support Vector Machine (eSVM), and the applications of SVM to image search and video retrieval are discussed. Finally, other popular kernel methods and new similarity measures are briefly reviewed.

2.1 Introduction

The applications in intelligent image search and video retrieval cover all corners of our society from searching the web to scientific discovery and societal security. For example, the New Solar Telescope (NST) at the Big Bear Solar Observatory (BBSO) produces over one terabytes of data daily, and the Solar Dynamic Observatory (SDO)

A. Puthenputhussery (✉) · J. Lee · L. Spasovic · C. Liu (✉)
New Jersey Institute of Technology, Newark, NJ 07102, USA
e-mail: avp38@njit.edu

C. Liu
e-mail: chengjun.liu@njit.edu

J. Lee
e-mail: jo.y.lee@njit.edu

L. Spasovic
e-mail: spasovic@njit.edu

S. Chen (✉)
The Neat Company, Philadelphia, PA 19103, USA
e-mail: sc77@njit.edu

C. Liu (ed.), *Recent Advances in Intelligent Image Search and Video Retrieval*,
Intelligent Systems Reference Library 121, DOI 10.1007/978-3-319-52081-0_2

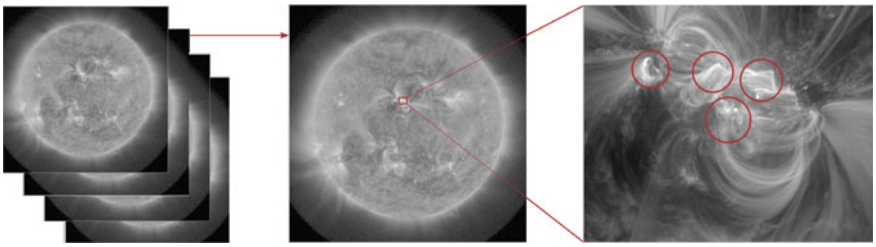

Fig. 2.1 Video analysis in solar physics for solar event detection

launched by NASA generates about four Terabytes of video data per day. Such huge amount of video data requires new digital image and video analysis techniques to advance the state-of-the-art in solar science. Motion analysis methods based on the motion field estimation and Kalman filtering may help characterize the dynamic properties of solar activities in high resolution. Figure 2.1 shows an example for solar features and events detection. First, a sequence of SDO images is processed based on motion field analysis to locate candidate regions of interest for features and events. The differential techniques that apply the optical flow estimation and the feature-based techniques that utilize feature matching and Kalman tracking approaches may be applied to locate the candidate regions for features and events. Other techniques, such as the conditional density propagation method and the particle filtering method, will further enhance the localization performance. Second, a host of feature and event detection and recognition methods will then analyze the candidate regions for detecting features and events as indicated in the right image in Fig. 2.1.

Another example of video-based applications for societal security is police body-worn cameras, which present an important and innovative area of criminal justice research with the potential to significantly advance criminal justice practice. The Community Policing Initiative proposed by the White House would provide a 50% match to states that purchase such cameras [83], and the recent Computing Community Consortium whitepaper on body-worn cameras released by a panel of computer vision experts and law enforcement personnel recommends increased research funding for technology development [17]. Figure 2.2 shows the idea of innovative police body-worn cameras that recognize their environment. Specifically, advanced face detection and facial recognition technologies, which are robust to challenging factors such as variable illumination conditions, may be applied for suspect detection in video to improve public safety and well-being of communities. The state-of-the-art image indexing and video retrieval methods should be utilized for searching, indexing, and triaging the large amount of video data in order to meet various criminal justice needs, such as forensic capabilities and the freedom of information act (FOIA) services.

This chapter reviews some representative learning and recognition methods that have broad applications in intelligent image search and video retrieval. We first discuss some popular deep learning methods, such as the feedforward deep neural

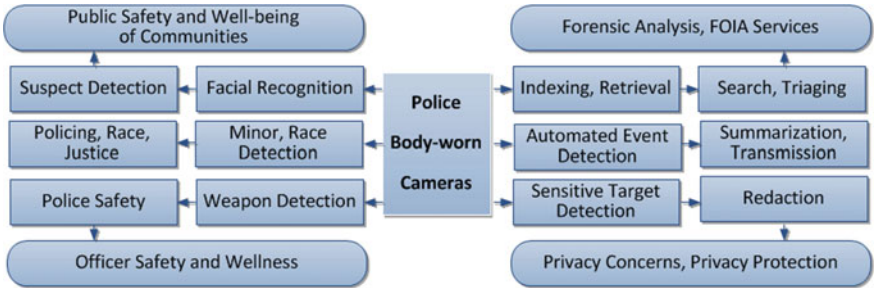

Fig. 2.2 Innovative police body-worn cameras that recognize their environment

networks [35, 49], the deep autoencoders [2, 81], the convolutional neural networks [23], and the Deep Boltzmann Machine (DBM) [22, 61]. We then discuss one of the popular machine learning methods, namely, Support Vector Machine (SVM) [77]. Specifically we review the linear support vector machine [78], the soft-margin support vector machine [78], the non-linear support vector machine [77], the simplified support vector machine [11, 54], the efficient Support Vector Machine (eSVM) [13, 14], and the applications of SVM to image search and video retrieval [25, 37, 52, 62, 75, 87]. We finally briefly review some other popular kernel methods and new similarity measures [40, 41, 44].

2.2 Deep Learning Networks and Models

Deep artificial neural network is an emerging research area in computer vision and machine learning and has gained increasing attention in recent years. With the advent of big data, the need for efficient learning methods to process an enormous number of images and videos for different kinds of visual applications has greatly increased. The task of image classification is a fundamental and important computer vision and machine learning problem wherein after learning a model based on a set of training and validation data, the labels for the test data have to be predicted. Image classification is a challenging problem as there can be many variations in the background noise, illumination conditions, as well as multiple poses, distortions and occlusions in the image. In recent years, different deep learning methods such as the feedforward deep neural networks [35, 49], the deep autoencoders [2, 81], the convolutional neural networks [23], and the Deep Boltzmann Machine (DBM) [22, 61], have been shown to achieve good performance for image classification problems. One possible reason for the feasibility of the deep learning methods is due to the discovery of multiple levels of representation within an image that leads to a better understanding of the semantics of the image.

2.2.1 Feedforward Deep Neural Networks

In this section, we discuss the architecture, the different layers, and some widely used activation functions in the feedforward deep neural networks. A feedforward deep neural network can be considered as an ensemble of many units that acts as a parametric function with many inputs and outputs to learn important features from the input image. Feedforward deep neural networks are also called multilayer perceptrons [59] and typically contain many hidden layers. Figure 2.3 shows the general architecture of a deep feedforward neural network with N hidden layers. Now let's consider a feedforward network with one hidden layer. Let the input layer be denoted as $\mathbf{I}$, the output layer as $\mathbf{O}$, the hidden layer as $\mathbf{H}$, and the weight vector for connections from the input layer to the hidden layer as $\mathbf{W}_1$. Therefore, the hidden unit vector can be computed as $\mathbf{H} = f(\mathbf{W}_1^T\mathbf{I} + \mathbf{b}_1)$, where $\mathbf{b}_1$ is the bias vector for the hidden layer and $f(.)$ is the activation function. Similarly, the output vector can be computed as $\mathbf{O} = f(\mathbf{W}_2^T\mathbf{H} + b_2)$, where $\mathbf{W}_2$ is the weight matrix for connections from the hidden layer to the output layer and $\mathbf{b}_2$ is the bias vector for the output layer.

