Video event detection using generalized subclass discriminant analysis and linear support vector machines

Nikolaos Gkalelis
Information Technologies Institute / CERTH
Thermi 57001, Greece
gkalelis@iti.gr

Vasileios Mezaris
Information Technologies Institute / CERTH
Thermi 57001, Greece
bmezaris@iti.gr

ABSTRACT

In this paper, a two-phase approach to event detection in video is proposed. This combines a novel nonlinear Discriminant Analysis (DA) method called Generalized Subclass DA (GSDA), to identify a discriminant subspace, and a Linear Support Vector Machine (LSVM), to efficiently learn the event in the derived subspace. The proposed GSDA-LSVM framework is used as an alternative to the Kernel Support Vector Machine (KSVM) approach, which despite its excellent classification accuracy requires significant computational resources for learning the events (i.e., for identifying the kernel parameters and KSVM penalty term) in large-scale video collections. In contrary, using the GSDA-LSVM approach the SVM penalty term can be rapidly identified in the lower dimensional subspace. Moreover, an additional speed up in deriving this lower-dimensional space is achieved by using the proposed GSDA method instead of conventional nonlinear subclass DA methods such as KSDA or KMSDA. This is made possible by GSDA exploiting the special structure of the inter-between-subclass scatter matrix to reformulate the original KSDA eigenvalue problem to one involving matrices of much smaller dimension. The proposed GSDA-LSVM approach leads to more accurate event detection and to computational efficiency gains, as shown by experimental results on the extensive TRECVID MED 2010 and 2012 datasets.

Categories and Subject Descriptors

H.3.1 [**INFORMATION STORAGE AND RETRIEVAL**]: Content Analysis and Indexing; I.5.2 [**ARTIFICIAL INTELLIGENCE**]: Vision and Scene Understanding—*Video analysis*; I.2.10 [**PATTERN RECOGNITION**]: Design Methodology—*Classifier design and evaluation*

General Terms

Algorithms, Theory, Performance, Experimentation.

ICMR'14 April 01 - 04 2014, Glasgow, United Kingdom

http://dx.doi.org/10.1145/2578726.2578745.

Keywords

High-level event detection, model vectors, kernel subclass discriminant analysis, support vector machine, speed up.

1. INTRODUCTION

The widespread use of video capture devices and video transmission and consumption applications in many areas, including entertainment, surveillance, the World Wide Web and the social Web, has caused a tremendous growth of video content. The automatic understanding, indexing and retrieval of this content in large-scale datasets is still a very challenging problem mainly for two reasons: a) the semantic gap between the generated video descriptions and the interpretation of the same video data by humans, b) the computational difficulties associated with processing these large-scale video collections.

Recent studies in neuroscience suggest that humans organize their memory using real-life experiences structured around high-level events. According to [23], high-level events are defined as "purposeful activities, involving people, acting on objects and interacting with each other to achieve some result". Inspired from these studies, researchers in the multimedia understanding community have started focusing on the detection of high-level events as a way to reduce the semantic gap. One major effort towards this direction is the annual multimedia event detection (MED) task initiated by NIST TRECVID in 2010 [25]. In this task a large-scale video dataset is provided for training and evaluating different event detection approaches.

To deal with the complexity and variability of video events, most event detection approaches extract a rich set of low-level features (visual and/or audio, static and/or dynamic, etc.) in order to generate an informative video content representation [13]. For each feature type a base event classifier is then created, and the different classifiers are combined utilizing an appropriate fusion scheme. In [29], motion features (STIP, DT) are extracted, Fisher vector (FV) encoding is applied to represent a video signal, and KSVMs with Gaussian radial basis function (RBF) kernel are used to build the event detectors. Experimental results in a subset of the MED 2012 dataset containing 25 events showed that FV-based representation provides superior performance in comparison to the Bag-of-Words-based (BoW) one. The same subset of MED 2012 is used in [24] to evaluate the algorithm proposed there, which exploits a compact set of low-level features (SIFT, MBH and MFCC), FV encoding, power normalization (which can be seen as explicit non-linear embedding) and linear SVMs (LSVMs). From their evaluation

the authors conclude that the MBH features provide rich video content information, and their combination with SIFT features (and to a smaller degree with the MFCC audio features) leads to significant performance improvements.

According to the event definition provided above, an event encompasses several other, less complex semantic entities, such as elementary actions, objects, etc. (called hereafter concepts). Based on this observation, another direction to event detection that several researchers investigate is exploiting a set of concept detectors to provide a more informative video representation, instead of trying to learn to detect the events by looking directly at the low-level features. For instance, in [17], videos are represented using the so-called model vectors [18, 9] (feature vectors describing detection results for the TRECVID semantic indexing (SIN) concepts), and a cross-entropy based criterion is applied to select the most informative concepts for the MED 2011 events. In [12], a subset of the MED 2012 dataset is used to study the effect of concept vocabulary properties in the performance of the event detectors, such as vocabulary size, diversity, and other. In [16], the authors propose a method that jointly learns the event detector and an intermediate semantic representation for the specific event.

Finally, low-level video features and model vector representations are often combined, aiming at better event detection performance [18, 22, 21]. In [18], model vectors consisting of 280 concept detector responses are combined with several low-level features (SIFT, STIP, etc.) and a hierarchical fusion strategy is used for detecting MED 2010 events. In this work it was shown that model vector-based event detectors outperformed classifiers trained on low-level features, and the combination of the two provided small but noticeable performance gains. The authors of [22] extract a very large variety of video features (SIFT, STIP, videotext, ASR transcripts, object spatial probability maps, etc.) and combine them using a multistage fusion strategy. In [21], the comparison of several different fusion strategies in the MED 2011 dataset for combining different video features (e.g., model vectors, color SIFT variants, MoSIFT, MFCC, etc.) showed that simple fusion strategies, such as arithmetic or geometric mean, can provide competitive performance to more complex ones. Finally, in the context of the MED 2013 challenge, MultiModal Pseudo Relevance Feedback (MMPRF) is proposed in [14] to leverage information across different feature modalities (ASR, model vectors, dense trajectories, SIFT, etc.), while [26] introduces a new feature (improved dense trajectories) which is shown to lead to very good detection performance.