The feedforward deep neural network can be optimized with many different optimization procedures but a common approach is to use momentum based stochastic gradient descent. An activation function takes in an input and performs some mathematical operation so that the output lies within some desired range. We next discuss some activation functions that are commonly used in the literature for deep neural networks. The rectified linear unit (ReLU) is the most popular activation function used for deep neural networks. It acts as a function that thresholds the input value at zero and has the mathematical form $g(y) = \max(0, y)$. Some advantages of the

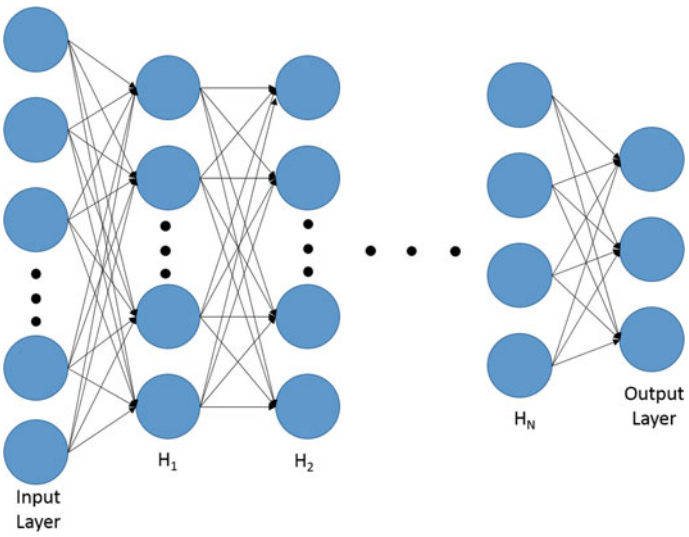

Fig. 2.3 The general architecture of feedforward deep neural networks

ReLU include that it helps the stochastic gradient descent process to converge faster [35] and can be implemented as a lightweight operation. Many different variants of the ReLU have been proposed to improve upon the ReLU. A leaky ReLU was propose by Maas et al. [49] wherein the function would never become zero but will be equal to a small constant. The leaky ReLU has the form $f(y) = \max(0, y) + c\min(0, y)$, where c is a small constant. Another variant known as the PReLU was proposed by He et al. [27] which considers c as a parameter learned during the training process. The sigmoid activation function produces an output in the range between 0 and 1 and has the form $\sigma(y) = 1/(1 + e^{-y})$. Some issues with the sigmoid activation function are that the output produced is not centered and it reduces the gradient to zero. The tanh activation function has the mathematical form $\tanh(y) = 2\sigma(2y) - 1$ and is a scaled version of the sigmoid activation function. It produces a centered output between -1 and 1, and overcomes the disadvantage of the sigmoid activation function.

2.2.2 *Deep Autoencoders*

The autoencoders are based on an unsupervised learning algorithm to develop an output representation that is similar to the input representation with the objective of minimizing the loss of information. Figure 2.4 shows the general architecture of a deep autoencoder where encoding takes place to transform the input vector into a compressed representation and decoding tries to reconstruct the original representation with minimum distortion [2, 81]. Let I be the set of training vectors, and the autoencoder problem may be formulated as follows: finding $\{\mathbf{W}_1, \mathbf{W}_2, b_1, b_2\}$ from $\mathbf{H} = f(\mathbf{W}_1^T\mathbf{I} + \mathbf{b}_1)$ and $\mathbf{O} = f(\mathbf{W}_2^T\mathbf{H} + b_2)$, such that $\mathbf{O}$ and $\mathbf{I}$ are similar with the

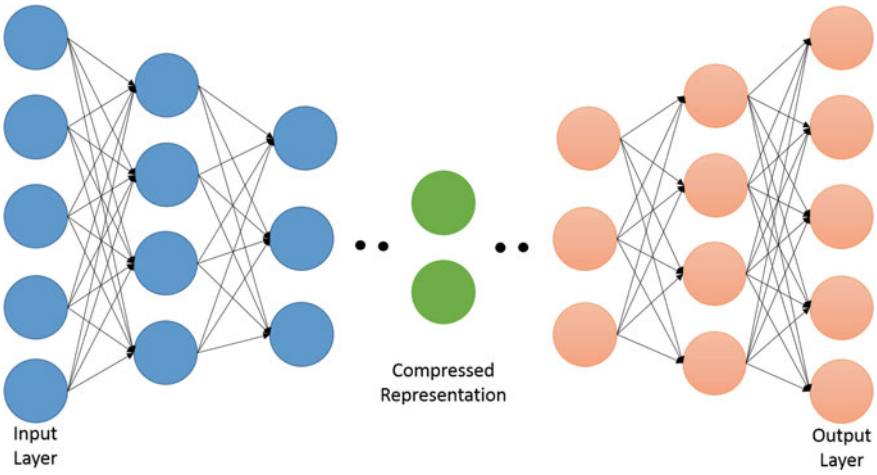

Fig. 2.4 The general architecture of a deep autoencoder

minimum loss of information. A popular optimization procedure used for solving the autoencoders is the back propagation method for computing the gradient weights.

Different variants of autoencoders have been proposed so as to make them suitable for different applications. A sparse deep autoencoder [21, 57] integrates a sparsity criterion into the objective function to learn feature representation from images. Another variant is a denoising autoencoder [5] that is trained to reconstruct the correct output representation from a corrupted input data point. Deep autoencoders have been extensively applied for dimensionality reduction and manifold learning with applications to different visual classification tasks [21, 57].

2.2.3 Convolutional Neural Networks (CNNs)

In this section, we discuss the different layers, the formation of different layer blocks in a Convolutional Neural Network (CNN) [23] and some state-of-the-art CNNs [28, 35, 68, 70, 88] for the ImageNet challenge. A CNN is similar to a regular neural network but is more specifically designed for images as input and uses a convolution operation instead of a matrix multiplication. The most common layers in a CNN are the input layer, the convolution layer, the rectified linear unit (ReLU) layer, the pooling layer, and a fully connected layer. The convolution layer computes the dot product between the weights and a small region connected to the input image, whereas the ReLU layer performs the elementwise activation using the function $f(y) = \max(0, y)$. The pooling layer is used to reduce the spatial dimensions as we go deeper into the CNN, and finally the fully connected layer computes the class score for every class label of the dataset. An example of a convolutional neural network architecture is shown in Fig. 2.5 that contains two convolutional and pooling layers followed by an ReLU layer and a fully connected layer.

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [60] is a challenging and popular image database having more than a million train images and 100,000 test images. Table 2.1 shows the performance of different CNNs on the ImageNet dataset for the ILSVRC challenge. The AlexNet was the first CNN that won the ILSVRC 2012 challenge and was a network with five convolutional

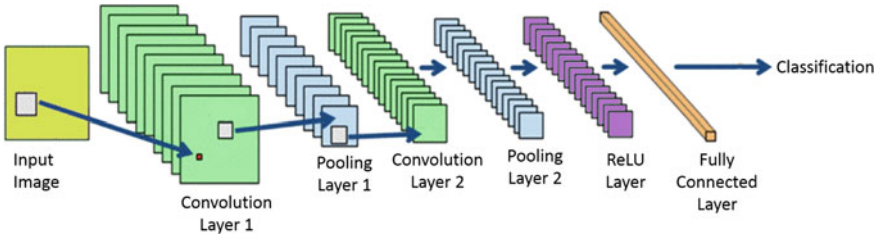

Fig. 2.5 An example of a convolutional neural network (CNN) architecture

Table 2.1 The top 5 error in the classification task using the ImageNet dataset

No.	CNN	Top 5 error (%)
1	AlexNet [35]	16.40
2	ZFNet [88]	11.70
3	VGG Net [68]	7.30
4	GoogLeNet [70]	6.70
5	ResNet [28]	3.57

layers, five max-pooling layers and three fully-connected layers. The ZFNet was then developed that improved the AlexNet by fine-tuning the architecture of the AlexNet. The ILSVRC 2014 challenge winner was the GoogLeNet which used an inception module to remove a large number of parameters from the network for improved efficiency. The current state-of-the-art CNN and the winner of ILSVRC 2015 is the ResNet which introduces the concept of residual net with skip connections and batch normalization with 152 layers in the architecture.

2.2.4 Deep Boltzmann Machine (DBM)

A Deep Boltzmann Machine (DBM) [22, 23, 61] is a type of generative model where the variables with each layer depend on each other conditioned on the neighbouring variables. A DBM is a probablistic graph model containing stacked layers of a Restricted Boltzmann Machine (RBM) where the connections between all the layers are undirected. For a DBM with a visible layer $\mathbf{v}$, three hidden layers $\mathbf{h}_1$, $\mathbf{h}_2$ and $\mathbf{h}_3$, weight matrices $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$, and the partition function Z, the joint probability is given as follows [61]:

$$P(\mathbf{v}) = \sum_{\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3} \frac{1}{Z} \exp[\mathbf{v}^T \mathbf{W}_1 \mathbf{h}_1 + \mathbf{h}_1{}^T \mathbf{W}_2 \mathbf{h}_2 + \mathbf{h}_2{}^T \mathbf{W}_3 \mathbf{h}_3]$$

The pre-training for a DBM must be initialized from stacked restricted Boltzmann machines or RBMs, and a discriminative fine tuning is performed using the error back propagation algorithm. A DBM derives a high level representation from the unlabeled data while the labeled data is only used to slightly fine-tune the data. Experimental results on several visual recognition datasets show that the DBM achieves better performance than some other learning methods [22, 61].

2.3 Support Vector Machines

Support Vector Machine (SVM) is one of the popular machine learning methods. The fundamental idea behind SVM is a novel statistical learning theory that was

proposed by Vapnik [77]. Unlike traditional methods such as Neural Networks which are based on the empirical risk minimization (ERM), SVM was based on the VC dimension and the structural risk minimization (SRM) [77]. Since its introduction, SVM has been applied to a number of applications ranging from text detection and categorization [32, 67], handwritten character and digit recognition [73], speech verification and recognition [48], face detection and recognition [72], to object detection and recognition [53].

Though SVM achieves better generalization performance compared with many other machine learning technologies, when solving large-scale and complicated problems, the learning process of SVM tends to define a complex decision model due to the increasing number of support vectors. As a result, SVM becomes less efficient due to the expensive computation cost, which involves the inner product computation for a linear SVM and the kernel computation for a kernel SVM for all the support vectors. Many new SVM approaches have been proposed to address the inefficiency problem (i.e. the large number of support vectors) of the conventional SVM [8, 11, 15, 36, 38, 54, 58, 65]. We will discuss these approaches in Sect. 2.3.4 and present a new efficient Support Vector Machine (eSVM) in Sect. 2.3.5 [12–14].

2.3.1 Linear Support Vector Machine

To introduce the linear SVM we will start with outlining the application of SVM to the simplest case of binary classification that is linearly separable. Let the training set be $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l)\}$, where $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$ indicate the two different classes, and l is the number of the training samples. An n dimensional hyperplane that can completely separate the samples may be defined as:

$$\mathbf{w}^t\mathbf{x} + b = 0 \tag{2.1}$$

This hyperplane is defined such that $\mathbf{w}^t\mathbf{x} + b \geq +1$ for the positive samples and $\mathbf{w}^t\mathbf{x} + b \leq -1$ for the negative samples. As mentioned above, SVM searches the optimal separating hyperplane by maximizing the geometric margin. This margin can be maximized as follows:

$$\begin{aligned} &\min_{\mathbf{w},b} \tfrac{1}{2}\mathbf{w}^t\mathbf{w}, \\ &subject\ to\ \ y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, l. \end{aligned} \tag{2.2}$$

The Lagrangian theory and the Kuhn-Tucker theory are then applied to optimize the functional in Eq. 2.2 with inequality constraints [78]. The optimization process leads to the following quadratic convex programming problem:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \\ &subject\ to \ \sum_{i=1}^{l} y_i \alpha_i = 0, \\ &\alpha_i \geq 0,\ i = 1, 2, \ldots l \end{aligned} \tag{2.3}$$

From the Lagrangian theory and the Kuhn-Tucker theory, we also have:

$$\mathbf{w} = \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = \sum_{i \in SV} y_i \alpha_i \mathbf{x}_i \tag{2.4}$$

where SV is the set of Support Vectors (SVs), which are the training samples with nonzero coefficients α_i. The decision function of the SVM is therefore derived as follows:

$$f(x) = sign(\mathbf{w}\mathbf{x} + b) = sign(\sum_{i \in SV} y_i \alpha_i \mathbf{x}_i \mathbf{x} + b) \tag{2.5}$$

2.3.2 Soft-Margin Support Vector Machine

In applications with real data, the two classes are usually not completely linearly separable. The soft-margin SVM was then proposed with a tolerance of misclassification error [78]. The fundamental idea of the soft-margin SVM is to maximize the margin of the separating hyperplane while minimizing a quantity proportional to the misclassification errors. To do this, the soft-margin SVM introduces the slack variables $\xi_i \geq 0$ and a regularizing parameter $C > 0$. The soft-margin SVM is defined as follows:

$$\begin{aligned} &\min_{\mathbf{w}, b, \xi_i} \tfrac{1}{2} \mathbf{w}^t \mathbf{w} + C \sum_{i=1}^{l} \xi_i\ , \\ &subject\ to \ \ y_i (\mathbf{w}^t \mathbf{x}_i + b) \geq 1 - \xi_i\ , \\ &\xi_i \geq 0,\ \ i = 1, 2, \ldots, l. \end{aligned} \tag{2.6}$$

Using the Lagrangian theory, its quadratic convex programming program is defined as follows:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \mathbf{x}_j \\ &subject\ to \ \sum_{i=1}^{l} y_i \alpha_i = 0, \\ &0 \leq \alpha_i \leq C,\ i = 1, 2, \ldots l \end{aligned} \tag{2.7}$$

From Eq. 2.6 we can observe that the standard SVM is defined on the trade-off between the least number of misclassified samples ($\min_{\xi_i} C \sum_{i=1}^{l} \xi_i$) and the maximum

margin ($\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^t\mathbf{w}$) of the separating hyperplane. The decision function of the soft-margin SVM is the same with the linear SVM.