Most of the event detection approaches proposed until now put the emphasis on how the video is represented and exploit new such techniques for improving the accuracy of the event detection system. On the machine learning front, for learning event detectors from these representations, standard machine learning methods (typically KSVMs) are employed. In contrary, in this paper we focus on classifier design for improving event detection in terms of both computational efficiency and accuracy. Other recent methods that focus on the machine learning part for improving event detection include [9, 11]. In [11], a linear feature extraction method is used to derive a discriminant subspace and the median Hausdorff distance is applied in this subspace for event detection. To handle data nonlinearities more effectively, in [9] a SRECOC framework is presented, combining multiple KSVM-based classifiers trained at different regions of the feature space. It has been shown that such fusion schemes based on KSVM detectors are among the most effective pattern classifiers. However, the direct exploitation in SVMs of feature vectors, which usually contain noise or irrelevant components, can degrade the classification performance. Moreover, KSVM-based techniques are very difficult to scale during training in big data problems. For instance, in the TRECVID MED 2012 challenge [25] the reported learning times typically range from a few days to several weeks, depending on the employed computational resources, i.e., the use of supercomputers or small-sized clusters, respectively. Therefore, more efficient computational approaches are necessary for the practical use of event detection in time-critical applications (e.g. interactive event learning and detection, event-based video surveillance, etc.).

Motivated from the above discussion, we propose in this paper the use of a nonlinear discriminant analysis (DA) algorithm [3, 19] to derive a lower dimensional embedding of the original data, and then use fast LSVMs in the resulting subspace to learn the events. DA methods have shown promising performance in several application domains such as face recognition [34], animation production [37, 36], etc. One of the major advantages of the proposed approach is that the optimization of the LSVM penalty term can be rapidly performed in the lower dimensional subspace, saving significant computational time in comparison to directly using SVMs in the original feature space. For realizing dimensionality reduction we utilize kernel subclass-based methods, which have been shown to outperform other DA approaches [6, 35, 10]. In particular, a new method called generalized subclass DA (GSDA) is proposed which exploits the special structure of the inter-between-subclass scatter [3, 4] to provide an efficient solution to the KSDA eigenvalue problem [6, 35]. The evaluation of the proposed GSDA-LSVM framework is performed in the TRECVID MED 2010 and 2012 video collections for the detection of 28 events. The experimental results show that the proposed approach outperforms KSVM in both efficiency and accuracy. In summary, the contributions of the paper are:

- The introduction of a new DA method, GSDA, that exploits the structure of the inter-between-subclass scatter matrix to efficiently compute a nonlinear data embedding.

- The presentation of a new event detection method that combines GSDA and LSVM, outperforming conventional KSVM in terms of both accuracy and efficiency. To the best of our knowledge, the combination of any DA method with LSVM has not yet been examined in the field of event detection.

The paper is structured as follows. In Section 2, the proposed method is described; specifically, the event detection problem is formulated, the model vector approach for representing video signals is outlined, and the two phase GSDA-LSVM pattern classifier is described. The proposed GSDA method is presented in detail in Section 3, while experimental evaluation results in the TRECVID MED corpus are discussed in Section 4. Finally, conclusions and future work are considered in Section 5.

2. EVENT DETECTION USING DISCRIMINANT ANALYSIS AND LSVM

2.1 Problem formulation

Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$ be an event-annotated set of feature vectors (derived from a corresponding set of videos), where the block matrix $\mathbf{X}_i = [\mathbf{x}_i^1, \ldots, \mathbf{x}_i^{N_i}]$ contains the N_i feature vectors of the target event class ($i = 1$) or the "rest of the world" class ($i = 2$), and $\mathbf{x}_i^n \in \mathbb{R}^F$ is the feature vector representation of the n-th video in the i-th class. Our goal is given $\mathbf{X}$ to learn an event detector $h : \mathbb{R}^F \to [0, 1]$ providing a likelihood value regarding the presence of the target event in the video.

2.2 Video representation

Model vectors are adopted in this work as feature vectors for the purpose of event detection. A model vector representation of videos is created, similarly to [9, 18], in three steps: a) low-level feature extraction, b) evaluation of a set of external concept detectors at keyframe level, and, c) a pooling strategy to retrieve a single model vector at video level. Specifically, a video signal is represented with a sequence of keyframes extracted at uniform time intervals, and a feature extraction procedure is applied to derive a low-level feature vector representation of each keyframe. This includes a point sampling strategy (e.g., Harris-Laplace, dense sampling), the extraction of local feature descriptors (e.g., SIFT, color SIFT variants), and a coding technique (e.g., BoW with hard/soft assignment, Fisher vectors; pyramidal decomposition) to represent each keyframe with a fixed dimensional feature vector $\mathbf{s} \in \mathbb{R}^S$. By applying the above feature extraction procedure and using a pool of F external concept detectors, $\mathcal{G} = \{x^f(\mathbf{s}) : \mathbb{R}^S \to [0, 1] | f = 1, \ldots, F\}$, the model vector $\mathbf{x}_i^{n,t} = [x_i^{n,t,1}, \ldots, x_i^{n,t,F}]^T$, corresponding to the t-th keyframe of the n-th video in class i is formed; that is, the element $x_i^{n,t,f}$ is the response of the f-th concept detector expressing the DoC that the respective concept is depicted in the keyframe. The concept detectors in $\mathcal{G}$ are created by exploiting an external dataset of videos or images annotated at concept level (e.g., TRECVID SIN [25], ImageNet LSVRC [1], etc.), the adopted feature extraction procedure, and an LSVM pattern classifier. Finally, average pooling along the keyframes is performed to retrieve the model vector $\mathbf{x}_i^n \in [0, 1]^F$ at video level, i.e., $\mathbf{x}_i^n = (T_i^n)^{-1} \sum_{t=1}^{T_i^n} \mathbf{x}_i^{n,t}$, where T_i^n is the number of keyframes of the n-th video in class i.

2.3 Event detection

In order to build an event detector, the derived model vectors in the columns of matrix $\mathbf{X}$ are initially used as input to a DA algorithm. DA algorithms compute a transformation matrix $\mathbf{\Psi} \in \mathbb{R}^{D \times F}$, $D \ll F$ for mapping the F-dimensional observation $\mathbf{x}_i^n$ to a vector $\mathbf{z}_i^n$ in the D-dimensional discriminant subspace by $\mathbf{z}_i^n = \mathbf{\Psi}^T \mathbf{x}_i^n$. The transformation matrix is identified by solving a generalized eigenvalue problem of the form $\grave{\mathbf{S}}\mathbf{\Psi} = \mathbf{S}\mathbf{\Psi}\mathbf{\Delta}$, where the matrices $\mathbf{S}$ and $\grave{\mathbf{S}}$ express the within- and between-class scatter respectively, and $\mathbf{\Delta}$ is the diagonal eigenvalue matrix. Several choices for the above matrices have been proposed in the literature exploiting different properties of the data [8, 35, 10]. In our case, a new DA algorithm, GSDA, is used. The theory behind GSDA is developed in Section 3.

Following dimensionality reduction, an LSVM classifier is used for learning an event detector h in the discriminant subspace. Given the lower dimensional embedding of the training set, $\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2]$, $\mathbf{Z}_i = [\mathbf{z}_i^1, \ldots, \mathbf{z}_i^{N_i}]$, $\mathbf{z}_i^n \in \mathbb{R}^D$, the LSVM optimization problem is defined as

$$\min_{\mathbf{g}, b, \xi_i^n} \frac{1}{2}\|\mathbf{g}\|^2 + C \sum_{i=1}^{2} \sum_{n=1}^{N_i} \xi_i^n, \tag{1}$$

subject to the constraints

$$y_i^n(\mathbf{g}^T \mathbf{z}_i^n + b) \geq 1 - \xi_i^n, \ \forall i, n \tag{2}$$

$$\xi_i^n \geq 0, \ \forall i, n, \tag{3}$$

where $\mathbf{g} \in \mathbb{R}^D, b \in \mathbb{R}$ are the weight vector and bias, respectively, defining the separating hyperplane between the two classes, $C > 0$ is the penalty term, and ξ_i^n and y_i^n is the slack variable and class label corresponding to $\mathbf{x}_i^n$ ($y_i^n = 3 - 2i$). The above problem (3) is usually reformulated to its dual quadratic form using standard Lagrangian methods, and solved using an appropriate optimization technique. The decision function that classifies a test observation $\mathbf{z}^t$ is then given by

$$h(\mathbf{z}^t) = \text{sign}(\mathbf{g}^T \mathbf{z}^t + b) \tag{4}$$

In addition to the binary classification decisions, class likelihoods, which are very useful for event-based retrieval applications and for the evaluation of event detection accuracy using measures such as average precision, are derived using an appropriate sigmoid function that maps SVM outputs to probabilities [15, 7].

3. GENERALIZED SUBCLASS DISCRIMINANT ANALYSIS

Subclass kernel DA methods have exhibited excellent generalization performance in a variety of real-world applications [6, 35, 10]. However, their high computational cost required during training hinders their use in applications involving large-scale datasets. In this section, a new algorithm called GSDA is presented, reformulating the KSDA criterion so that a significant speed-up can be achieved.

3.1 Fundamentals of kernel subclass DA

A basic assumption of conventional linear DA (LDA) is that class distributions are Guassian homoscedastic, which is rarely true in practice. Kernel DA (KDA) approaches [3, 19] exploit a nonlinear feature transformation $\boldsymbol{\phi}(\cdot) : \mathbb{R}^F \mapsto \mathcal{F}$ to map the data into a high (or even infinitely) dimensional space $\mathcal{F}$ where classes are expected to be linearly separable. However, such a mapping may be difficult or very expensive (in terms of computations) to identify. To this end, subclass extensions of KDA approaches [6, 35, 10] have been proposed imposing a less strict requirement for the feature mapping, i.e., the identification of a new feature space where subclasses belonging to different classes are linearly separable[1]. These methods exploit a subclass partition of the training

[1]The number of classes is denoted with Ω in the sequel. In our application, where a detector is learned separately for each event of interest, $\Omega = 2$ (the event of interest and the rest-of-the-world classes). However, GSDA is also applicable to multiclass problems. Thus, when proposing GSDA, the mathematical formulation for arbitrary number of classes is given.

data $\mathbf{X} = [\mathbf{X}_{1,1}, \ldots, \mathbf{X}_{\Omega,H_\Omega}]$, where $\mathbf{X}_{i,j} = [\mathbf{x}_{i,j}^1, \ldots, \mathbf{x}_{i,j}^{N_{i,j}}]$ contains the observations of the (i,j) subclass, to optimize the following criterion

$$\underset{\mathbf{\Psi}}{\operatorname{argmax}} \operatorname{tr}((\mathbf{\Psi}^T \mathbf{S}_{bsb}^\phi \mathbf{\Psi})^{-1}(\mathbf{\Psi}^T \mathbf{S}^\phi \mathbf{\Psi}))\,, \tag{5}$$

where, tr() is the matrix trace operator,

$$\mathbf{S}_{bsb}^\phi = \sum_{i=1}^{\Omega-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{\Omega}\sum_{l=1}^{H_k} \hat{p}_{i,j}\hat{p}_{k,l}(\hat{\boldsymbol{\mu}}_{i,j}^\phi - \hat{\boldsymbol{\mu}}_{k,l}^\phi)(\hat{\boldsymbol{\mu}}_{i,j}^\phi - \hat{\boldsymbol{\mu}}_{k,l}^\phi)^T, \tag{6}$$

is the inter-between-subclass scatter matrix,

$$\hat{\boldsymbol{\mu}}_{i,j}^\phi = \frac{1}{N_{i,j}}\sum_{n=1}^{N_{i,j}} \boldsymbol{\phi}_{i,j}^n \tag{7}$$

$$\hat{\boldsymbol{\mu}}^\phi = \sum_{i=1}^{\Omega}\sum_{j=1}^{H_i} \hat{p}_{i,j}\hat{\boldsymbol{\mu}}_{i,j}^\phi, \tag{8}$$

are the sample mean of (i,j) subclass and total sample mean in $\mathcal{F}$; $\hat{p}_i = N_i/N$, $\hat{p}_{i,j} = N_{i,j}/N$ are the estimated prior of class i and its j-th subclass respectively, and $\boldsymbol{\phi}_{i,j}^n = \boldsymbol{\phi}(\mathbf{x}_{i,j}^n)$ is the mapping of observation $\mathbf{x}_{i,j}^n$ in $\mathcal{F}$. Different scatter matrices have been used for representing $\mathbf{S}^\phi$ in (5) [8, 35, 10], such as the within-subclass scatter matrix $\mathbf{S}_{ws}^\phi = (1/N)\sum_{i=1}^{\Omega}\sum_{j=1}^{H_i}\sum_{n=1}^{N_{i,j}}(\boldsymbol{\phi}(\mathbf{x}_{i,j}^n) - \hat{\boldsymbol{\mu}}_{i,j}^\phi)(\boldsymbol{\phi}(\mathbf{x}_{i,j}^n) - \hat{\boldsymbol{\mu}}_{i,j}^\phi)^T$, the modified mixture scatter matrix $\check{\mathbf{S}}_m^\phi = \mathbf{S}_{bsb}^\phi + \mathbf{S}_{ws}^\phi$ or the mixture scatter matrix

$$\mathbf{S}_m^\phi = \frac{1}{N}\sum_{i=1}^{\Omega}\sum_{j=1}^{H_i}\sum_{n=1}^{N_{i,j}}(\boldsymbol{\phi}(\mathbf{x}_{i,j}^n) - \hat{\boldsymbol{\mu}}^\phi)(\boldsymbol{\phi}(\mathbf{x}_{i,j}^n) - \hat{\boldsymbol{\mu}}^\phi)^T \tag{9}$$