2.3.3 Non-linear Support Vector Machine

The linear SVM and the soft-margin SVM are generally not suitable for complex classification problems which are completely inseparable. Non-linear Support Vector Machine solves the non-linear classification by mapping the data from the input space into a high dimensional feature space using a non-linear transformation $\Phi : x_i \rightarrow \phi(x_i)$. Cover's theorem states that if the transformation is nonlinear and the dimensionality of the feature space is high enough, then the input space may be transformed into a new feature space where the data are linearly separable with high probability [77]. This nonlinear transformation is performed in an implicit way through kernel functions [77].

Specifically, the non-linear SVM is defined as follows:

$$\begin{gathered} \min_{\mathbf{w},b,\xi_i} \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_{i=1}^{l}\xi_i \ , \\ subject\ to\ \ y_i(\mathbf{w}^t\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \ , \\ \xi_i \geq 0, \ \ i = 1, 2, \ldots, l. \end{gathered} \tag{2.8}$$

Its corresponding quadratic convex programming program is as follows:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &subject\ to\ \ \sum_{i=1}^{l} y_i\alpha_i = 0, \\ &0 \leq \alpha_i \leq C, \ i = 1, 2, \ldots l \end{aligned} \tag{2.9}$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^t\phi(\mathbf{x}_j)$ is the kernel function. The decision function of the non-linear SVM is defined as follows:

$$f(x) = sign(\sum_{i \in SV} y_i\alpha_i K(\mathbf{x}_i, \mathbf{x}) + b) \tag{2.10}$$

Typically, there are three types of kernel functions, namely, the polynomial kernel functions, the Gaussian kernel functions, and the Sigmoid functions (though strictly speaking, the Sigmoid functions are not kernel functions).

1. Polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^t\mathbf{x}_j + 1)^d$
2. Gaussian: $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$
3. Sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = tanh(\beta_0\mathbf{x}_i^t\mathbf{x}_j + \beta_1)$

2.3.4 Simplified Support Vector Machines

Although SVM exhibits many theoretical and practical advantages like good generalization performance, the decision function of SVM involves a kernel computation with all Support Vectors (SVs) and thus leads to a slow testing speed. This situation becomes even worse when SVM is applied to large-scale and complicated problems like image search and video retrieval. The decision function becomes over complex due to the large number of SVs and the testing speed is extremely slow because of the expensive kernel computation cost. To address this problem, much research has been carried out to simplify the SVM model and some simplified SVMs have been proposed. In this section, we will first thoroughly analyze the structure and distribution of SVs in the traditional SVM as well as its impact on the computation cost and generalization performance. We will then review some representative simplified SVMs.

Previous research shows that the complexity of a classification model depends on the size of its parameters [1]. A simple model can generate a fast system but has poor accuracy. In contrast, a complex model can reach a higher classification accuracy on the training data but will lead to lower efficiency and poor generalization performance.

From Eq. 2.10, it is observed that the complexity of SVM model depends on the size of SVs, which are defined as a subset of training samples whose corresponding α_i is not equal to zero. According to the Karush-Kuhn-Tucker (KKT) conditions, the optimization problem of the standard SVM defined in Eq. 2.4 should satisfy the following equation:

$$\alpha_i[y_i(\omega^t\phi(x)+b)-1+\xi_i]=0,\ i=1,2,\ldots l \tag{2.11}$$

where $\alpha_i \neq 0$ when $y_i(\omega^t\phi(x)+b)-1+\xi_i=0$. Because of the flexibility of the parameter ξ_i, the probability that $y_i(\omega^t\phi(x)+b)-1+\xi_i=0$ holds is very high, and thus α_i is more likely to get a nonzero value. More specifically, in Eq. 2.11, SVs are those samples between and on the two separating hyperplanes $\omega^t\phi(x)+b=-1$ and $\omega^t\phi(x)+b=1$ (Fig. 2.6). For a complicated large-scale classification problem, since many misclassified samples exist between these two hyperplanes during training, the size of SVs will be very large and thus an over complex decision model will be generated.

As we mentioned above, the primary impact of an over complex model is on its computation efficiency. From Eq. 2.10, it is observed that the decision function of SVM involves a kernel computation with all SVs. Therefore, the computational complexity of SVM is proportional to the number of SVs. An over complex model contains a large number of SVs and thus its computational cost becomes very expensive. This will lead to a slower classification speed that restricts the application of SVM to real-time applications. Another potential harm of an over complex SVM model is to reduce its generalization performance. SVM is well known for its good generalization performance. Unlike the techniques such as the Artificial Neural

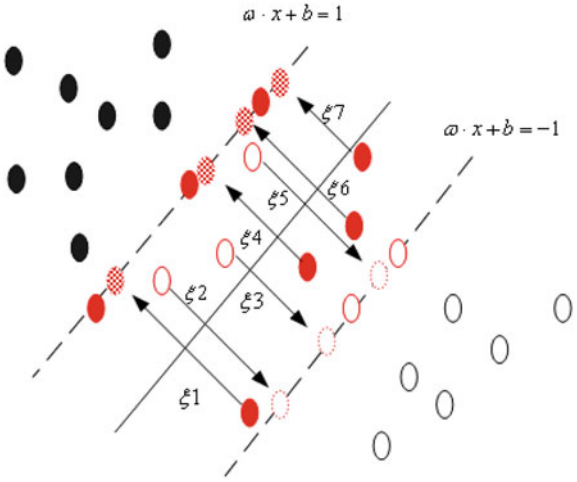

Fig. 2.6 SVM in 2D space (*Red circles* represent support vectors)

Networks (ANNs) that are based on the Empirical Risk Minimization (ERM) principle, SVM is based on the Structural Risk Minimization (SRM) principle [77]. the SRM principle empowers SVM with good generalization performance by keeping a balance between seeking the best classifier using the training data and avoiding overfitting during the learning process. However, an over complex model is likely to break this balance and increases the risk of overfitting and thus reduces its generalization performance.

Burges [8] proposed a method that computes an approximation to the decision function in terms of a reduced set of vectors to reduce the computational complexity of the decision function by a factor of ten. The method was then applied to handwritten digits recognition [65] and face detection [58]. However, this method not only reduces the classification accuracy but also slows down the training speed due to the higher computational cost for the optimal approximation. A Reduced Support Vector Machine (RSVM) was then proposed as an alternative to the standard SVM [36, 38]. A nonlinear kernel was generated based on a separating surface (decision function) by solving a smaller optimization problem using a subset of the training samples. The RSVM successfully reduced the model's complexity but it also decreased the classification rate. Furthermore, a new SVM, named υ-SVM, was proposed [15]. The relationship among the parameter υ, the number of support vectors, and the classification error was thoroughly discussed. However, this method would reduce the generalization performance when the parameter υ is too small. Other simplified support vector machine models are introduced in [11, 54]. To summarize, most of these simplified SVMs are able to reduce the computational cost but often at the expense of accuracy.