In the next section, we focus on the KSDA criterion [35], i.e., the case where $\mathbf{S}^\phi$ in (5) is replaced by $\mathbf{S}_m^\phi$ presented above. In particular, we show how the the special structure of $\mathbf{S}_{bsb}^\phi$ (6) can be exploited to speed-up the computation of the KSDA transformation matrix in $\mathcal{F}$.

3.2 Efficient computation of the transformation matrix

To avoid working in $\mathcal{F}$, which may have very high or even infinite dimensionality, the optimization criterion is reformulated in $\mathcal{F}$ using dot products and an appropriate Mercer kernel $k_{i,j.k,l}^{n,\nu} = k(\mathbf{x}_{i,j}^n, \mathbf{x}_{k,l}^\nu) = (\boldsymbol{\phi}_{i,j}^n)^T\boldsymbol{\phi}_{k,l}^\nu$ (e.g., Gaussian RBF, polynomial). Moreover, without loss of generality we assume that the total mean is zero ($\hat{\boldsymbol{\mu}}^\phi = \mathbf{0}$) [3]. To this end, the scatter matrices are reformulated as [6, 10]

$$\mathbf{S}_{bsb}^\phi = \mathbf{\Phi}\mathbf{A}\mathbf{\Phi}^T \tag{10}$$

$$\mathbf{S}_m^\phi = \mathbf{\Phi}\mathbf{\Phi}^T \tag{11}$$

where $\mathbf{\Phi} = [\mathbf{\Phi}_{1,1}, \ldots, \mathbf{\Phi}_{\Omega,H_\Omega}]$, $\mathbf{\Phi}_{i,j} = [\boldsymbol{\phi}_{i,j}^1, \ldots, \boldsymbol{\phi}_{i,j}^{N_{i,j}}]$ and

$$\mathbf{A}_{i,j.k,l} = \begin{cases} \tilde{p}_{i,j}(1-\tilde{p}_i)/(N_{i,j}N_{k,l}), & \text{if } (i,j) = (k,l), \\ 0 & \text{if } i = k, j \neq l, \\ -\tilde{p}_{i,j}\tilde{p}_{k,l}/(N_{i,j}N_{k,l}), & \text{else.} \end{cases} \tag{12}$$

According to the theory of reproducing kernels the column vectors of $\mathbf{\Psi}$ must lie in the span of all training samples in $\mathcal{F}$ [19], and therefore we can express it as

$$\mathbf{\Psi} = \mathbf{\Phi}\mathbf{\Gamma} \tag{13}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{N\times H-1}$ contains the expansion coefficients. Substituting (10), (11), (13) in (5) we get

$$\underset{\mathbf{\Gamma}}{\operatorname{argmax}} \operatorname{tr}((\mathbf{\Gamma}^T\mathbf{K}\mathbf{A}\mathbf{K}\mathbf{\Gamma})^{-1}(\mathbf{\Gamma}^T\mathbf{K}\mathbf{K}\mathbf{\Gamma}))\,, \tag{14}$$

where $\mathbf{K}$ is the kernel Gram matrix. The above problem is ill-posed as we estimate an N-dimensional covariance matrix from N observations. There are mainly two approaches to overcome this problem, i.e., either regularizing matrix $\mathbf{K}$ [19] and solving the following generalized eigenvalue problem [20]

$$\mathbf{K}\mathbf{A}\mathbf{K}\mathbf{\Gamma} = \mathbf{K}\mathbf{K}\mathbf{\Gamma}\mathbf{\Lambda} \tag{15}$$

(where $\mathbf{\Lambda}$ is a diagonal matrix containing the generalized eigenvalues), or exploiting the eigenvalue decomposition of $\mathbf{K}$ [3, 39, 4]. Following the latter approach [3, 4] we write $\mathbf{K}$ as

$$\mathbf{K} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T, \tag{16}$$

where $\mathbf{\Sigma} \in \mathbb{R}^{L\times L}$ is a diagonal matrix containing the $L < N$ nonzero eigenvalues of $\mathbf{K}$, $\mathbf{U} \in \mathbb{R}^{N\times L}(\mathbf{U}^T\mathbf{U} = \mathbf{I})$ is the orthonormal matrix containing the corresponding normalized eigenvectors, and $\mathbf{I}$ is the identity matrix. Substituting (16) in (14) and setting

$$\mathbf{W} = \mathbf{\Sigma}\mathbf{U}^T\mathbf{\Gamma}, \tag{17}$$

the optimization problem can be expressed as

$$\underset{\mathbf{W}}{\operatorname{argmax}} \operatorname{tr}((\mathbf{W}^T\mathbf{U}^T\mathbf{A}\mathbf{U}\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{W})). \tag{18}$$

This is equivalent to finding all the eigenvectors of the $L\times L$ matrix $\mathbf{M} = \mathbf{U}^T\mathbf{A}\mathbf{U}$ satisfying the eigenvalue problem below

$$\mathbf{M}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}\,. \tag{19}$$

For the rank of $\mathbf{M}$ the following inequality holds $\operatorname{rank}(\mathbf{M}) < \min(H, L)$ [10], where typically $L \gg H$ (e.g. in the experimental results of Section 4 we vary H in the range $[3, 7]$).

Instead of directly performing the eigenvalue decomposition of $\mathbf{M}$, the special structure of matrix $\mathbf{A}$ can be exploited to significantly reduce the computational cost, as explained in the following. Setting $\mathbf{V} = \mathbf{U}^T = [\mathbf{V}_{1,1}, \ldots, \mathbf{V}_{\Omega,H_\Omega}]$, where $\mathbf{V}_{i,j} = [\mathbf{v}_{i,j}^1, \ldots, \mathbf{v}_{i,j}^{N_{i,j}}]$, $\mathbf{v}_{i,j}^n \in \mathbb{R}^L$, $\mathbf{M}$ can be expressed as

$$\mathbf{M} = \sum_{i=1}^{\Omega-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{\Omega}\sum_{l=1}^{H_k} \hat{p}_{i,j}\hat{p}_{k,l}(\mathbf{m}_{i,j} - \mathbf{m}_{k,l})(\mathbf{m}_{i,j} - \mathbf{m}_{k,l})^T, \tag{20}$$

where $\mathbf{m}_{i,j} = (1/N_{i,j})\sum_{n=1}^{N_{i,j}}\mathbf{v}_{i,j}^n$. Using (20), $\mathbf{M}$ can be easily factorized as

$$\mathbf{M} = \mathbf{F}\mathbf{F}^T \tag{21}$$

where, $\mathbf{F} \in \mathbb{R}^{L\times J}$ is

$$\mathbf{F} = [\sqrt{\hat{p}_{1,1}\hat{p}_{2,1}}(\mathbf{m}_{1,1} - \mathbf{m}_{2,1}), \ldots, \sqrt{\hat{p}_{\Omega-1,H_{\Omega-1}}\hat{p}_{\Omega,H_\Omega}}(\mathbf{m}_{\Omega-1,H_{\Omega-1}} - \mathbf{m}_{\Omega,H_\Omega})]. \tag{22}$$

The number of columns J of the matrix $\mathbf{F}$ can be computed using $J = \sum_{i=1}^{\Omega-1}\sum_{j=i+1}^{\Omega} H_iH_j$, where again $L \gg J$ (e.g. in Section 4 we have $\Omega = 2$, $H_1 \in [2, 6]$ and $H_2 = 1$).

The factorization of $\mathbf{M}$ in (20) can now be exploited by the cross-product algorithm [28, 4] to efficiently compute the $D = H - 1$ eigenvectors of $\mathbf{M}$ (instead of directly operating on the $L \times L$ matrix $\mathbf{M}$) as explained in the following. Let $\mathbf{F} = \mathbf{P}\mathbf{\Upsilon}\mathbf{Q}^T$ be the singular value decomposition of $\mathbf{F}$, where

$\mathbf{P} \in \mathbb{R}^{L \times L}$, $\mathbf{Q} \in \mathbb{R}^{J \times J}$ are orthogonal matrices and $\mathbf{\Upsilon}$ is a diagonal matrix containing the singular values of $\mathbf{F}$. Then, the columns of $\mathbf{P}$ (i.e. the left singular vectors of $\mathbf{F}$) are the eigenvectors of $\mathbf{M}$, and the columns of $\mathbf{Q}$ (i.e. the right singular vectors of $\mathbf{F}$) are the eigenvectors of $\mathbf{R} = \mathbf{F}^T\mathbf{F}$, $\mathbf{R} \in \mathbb{R}^{J \times J}$. Therefore, to efficiently derive the eigenvectors of $\mathbf{M}$, we first compute the spectral decomposition of $\mathbf{R} = \mathbf{Q}\mathbf{\Xi}\mathbf{Q}^T$ (which is of significantly smaller size than $\mathbf{M}$), and then compute the first J eigenvectors of $\mathbf{M}$ using $\mathbf{P}_J = \mathbf{F}\mathbf{Q}$, where $\mathbf{P}_J$ contains the eigenvectors of $\mathbf{M}$ corresponding to the nonzero eigenvalues. Therefore, the projection matrix $\mathbf{W}$ can be formed by the first $D(< J)$ columns of $\mathbf{P}_J$. Then, using (17) $\mathbf{\Gamma}$ can be computed as $\mathbf{\Gamma} = \mathbf{U}\mathbf{\Sigma}^{-1}\mathbf{W}$. The derived $\mathbf{\Gamma}$ can then be used for the projection of an observation $\mathbf{x}$ in the discriminant subspace using

$$\mathbf{z} = \mathbf{\Psi}^T\boldsymbol{\phi}(\mathbf{x}) = \mathbf{\Gamma}^T\mathbf{k}, \tag{23}$$

where $\mathbf{k} = [k(\mathbf{x}^1_{1,1}, \mathbf{x}), \ldots, k(\mathbf{x}^{N_\Omega, H_\Omega}_{\Omega, H_\Omega}, \mathbf{x})]^T$ and $\mathbf{z}$ is the projection of $\boldsymbol{\phi}(\mathbf{x})$.

E09 E14

Figure 1: Example keyframes for events E09 and E14 of the MED 2012 dataset.

4. EXPERIMENTAL EVALUATION

In this section, the MED 2010 and 2012 video collections are used for the comparison of the proposed method (GSDA-LSVM) with LSVM [12] and KSVM [32]. The Matlab [2] environment is used for the implementation of GSDA[2], while for LSVM and KSVM the libsvm [5] package was utilized. Moreover, for GSDA and KSVM the Gaussian radial basis function ($k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\rho\|\mathbf{x}_i - \mathbf{x}_j\|)$, $\rho \in \mathbb{R}_+$) is used as the base kernel [32].

4.1 Datasets

The video collections provided by the TRECVID MED evaluation task are among the most challenging in the field of event detection. For the evaluation of the proposed algorithm the MED 2010 and a subset of the MED 2012 datasets are utilized. In the following, we briefly describe the above datasets and their preprocessing for extracting the representations that are used as input to event detectors.

4.1.1 MED 2010

The TRECVID MED 2010 dataset consists of 1745 training and 1742 test videos belonging to one of 3 target events or to the "rest-of-world" event category. The target event names are provided in Table 1 (events T01-T03). For extracting the model vectors representing these videos, we follow the procedure described in Section 2.2. More specifically, the video signal is decoded and one keyframe every 6 seconds is extracted. The spatial information within each keyframe is encoded using a 1×3 pyramidal decomposition scheme, a dense sampling strategy and the opponentSIFT descriptor [30]. Subsequently, for each pyramid cell a Bag-of-Words (BoW) model of 1000 visual words is derived using the k-means algorithm and a large set of feature vectors. A soft assignment technique is then applied to represent each keyframe with a BoW feature vector in $\mathbb{R}^S$ [31], where $S = 4000$ is the total number of BoW centers along all pyramid levels. Then, a set of $F = 346$ visual concept detectors, based on LSVM classifiers and trained on the TRECVID SIN 2012 dataset [25], is used for deriving a model vector for each keyframe. The final model vector at video-level is computed by averaging the keyframe model vectors along the video.