2.3.5 *Efficient Support Vector Machine*

We now introduce an efficient Support Vector Machine (eSVM), which significantly improves the computational efficiency of the traditional SVM without sacrificing its generalization performance [13]. The eSVM has been successfully applied to eye detection [14]. Motivated by the above analysis that it is the flexibility of the parameter ξ_i that leads to the large number of support vectors, the eSVM implements the second optimization of Eq. 2.8 as follows:

$$\begin{aligned} &\min_{\omega,b,\xi} \tfrac{1}{2}\omega^t\omega + C\xi \ , \\ &subject\ to\ \ y_i(\omega^t\phi(x_i)+b) \geq 1 \ , \ i \in V - MV \\ &\qquad y_i(\omega^t\phi(x_i)+b) \geq 1-\xi \ , \ i \in MV \ , \ \xi \geq 0 \end{aligned} \tag{2.12}$$

where M is the set of the misclassified samples in the traditional SVM and V is the set of all training samples. Its dual quadratic convex programming problem is:

$$\begin{aligned} &\max_{\alpha} \sum_{i\in V}\alpha_i - \tfrac{1}{2}\sum_{i,j\in V}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \\ &subject\ to\ \ \sum_{i\in V} y_i\alpha_i = 0, \ \left(\sum_{i\in MV}\alpha_i\right) \leq C, \\ &\qquad \alpha_i \geq 0, \ i \in V \end{aligned} \tag{2.13}$$

Note that instead of the flexibility of the slack variables in Eq. 2.8, we set these slack variables to a fixed value in Eq. 2.12. Now the new KKT conditions of Eq. 2.12 become:

$$\begin{aligned} &\alpha_i[y_i(\omega^t\phi(x)+b)-1] = 0, \ i \in V - MV \\ &\alpha_i[y_i(\omega^t\phi(x)+b)-1+\xi] = 0, \ i \in MV \end{aligned} \tag{2.14}$$

According to Eq. 2.14, $\alpha_i \neq 0$ when $y_i(\omega^t\phi(x)+b)-1=0$, $i \in V-MV$ or $y_i(\omega^t\phi(x)+b)-1+\xi=0$, $i \in MV$. The support vectors are those samples on the two separating hyperplanes $\omega^t\phi(x)+b=-1$ and $\omega^t\phi(x)+b=1$ and the misclassified samples farthest away from the hyperplanes (Fig. 2.7). As a result, the number of support vectors is much less than those defined by Eq. 2.11.

Compared with the traditional SVM, which is defined on the trade-off between the least number of the misclassified samples ($\min_{\xi_i} C\sum_{i=1}^{l}\xi_i$) and the maximum margin ($\min_{\omega,b}\frac{1}{2}\omega^t\omega$) of the two separating hyperplanes, the eSVM is defined on the trade-off between the minimum margin of the misclassified samples ($\min_{\xi} C\xi$) and the maximum margin ($\min_{\omega,b}\frac{1}{2}\omega^t\omega$) between the two separating hyperplanes. For complicated classification problems, the traditional SVM builds up a complex SVM model in pursuit of the least number of misclassified samples to some extend. According to SRT, it will increase the risk of overfitting on the training samples and thus

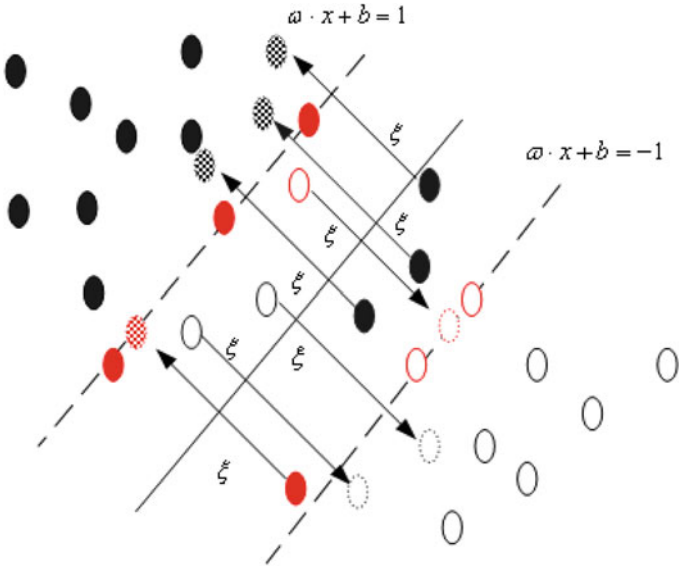

Fig. 2.7 eSVM in 2D space (*Red circles* represent support vectors)

reduces its generalization performance. The eSVM pursues the minimal margin of misclassified samples and its decision function is more concise. Therefore, the eSVM can be expected to achieve higher classification accuracy than the traditional SVM.

2.3.6 Applications of SVM

This section presents a survey of the applications of SVM to image search and video retrieval. Rapidly increasing use of smart phones and significantly reduced storage cost have resulted in the explosive growth of digital images and videos over the internet. As a result, image or video based search engines are in an urgent demand to find similar images or videos from a huge image or video database. Over the last decade many learning based image search and video retrieval techniques have been presented. Among them SVM as a powerful learning tool is widely used.

Chang and Tong [75] presented an image search method using a so called support vector machine active learning (SVM_{Active}). SVM_{Active} combines active learning with support vector machine. Intuitively, SVM_{Active} works by combining three ideas: (1) SVM_{Active} regards the task of learning a target concept as one of learning an SVM binary classifier. An SVM captures the query concept by separating the relevant images from irrelevant ones with a hyperplane in a projected space, usually a very high-dimensional one. The projected points on one side of the hyperplane are considered relevant to the query concept and the rest irrelevant; (2) SVM_{Active} learns

the classifier quickly via active learning. The active part of SVM_{Active} selects the most informative instances with which to train the SVM classifier. This step ensures fast convergence to the query concept in a small number of feedback rounds; (3) once the classifier is trained, SVM_{Active} returns the top-k most relevant images. These are the k images farthest from the hyperplane on the query concept side. SVM_{Active} needs at least one positive and one negative example to start its learning process. Two seeding methods were also presented: one by MEGA and one by keywords. To make both concept-learning and image retrieval efficient, a multi-resolution image-feature extractor and a high-dimensional indexer were also applied. Experiments ran on three real-world image data sets that were collected from Corel Image CDs and the internet (http://www.yestart.com/pic/). Those three data sets contain a four-category, a ten-category, and a fifteen-category image sets, respectively. Each category consists of 100–150 images. Experimental results show that SVM_{Active} achieves significantly higher search accuracy than the traditional query refinement schemes.

Guo et al. [25] presented a novel metric to rank the similarity for texture image search. This metric was named distance from boundary (DFB), in which the boundary is obtained by SVM. In conventional texture image retrieval, the Euclidean or the Mahalanobis distances between the images in the database and the query image are calculated and used for ranking. The smaller the distance, the more similar the pattern to the query. But this kind of metric has some limitations: (1) the retrieval results corresponding to different queries may be much different although they are visually similar; (2) the retrieval performance is sensitive to the sample topology; (3) the retrieval accuracy is low. The basic idea of the DFB is to learn a non-linear boundary that separates the similar images with the query image from the remaining ones. SVM is applied to learn this non-linear boundary due to its generalization performance. Compared with the traditional similarity measure based ranking, the DFB method has three advantages: (1) the retrieval performance is relatively insensitive to the sample distribution; (2) the same results can be obtained with respect to different (but visually similar) queries; (3) the retrieval accuracy is improved compared with traditional methods. Experiments on the Brodatz texture image database [7] with 112 texture classes show the effectiveness of the DFB method.