[2]The Matlab implementation of GSDA is provided in `http://www.iti.gr/~bmezaris/downloads.html`.

T01: Assembling a shelter
T02: Batting a run in
T03: making a cake
E01: Attempting a board trick
E02: Feeding an animal
E03: Landing a fish
E04: Wedding ceremony
E05: Working on a woodworking project
E06: Birthday party
E07: Changing a vehicle tire
E08: Flash mob gathering
E09: Getting a vehicle unstuck
E10: Grooming an animal
E11: Making a sandwich
E12: Parade
E13: Parkour
E14: Repairing an appliance
E15: Working on a sewing project
E21: Attempting a bike trick
E22: Cleaning an appliance
E23: Dog show
E24: Giving directions to a location
E25: Marriage proposal
E26: Renovating a home
E27: Rock climbing
E28: Town hall meeting
E29: Winning a race without a vehicle
E30: Working on a metal crafts project

Table 1: Target events of TRECVID MED 2010 (T01-T03) and 2012 (E01-E15, E21-E30) datasets.

4.1.2 MED 2012

The TRECVID MED 2012 video corpus [25] consists of more than 5500 hours of user-generated video belonging to one of 25 target events or to other uninteresting events. The names of the target events are given in Table 1 (events E01-E15, E21-E30), while two example keyframes for events E09 and E14 are depicted in Fig. 1. For ease of comparison, we use the publicly available dataset and corresponding model vectors provided in [12]. This subset comprises 13274 annotated model vectors corresponding to an equal number of MED 2012 videos, and is divided to a training and evaluation partition of 8840 and 4434 model vectors, respectively. These model vectors were extracted using a procedure similar to that of Section 2.2. Specifically, low-level features were extracted using a 1×3 spatial pyramid decomposition scheme, three SIFT-based descriptors (SIFT, opponentSIFT and C-SIFT) and Fisher vector coding [12]. The above feature extraction procedure along with the TRECVID SIN 2012 [25] and ImageNet ILSVRC 2011 [1] datasets, annotated with 346 and 1000 concepts respectively, were used for creating a pool of $F = 1346$ LSVM-based concept detectors. Then, the MED 2012 model vectors were extracted by applying the concept detectors to one frame every two seconds, and averaging them along the video as previously.

4.2 Evaluation measure

The average precision AP_i [27] is utilized for assessing the retrieval performance of the i-th event detector

$$AP_i = \frac{1}{O_i} \sum_{n=1}^{N} \frac{O_i^n}{n} \zeta_n^i, \tag{24}$$

where, O_i, O_i^n, are the number of test videos belonging to the i-th event, and the number of i-th event videos in the n-top ranked list returned by the detection method; ζ_n^i is an indicator function with $\zeta_n^i = 1$ if the n-th video in the ranked list belongs to the i-th event and $\zeta_n^i = 0$ otherwise. The overall performance of a method is then measured using the mean AP (MAP) along all events in a dataset

$$MAP = \frac{1}{\Pi} \sum_{i=1}^{\Pi} AP_i \,, \tag{25}$$

where Π is the number of target events.

Event	*LSVM*	*KSVM*	*GSDA-LSVM*	*% Boost*
T01	0.106	0.213	**0.252**	18.3%
T02	0.477	0.651	**0.678**	4.1%
T03	0.103	0.293	**0.295**	0.6%
MAP	0.229	0.385	**0.408**	5.8%

Table 2: Performance evaluation on the TRECVID MED 2010 dataset; the last column depicts the boost in performance of GSDA-LSVM over KSVM.

Event	*LSVM*	*KSVM*	*GSDA-LSVM*	*% Boost*
E01	0.156	0.488	**0.583**	19.5%
E02	0.030	**0.175**	0.161	-7.8%
E03	0.234	0.441	**0.460**	4.4%
E04	0.273	0.579	**0.668**	15.4%
E05	0.051	0.156	**0.256**	64.2%
E06	0.131	0.181	**0.243**	34.6%
E07	0.059	0.285	**0.383**	34.4%
E08	0.383	0.564	**0.577**	2.4%
E09	0.252	0.463	**0.464**	0.2%
E10	0.061	0.260	**0.285**	9.8%
E11	0.043	**0.308**	0.307	-0.2%
E12	0.115	0.253	**0.286**	13.1%
E13	0.078	0.480	**0.510**	6.4%
E14	0.175	0.512	**0.515**	0.7%
E15	0.112	0.388	**0.451**	16.2%
E21	0.406	0.556	**0.572**	2.9%
E22	0.045	**0.174**	0.168	-3.5%
E23	0.406	0.612	**0.633**	3.5%
E24	0.032	**0.150**	0.142	-5.2%
E25	0.043	0.047	**0.078**	66.4%
E26	0.086	0.288	**0.327**	13.8%
E27	0.331	0.382	**0.441**	15.6%
E28	0.354	0.410	**0.479**	17.1%
E29	0.124	0.252	**0.277**	10.3%
E30	0.020	0.142	**0.197**	39.2%
MAP	0.160	0.341	**0.379**	10.9%

Table 3: Performance evaluation on the TRECVID MED 2012 dataset; the last column depicts the boost in performance of GSDA-LSVM over KSVM.

4.3 Experimental setup and results

In this section, the proposed GSDA-LSVM approach is evaluated using the TRECVID MED datasets described in Section 4.1, and its performance is compared with that of KSVM and LSVM.

4.3.1 Experimental setup

In the stage of model selection during the training of event detectors, for LSVM we need to identify the penalty term C, for KSVM both C and the scale parameter ρ of the Gaussian RBF kernel, while for GSDA-LSVM we additionally need to estimate the number of subclasses H. These parameters are estimated using a grid search on a 3-fold cross-validation procedure, where at each fold the development set is split to 70% training set and 30% validation set. During optimization, the LSVM parameter C and the Gaussian RBF parameter ρ of KSVM and GSDA-LSVM are searched in the range $[2^{-10}, 2^4]$. For the identification of the optimum number of GSDA subclasses, the k-means algorithm is used to evaluate different data partitions by varying H_1 in the range $[2, 6]$ (i.e., the "rest-of-the-world" class is not divided into subclasses).

4.3.2 Event detection accuracy

The performance of the different methods in terms of AP (and MAP along all events) in MED 2010 and 2012 is shown in Tables 2 and 3 respectively, where the best performance is printed in bold. Moreover, two keyframes for each of the 5 top ranked videos retrieved using the GSDA-LSVM algorithm for events E01 to E05 are shown in Fig. 2, where wrongly detected videos are presented within red colored frames. From the obtained results we observe that GSDA-LSVM provides the best performance in both datasets. In more detail, in the MED 2010 dataset we observe that GSDA-LSVM provides an approximate boost in performance over KSVM of approximately 5.8%, and that both kernel-based methods (KMSDA, GSDA-LSVM) achieve a MAP boost of more than 68% over the linear one (LSVM). From Table 3 we see that the performance of all methods is somewhat lower, in absolute numbers, in the more challenging (in terms of event diversity and scale) MED 2012 dataset. However, the performance differences between the methods are increased, in comparison to the differences observed in MED 2010; GSDA-LSVM provides a MAP boost of approximately 11% and 137% over KSVM and LSVM respectively.

4.3.3 Time complexity

For the evaluation of the proposed method in terms of time complexity two experiments are performed, as described in the following. GSDA extends KSDA (and similarly other subclass DA methods) by providing a new formulation of the eigenvalue problem that can be solved more efficiently. To this end, we compare GSDA with KSDA in terms of computational time for learning one MED 2012 event. In this experiment a speed up of around 25% of GSDA over KSDA was observed. This performance gain is achieved because the eigenanalysis performed in GSDA involves matrices of smaller dimension as explained in Section 3.2. In more detail, KSDA solves the generalized eigenvalue problem presented in (15) involving two $N \times N$ matrices, where $N = 8840$ (N is the number of kernel evaluations executed for constructing the kernel Gram matrix in (14), which is equal to the number of videos in the training dataset); in

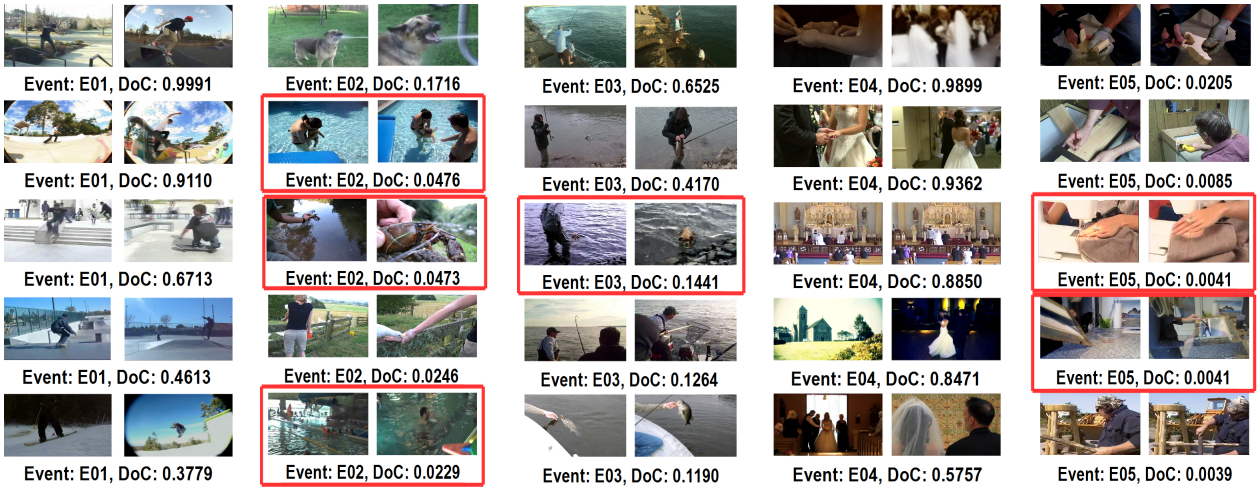

Figure 2: Example keyframes for the 5 top ranked videos retrieved using GSDA-LSVM algorithm for events E01 to E05; wrongly detected videos are presented within red colored frames.

contrary, GSDA requires the spectral decomposition of two matrices, specifically, an $N \times N$ matrix ($\mathbf{K}$ in (16)) and a much smaller one ($\mathbf{R}$ in Section 3.2), of dimension $J \times J$, where $J \in [3, 7]$.

Secondly, GSDA-LSVM was compared with KSVM and LSVM. In this experiment, a grid search was performed for identifying the optimum parameters of the above approaches on an Intel i7 3.5-GHz machine. In particular, we recorded the training times of GSDA-LSVM (where the number of subclasses remained fixed to $H = 2$), and KSVM for identifying ρ and C in a 5×5 optimization grid; for LSVM only a 5×1 grid as this approach includes only one parameter (the penalty term C). The evaluation results are shown in Table 4. From the obtained results we can see that GSDA-LSVM is approximately two times faster than KSVM. This performance gain is achieved because GSDA-LSVM can efficiently identify the best C in the reduced dimensionality space after the application of the GSDA phase of this approach. It should also be noted that the above results were obtained using an unoptimized Matlab implementation of GSDA. Finally, concerning testing times, similar values were observed for both GSDA-LSVM and KSVM. This was expected as testing time performance in kernel approaches is dominated by the kernel evaluations between the test observation and the annotated observations in the training dataset.

	LSVM	*KSVM*	*GSDA-LSVM*
Time (min)	8.67	103.54	52.10

Table 4: Time (in minutes) for selecting the parameters C and ρ of KSVM and GSDA-LSVM with $H = 2$ during training in MED 2012 dataset from a 5×5 grid; for LSVM a 5×1 grid is used as only C needs to be identified.

5. CONCLUSIONS

A novel video event detection method was presented that exploits a new efficient kernel subclass DA algorithm (GSDA) to extract the most discriminant concepts of the event, and LSVM for detecting the event in the GSDA subspace. The evaluation of the proposed method on the TRECVID MED corpora of 2010 and 2012 for the detection of 28 events in total showed that it favorably compares to the corresponding KSVM-based one in terms of both efficiency and accuracy.

Interesting extensions include the exploitation of spectral regression [4] or QR decomposition [33] in the GSDA criterion to further enhance the computational efficiency of the proposed algorithm. Another interesting future work direction is the investigation of mathematical formulations combining GSDA and LSVM in a single optimization criterion [38].

6. ACKNOWLEDGMENTS

This work was supported by the European Commission under contracts FP7-287911 LinkedTV and FP7-318101 MediaMixer.