Recently the interactive learning mechanism was introduced to image search. The interactive learning involves a relevance feedback (RF) that is given by users to indicate which images they think are relevant to the query. Zhang et al. [89] presented an SVM based relevance feedback for image search. Specifically, during the process of relevance feedback, users can mark an image as either relevant or irrelevant. Given the top N_{RF} images in the result as the training data, a binary classifier can be learned using an SVM to properly represent the user's query. An SVM is chosen here due to its generalization performance. Using this binary classifier, other images can be classified into either the relevance class or the irrelevance class in terms of the distance from each image to the separating hyperplane. Obviously, in the first learning iteration, both marked relevant samples and unmarked irrelevant samples are all close to the query. Such samples are very suitable to construct the SVM classifier because support vectors are just those that lie on the separating margin while other samples far away from the hyperplane will contribute nothing to the classifier. In

the following iterations, more relevant samples fed back by users can be used to refine the classifier. Experiments were performed on a database that consists of 9,918 images from the Corel Photo CD (http://www.yestart.com/pic/). Five iterations were carried out to refine the SVM classifier, with each iteration allowing users to mark top 100 ($N_{RF} = 100$) images as feedback. A Gaussian kernel was chosen for the SVM. Experimental results show that both the recall rate and the precision rate are improved as the learning iteration progresses, and finally it reaches a satisfactory performance.

Hong et al. [31] presented a method that utilized SVM to update the preference weights (through the RF) that are used to evaluate the similarity between the relevant images. The similarity between two images – the query image and the searched image—is defined by summing the distances of individual features with fixed preference weights. The weights can be updated through the RF to reach better search performance. In [31], SVM was applied to perform non-linear classification on the positive and negative feedbacks through the RF. The SVM learning results are then utilized to automatically update the preference weights. Specifically, once the SVM separating hyperplane was trained, the distance between a feedback sample and the separating hyperplane indicates that how much this sample belonging to the assigned class is differentiated from the non-assigned one. In other words, the farther the positive sample feedbacks from the hyperplane, the more distinguishable they are from the negative samples. Therefore, those samples should be assigned a larger weight compared with other samples. In [31], the preference weight is set linearly proportional to the distance between the sample and the separating hyperplane. Experiments were performed on the COREL dataset (http://www.yestart.com/pic/), which contains 17,000 images. A polynomial kernel function with $d = 1$ was used for the SVM learning. The preference weights were normalized to the range [10, 100]. Experimental results show improved accuracy over other RF based methods.

Li et al. [37] presented a multitraining support vector machine (MTSVM) to further improve the SVM based RF for image retrieval. The MTSVM is based on the observation that (1) the success of the co-training model augments the labeled examples with unlabeled examples in information retrieval; (2) the advances in the random subspace method overcomes the small sample size problem. With the incorporation of the SVM and the multitraining model, the unlabeled examples can generate new informative training examples for which the predicted labels become more accurate. Therefore, the MTSVM method can work well in practical situations. In the MTSVM learning model, the majority voting rule (MVR) [34] was chosen as the similarity measure in combining individual classifiers since every single classifier has its own distinctive ability to classify relevant and irrelevant samples. Experiments were carried out upon a subset of images from the Corel Photo Gallery (http://www.yestart.com/pic/). This subset consists of about 20,000 images of very diverse subject matters for which each image was manually labeled with one of the 90 concepts. Initially, 500 queries were randomly selected, and the program autonomously performs a RF with the top five most relevant images (i.e., images with the same concept as

the query) marked as positive feedback samples within the top 40 images. The five negative feedback samples are marked in a similar fashion. The procedure is chosen to replicate a common working situation where a user would not label many images for each feedback iteration. Experimental results shown that MTSVM consistently improved the performance over conventional SVM-based RFs in terms of precision and standard deviation.

With the observation of the success application of RF and SVM to image retrieval, Yazdi et al. [87] applied RF and SVM to video retrieval. The proposed method consists of two major steps: key frame extraction and video shot retrieval. A new frame extraction method was presented using a hierarchical approach based on clustering. Using this method, the most representative key frame was then selected for each video shot. The video retrieval was based on an SVM based RF that was capable of combining both low-level features and high-level concepts. The low-level features are the visual image features such as color and texture, while the high-level concepts are the user's feedback through RF. The video database was finally classified into groups of relevant and irrelevant using this SVM classifier. The proposed method was validated on a video database with 800 shots from Trecvid2001 (http://www.open-video.org) and home videos. The video shots database includes airplanes, jungles, rivers, mountains, wild life, basketball, roads, etc. A total of 100 random queries were selected and judgements on the relevance of each shot to each query shot were evaluated. Different kernels for SVM-based learning in RF module were used. Experimental results show that SVM with the Gaussian function as kernel has better performance than the linear or polynomial kernel. The final experimental results show the improved performance after only a few RF iterations.

More applications of SVM to image search and video retrieval can be found in [10, 30, 52, 62, 71, 86].

2.4 Other Popular Kernel Methods and Similarity Measures

Kernel methods stress the nonlinear mapping from an input space to a high-dimensional feature space [18, 29, 63, 66]. The theoretical foundation for implementing such a nonlinear mapping is the Cover's theorem on the separability of patterns: "A complex pattern-classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space" [26]. Support Vector Machine (SVM) [18, 78], which defines an optimal hyperplane with maximal margin between the patterns of two classes in the feature space mapped nonlinearly from the input space, is a kernel method. Kernel methods have been shown more effective than the linear methods for image classification [3, 16, 50, 64]. Being

linear in the feature space, but nonlinear in the input space, kernel methods thus are capable of encoding the nonlinear interactions among patterns. Representative kernel methods, such as kernel PCA [64] and kernel FLD [3, 16, 50], overcome many limitations of the corresponding linear methods by nonlinearly mapping the input space to a high-dimensional feature space. Scholkopf et al. [64] showed that kernel PCA outperforms PCA using an adequate non-linear representation of input data. Yang et al. [85] compared face recognition performance using kernel PCA and the Eigenfaces method. The empirical results showed that the kernel PCA method with a cubic polynomial kernel achieved the lowest error rate. Moghaddam [51] demonstrated that kernel PCA with Gaussian kernels performed better than PCA for face recognition. Some representative kernel methods include kernel discriminant analysis [56, 84], kernel-based LDA [47], localized kernel eigenspaces [24], sparse kernel feature extraction [19], and multiple kernel learning algorithm [79, 80, 82].

Further research shows that new kernel methods with unconventional kernel models are able to improve pattern recognition performance [40]. One such kernel method is the multi-class Kernel Fisher Analysis (KFA) method [40]. The KFA method extends the two-class kernel Fisher methods [16, 50] by addressing multi-class pattern recognition problems, and it improves upon the traditional Generalized Discriminant Analysis (GDA) method [3] by deriving a unique solution. As no theoretical guideline is available in choosing a right kernel function for a particular application and the flexibility of kernel functions is restricted by the Mercer's conditions, one should investigate new kernel functions and new kernel models for improving the discriminatory power of kernel methods. The fractional power polynomial models have been shown to be able to improve image classification performance when integrated with new kernel methods [39, 40]. A fractional power polynomial, however, does not necessarily define a kernel function, as it might not define a positive semi-definite Gram matrix. Hence, a fractional power polynomial is called a model rather than a kernel function.