7. REFERENCES

[1] Imagenet large scale visual recognition challenge 2011. `http://www.image-net.org/challenges/LSVRC/2011`, accessed 2013-09-21.

[2] *MATLAB, User' s Guide*. The MathWorks, Inc., 1994-2001.

[3] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Comput.*, 12(10):2385–2404, Oct. 2000.

[4] D. Cai, X. He, and J. Han. Speed up kernel discriminant analysis. *The VLDB Journal*, 20(1):21–33, Feb. 2011.

[5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27, 2011.

[6] B. Chen, L. Yuan, H. Liu, and Z. Bao. Kernel subclass discriminant analysis. *Neurocomputing*, 71(1–3):455–458, Dec. 2007.

[7] V. Franc, A. Zien, and B. Schölkopf. Support vector machines as probabilistic models. In *Proc. Int. Conf.*

on Machine Learning, pages 665–672, Bellevue, Washington, USA, June/July 2011.

[8] K. Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[9] N. Gkalelis, V. Mezaris, and I. Kompatsiaris. High-level event detection in video exploiting discriminant concepts. In *Proc. CBMI*, pages 85–90, Madrid, Spain, June 2011.

[10] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki. Mixture subclass discriminant analysis link to restricted Gaussian model and other generalizations. *IEEE Trans. Neural Netw. Learn. Syst*, 24(1):8–21, Jan. 2013.

[11] N. Gkalelis and V. Mezaris et al. Video event detection using a subclass recoding error-correcting output codes framework. In *Proc. IEEE ICME*, pages 1–6, San Jose, CA, USA, July 2013.

[12] A. Habibian, K. E. A. van de Sande, and C. G. M. Snoek. Recommendations for video event recognition using concept vocabularies. In *Proc. ACM ICMR*, pages 89–96, Dallas, Texas, USA, 2013.

[13] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *Int. J. Multimed. Info. Retr.*, Nov. 2013.

[14] Z.-Z. Lan, L. Jiang, and S.-I. Yu et al. CMU-Informedia at TRECVID 2013 multimedia event detection. In *Proc. TRECVID 2013 Workshop*, Gaithersburg, MD, USA, Nov. 2013.

[15] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Mach. Learn.*, 68(3):267–276, Oct. 2007.

[16] Z. Ma, A. G. Hauptmann, Y. Yang, and N. Sebe. Classifier-specific intermediate representation for multimedia tasks. In *Proc. ACM ICMR*, pages 50:1–50:8, Hong Kong, China, June 2012.

[17] M. Mazloom, E. Gavves, K. E. A. van de Sande, and C. G. M. Snoek. Searching informative concept banks for video event detection. In *Proc. ACM ICMR*, pages 255–262, Dallas, Texas, USA, Apr. 2013.

[18] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev. Semantic model vectors for complex video event recognition. *IEEE Trans. Multimedia*, 14(1):88–101, Feb. 2012.

[19] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers. Fisher discriminant analysis with kernels. In *Proc. IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX*, pages 41–48, Madison, WI, USA, Aug. 1999.

[20] C. B. Moler and G. W. Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis*, 10(2):241–256, Apr. 1973.

[21] G. K. Myers, R. Nallapati, and J. van Hout et al. Evaluating multimedia features and fusion for example-based event detection. *Machine Vision and Applications*, July 2013.

[22] P. Natarajan, R. Prasad, and U. Park et al. Multimodal feature fusion for robust event detection in web videos. In *Proc. IEEE CVPR*, pages 1298–1305, Providence, RI, USA, June 2012.

[23] K. Nelson. *Event Knowledge: Structure and Function in Development*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1986.

[24] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *IEEE ICCV*, Sydney, Australia, Dec. 2013.

[25] P. Over et al. TRECVID 2012 - an introduction to the goals, tasks, data, evaluation mechanisms and metrics. In *Proc. TRECVID 2012 Workshop*, Gaithersburg, MD, USA, Nov. 2013.

[26] R. Aly et al. The AXES submissions at TRECVID 2013. In *Proc. TRECVID 2013 Workshop*, Gaithersburg, MD, USA, Nov. 2013.

[27] S. Robertson. A new interpretation of average precision. In *Proc. 31st Int. ACM SIGIR Conf. on Research and development in information retrieval*, pages 689–690, Singapore, Singapore, July 2008.

[28] G. W. Stewart. *Matrix Algorithms, Volume II: Eigensystems*. SIAM: Society for Industrial and Applied Mathematics, Philadelphia, USA, Aug. 2001.

[29] C. Sun and R. Nevatia. Large-scale web video event classification by use of Fisher vectors. In *IEEE Workshop on Applications of Computer Vision, WACV*, pages 15–22, Clearwater Beach, FL, USA, Jan. 2013.

[30] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1582–1596, Sept. 2010.

[31] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(7):1271–1283, Sept. 2010.

[32] V. Vapnik. *Statistical learning theory*. New York: Willey.

[33] T. Xiong, J. Ye, Q. Li, R. Janardan, and V. Cherkassky. Efficient kernel discriminant analysis via QR decomposition. In *Neural Information Processing Systems (NIPS)*, Vancouver, British Columbia, Canada, Dec. 2004.

[34] J. Yang and D. Chu et al. Sparse representation classifier steered discriminative projection with applications to face recognition. *IEEE Trans. Neural Netw. Learning Syst.*, 24(7):1023–1035, July 2013.

[35] D. You, O. C. Hamsici, and A. M. Martinez. Kernel optimization in discriminant analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(3):631–638, Mar. 2011.

[36] J. Yu, D. Liu, D. Tao, and H. S. Seah. Complex object correspondence construction in two-dimensional animation. *IEEE Trans. Image Process.*, 20(11):3257–3269, Nov. 2011.

[37] J. Yu, D. Liu, D. Tao, and H. S. Seah. On combining multiple features for cartoon character retrieval and clip synthesis. *IEEE Trans. Syst., Man, Cybern. B*, 42(5):1413–1427, Oct. 2012.

[38] S. Zafeiriou, A. Tefas, and I. Pitas. Minimum class variance support vector machines. *IEEE Trans. Image Process.*, 16(10):2551–2564, Oct. 2007.

[39] M. Zhu and A. M. Martínez. Pruning noisy bases in discriminant analysis. *IEEE Trans. Neural Netw.*, 19(1):148–157, Jan. 2008.