Similarity measures play an essential role in determining the performance of different learning and recognition methods [9, 43, 44, 46, 74]. Some image classification methods, such as the Eigenfaces method [33, 76], often apply the whitened cosine similarity measure for achieving good classification performance [6, 55]. Other methods, such as the Fisherfaces method [4, 20, 69], however, often utilize the cosine similarity measure for improving image classification performance [45, 46]. Further research reveals why the whitened cosine similarity measure achieves good image classification performance for the Principal Component Analysis (PCA) based methods [41, 42]. In addition, new similarity measures, such as the PRM Whitened Cosine (PWC) similarity measure and the Within-class Whitened Cosine (WWC) similarity measure, for further improving image classification performance have been presented [41]. The reason why the cosine similarity measure boosts the image classification performance for the discriminant analysis based methods has been discovered due to its theoretical roots in the Bayes decision rule for minimum error [44]. Furthermore, some inherent challenges of the cosine similarity, such as

its inadequacy in addressing distance and angular measures, have been investigated. And finally a new similarity measure, which overcomes the inherent challenges by integrating the absolute value of the angular measure and the l_p norm of the distance measure, is presented for further enhancing image classification performance [44].

2.5 Conclusion

We have reviewed in this chapter some representative learning and recognition methods that have broad applications in intelligent image search and video retrieval. In particular, we first discuss some popular deep learning methods, such as the feed-forward deep neural networks [35, 49], the deep autoencoders [2, 81], the convolutional neural networks [23], and the Deep Boltzmann Machine (DBM) [22, 61]. We then discuss one of the popular machine learning methods, namely, Support Vector Machine (SVM) [77]. Specifically we review the linear support vector machine [78], the soft-margin support vector machine [78], the non-linear support vector machine [77], the simplified support vector machine [11, 54], the efficient Support Vector Machine (eSVM) [13, 14], and the applications of SVM to image search and video retrieval [25, 37, 52, 62, 75, 87]. We finally briefly review some other popular kernel methods and new similarity measures [40, 41, 44].

References

1. Alpaydin, E.: Introduction to machine learning. The MIT Press, Cambridge (2010)
2. Baldi, P.: Autoencoders, unsupervised learning, and deep architectures. J. Mach. Learn. Res. (Proceedings of ICML unsupervised and transfer learning) **27**, 37–50 (2011)
3. Baudat, G., Anouar, F.: Generalized discriminant analysis using a kernel approach. Neural Comput. **12**(10), 2385–2404 (2000)
4. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
5. Bengio, Y., Yao, L., Alain, G., Vincent, P.: Generalized denoising auto-encoders as generative models. Adv, Neural Inf. Process. Syst. **26**, 899–907 (2013)
6. Beveridge, J., Givens, G., Phillips, P., Draper, B.: Factors that influence algorithm performance in the face recognition grand challenge. Comput. Vis. Image Underst. **113**(6), 750–762 (2009)
7. Brodatz, P.: Textures: a photographic album for artists and designers. Dover, New York (1966)
8. Burges, C.: Simplified support vector decision rule. In: Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96), Bari, Italy, July 3–6, 1996 (1996)
9. Chambon, S., Crouzil, A.: Similarity measures for image matching despite occlusions in stereo vision. Pattern Recognit. **44**(9), 2063–2075 (2011)
10. Chapelle, O., Haffner, P., Vapnik, V.N.: Support vector machines for histogram-based image classification. IEEE Trans. Neural Netw. **10**(5), 1055–1064 (1999)
11. Chen, J., Chen, C.: Reducing SVM classification time using multiple mirror classifers. IEEE Trans. Syst. Man Cybern. **34**(2), 1173–1183 (2004)
12. Chen, S., Liu, C.: Eye detection using color information and a new efficient SVM. In: IEEE Fourth International Conference on Biometrics: Theory, Applications, and Systems (BATS'10), Washington DC, USA (2010)

13. Chen, S., Liu, C.: A new efficient SVM and its application to a real-time accurate eye localization system. In: International Joint Conference on Neural Networks, San Jose, California, USA (2011)
14. Chen, S., Liu, C.: Eye detection using discriminatory haar features and a new efficient SVM. Image Vis. Comput. **33**(c), 68–77 (2015)
15. Chen, P., Lin, C., Scholkopf, B.: A tutorial on υ-support vector machines. Appl. Stoch. Models Bus. Ind. **21**, 111–136 (2005)
16. Cooke, T.: Two variations on Fisher's linear discriminant for pattern recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**(2), 268–273 (2002)
17. Corso, J.J., Alahi, A., Grauman, K., Hager, G.D., Morency, L.P., Sawhney, H., Sheikh, Y.: Video Analysis for Bodyworn Cameras in Law Enforcement. The Computing Community Consortium whitepaper (2015)
18. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
19. Dhanjal, C., Gunn, S., Shawe-Taylor, J.: Efficient sparse kernel feature extraction based on partial least squares. IEEE Trans. Pattern Anal. Mach. Intell. **31**(8), 1347–1361 (2009)
20. Etemad, K., Chellappa, R.: Discriminant analysis for recognition of human face images. J. Opt. Soc. Am. A **14**, 1724–1733 (1997)
21. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Aistats, vol. 15, p. 275 (2011)
22. Goodfellow, I., Mirza, M., Courville, A., Bengio, Y.: Multi-prediction deep boltzmann machines. In: Advances in Neural Information Processing Systems, pp. 548–556 (2013)
23. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning (2016). URL http://www.deeplearningbook.org (Book in preparation for MIT Press)
24. Gundimada, S., Asari, V.: Facial recognition using multisensor images based on localized kernel eigen spaces. IEEE Trans. Image Process. **18**(6), 1314–1325 (2009)
25. Guo, G., Zhang, H.J., Li, S.Z.: Distance-from-boundary as a metric for texture image retrieval. In: IEEE International Conference on Acoustics. Speech, and Signal Processing, vol. 3, pp. 1629–1632, Washington DC, USA (2001)
26. Haykin, S.: Neural Networks — A Comprehensive Foundation. Macmillan College Publishing Company, Inc., New York (1994)
27. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034 (2015)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
29. Herbrich, R.: Learning Kernel Classifiers: Theory and Algorithms. MIT Press, Cambridge (2002)
30. Hoi, C.H., Chan, C.H., Huang, K., Lyu, M.R., King, I.: Biased support vector machine for relevance feedback in image retrieval. In: 2004 IEEE International Joint Conference on Neural Networks, vol. 4, pp. 3189–3194 (2004)
31. Hong, P., Tian, Q., Huang, T.S.: Incorporate support vector machines to content-based image retrieval with relevance feedback. In: 2000 International Conference on Image Processing, vol. 3, pp. 750–753 (2000)
32. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: 10th European Conference on Machine Learning (1999)
33. Kirby, M., Sirovich, L.: Application of the Karhunen-Loeve procedure for the characterization of human faces. IEEE Trans. Pattern Anal. Mach. Intell. **12**(1), 103–108 (1990)
34. Kittler, J., Hatef, M., Duin, R., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**(3), 226–239 (1998)
35. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012)

36. Lee, Y., Mangasarian, O.: RSVM: Reduced support vector machines. In: The First SIAM International Conference on Data Mining (2001)
37. Li, J., Allinson, N., Tao, D., Li, X.: Multitraining support vector machine for image retrieval. IEEE Trans. Image Process. **15**(11), 3597–3601 (2006)
38. Lin, K., Lin, C.: A study on reduced support vector machine. IEEE Trans. Neural Netw. **14**(6), 1449–1559 (2003)
39. Liu, C.: Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **26**(5), 572–581 (2004)
40. Liu, C.: Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. IEEE Trans. Pattern Anal. Mach. Intell. **28**(5), 725–737 (2006)
41. Liu, C.: The Bayes decision rule induced similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1086–1090 (2007)
42. Liu, C.: Clarification of assumptions in the relationship between the bayes decision rule and the whitened cosine similarity measure. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 1116–1117 (2008)
43. Liu, C.: Effective use of color information for large scale face verification. Neurocomputing **101**, 43–51 (2013)
44. Liu, C.: Discriminant analysis and similarity measure. Pattern Recognit. **47**(1), 359–367 (2014)
45. Liu, Z., Liu, C.: Fusion of color, local spatial and global frequency information for face recognition. Pattern Recognit. **43**(8), 2882–2890 (2010)
46. Liu, C., Mago, V. (eds.): Cross Disciplinary Biometric Systems. Springer, New York (2012)
47. Liu, X., Chen, W., Yuen, P., Feng, G.: Learning kernel in kernel-based LDA for face recognition under illumination variations. IEEE Signal Process. Lett. **16**(12), 1019–1022 (2009)
48. Ma, C., Randolph, M., Drish, J.: A support vector machines-based rejection technique for speech recognition. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 381–384 (2001)
49. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of ICML, vol. 30 (2013)
50. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Mller, K.R.: Fisher discriminant analysis with kernels. In: Hu, Y.H., Larsen, J., Wilson, E., Douglas, S. (eds.) Neural Networks for Signal Processing IX, pp. 41–48. IEEE (1999)
51. Moghaddam, B.: Principal manifolds and probabilistic subspaces for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**(6), 780–788 (2002)
52. Nagaraja, G., Murthy, S.R., Deepak, T.: Content based video retrieval using support vector machine classification. In: 2015 IEEE International Conference on Applied and Theoretical Computing and Communication Technology, pp. 821–827 (2015)
53. Nakajima, C., Pontil, M., Poggio, T.: People recognition and pose estimation in image sequences. In: IEEE International Joint Conference on Neural Networks, vol. 4, pp. 189–194 (2000)
54. Nguyen, D., Ho, T.: An efficient method for simplifying support vector machines. In: International Conference on Machine Learning, Bonn, Germany (2005)
55. OToole, A.J., Phillips, P.J., Jiang, F., Ayyad, J., Penard, N., Abdi, H.: Face recognition algorithms surpass humans matching faces across changes in illumination. IEEE Trans. Pattern Anal. Mach. Intell. **29**(9), 1642–1646 (2007)
56. Pekalska, E., Haasdonk, B.: Kernel discriminant analysis for positive definite and indefinite kernels. IEEE Trans. Pattern Anal. Mach. Intell. **31**(6), 1017–1032 (2009)
57. Ranzato, M.A., Szummer, M.: Semi-supervised learning of compact document representations with deep networks. In: Proceedings of the 25th International Conference on Machine Learning, ICML '08, pp. 792–799 (2008)
58. Romdhani, S., Torr, B., Scholkopf, B., Blake, A.: Computationally efficient face detection. In: IEEE International Conference on Computer Vision (2001)
59. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)

60. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015). doi:10.1007/s11263-015-0816-y
61. Salakhutdinov, R., Hinton, G.: An efficient learning procedure for deep boltzmann machines. Neural Comput. **24**(8), 1967–2006 (2012)
62. Santhiya, G., Singaravelan, S.: Multi-SVM for enhancing image search. Int. J. Sci. Eng. Res. **4**(6) (2013)
63. Scholkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridege (2002)
64. Scholkopf, B., Smola, A., Muller, K.: Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. **10**, 1299–1319 (1998)
65. Scholkopf, B., Mika, S., Burges, C., Knirsch, P., Muller, K., Ratsch, G., Smola, A.: Input space versus feature space in kernel-based methods. IEEE Trans. Neural Netw. **10**(5), 1000–1017 (1999)
66. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
67. Shin, C., Kim, K., Park, M., Kim, H.: Support vector machine-based text detection in digital video. In: IEEE Workshop on Neural Networks for Signal Processing (2000)
68. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
69. Swets, D.L., Weng, J.: Using discriminant eigenfeatures for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **18**(8), 831–836 (1996)
70. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
71. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans. Pattern Anal. Mach. intell. **28**(7), 1088–1099 (2006)
72. Tefas, A., Kotropoulos, C., Pitas, I.: Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication. IEEE Trans. Pattern Anal. Mach. Intell. **23**(7), 735–746 (2001)
73. Teow, L., Loe, K.: Robust vision-based features and classification schemes for off-line handwritten digit recognition. Pattern Recognit. (2002)
74. Thung, K., Paramesran, R., Lim, C.: Content-based image quality metric using similarity measure of moment vectors. Pattern Recognit. **45**(6), 2193–2204 (2012)
75. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: the Ninth ACM International Conference on Multimedia, pp. 107–118 (2001)
76. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **13**(1), 71–86 (1991)
77. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York, NY (1995)
78. Vapnik, Y.N.: The Nature of Statistical Learning Theory, second edn. Springer, New York (2000)
79. Varma, M., Babu, B.: More generality in efficient multiple kernel learning. In: Proceedings of the International Conference on Machine Learning, Montreal, Canada (2009)
80. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: Proceedings of the International Conference on Computer Vision, Kyoto, Japan (2009)
81. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**, 3371–3408 (2010)
82. Wang, Z., Chen, S., Sun, T.: Multik-MHKS: a novel multiple kernel learning algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 348–353 (2008)
83. Wright, H.: Video Analysis for Body-worn Cameras in Law Enforcement (2015). http://www.cccblog.org/2015/08/06/video-analysis-for-body-worn-cameras-in-law-enforcement

84. Xie, C., Kumar, V.: Comparison of kernel class-dependence feature analysis (KCFA) with kernel discriminant analysis (KDA) for face recognition. In: Proceedings of IEEE on Biometrics: Theory, Application and Systems (2007)
85. Yang, M.H., Ahuja, N., Kriegman, D.: Face recognition using kernel Eigenfaces. In: Proc. IEEE International Conference on Image Processing, Vancouver, Canada (2000)
86. Yang, J., Yan, R., Hauptmann, A.G.: Cross-domain video concept detection using adaptive svms. In: 15th ACM International Conference on Multimedia, pp. 188–197 (2007)
87. Yazdi, H.S., Javidi, M., Pourreza, H.R.: SVM-based relevance feedback for semantic video retrieval. Int. J. Signal Imaging Syst. Eng. **2**(3), 99–108 (2009)
88. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer, New York (2014)
89. Zhang, L., Lin, F., Zhang, B.: Support vector machine learning for image retrieval. In: 2001 International Conference on Image Processing, vol. 2, pp. 721–724 (2001)

http://www.springer.com/978-3-319-52080-3

Recent Advances in Intelligent Image Search and Video Retrieval

Liu, C. (Ed.)
2017, XVII, 235 p. 88 illus., 85 illus. in color., Hardcover
ISBN: 978-3-319-52080-3