PER-EXEMPLAR ANALYSIS WITH MFOM FUSION LEARNING FOR MULTIMEDIA RETRIEVAL AND RECOUNTING

A Dissertation
Presented to
The Academic Faculty

By

Ilseo Kim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2013

PER-EXEMPLAR ANALYSIS WITH MFOM FUSION LEARNING FOR MULTIMEDIA RETRIEVAL AND RECOUNTING

Approved by:

Dr. Chin-Hui Lee, Advisor
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Bo Hong
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ghassan Al-Regib
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Sung Ha Kang
Professor, School of Mathematics
Georgia Institute of Technology

Dr. James McClellan
Professor, School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: May 1, 2013

To my wife and parents,

Min Jung Kim, Sangbae Kim, and Myounghee Park.

ACKNOWLEDGEMENTS

I owe my thanks to so many people around me for their guidance, encouragement, support, and love during my study.

I would like to express my profound gratitute to my advisor, Prof. Chin-Hui Lee, for his guidance and support throughout my Ph.D. study at Georgia Institute of Technology. He has supported me with his great insights and vision. Without his inspirational devotion and energy for research, I would not be able to complete this study. I have been so blessed to have him as a teacher.

I sincerely thank Dr. James McClellan, Dr. Ghassan Al-Regib, Dr. Bo Hong, and Dr. Sung Ha Kang for serving as members of my dissertation committee. They provided many comments and suggestions that are helpful to improve the quality of this work.

I also thank Dr. Sangmin Oh and Dr. Amitha Perera at Kitware Inc. for their careful guidance and valuable discussion during my internship. Working experience with them was a breakthrough in my research.

I consider myself fortunate to have many friends that contribute my studies. I would like to thank my group members: Yu Tsao, Sibel Yaman, Jeremy Reed, Byungki Byun, Aleem Mushtaq, You-Chi Cheng, I-Fan Chen, Zhen Huang, Kehuang Li, and Sadia Shakil for their friendship and fruitful collaborations. I would also like to acknowledge many friends from the Center for Signal and Image Processing (CSIP): Sunghwan Shin, Jonathan Kim, Seokchul Kwon, Ted Wada, and Marco Siniscalchi. I greatly appreciate my friends at Georgia Institute of Technology, Sungkap Yeo, Hyunwoong Kim, David Lee, Daehyun Kim, and Hanseung Lee as dependable friends for sharing unforgettable memory during the study.

Finally, I thank my parents, Sangbae Kim and Myounghee Park, for their love and consistent support throughout my life and during my Ph.D. studies. I would also like to express great thanks to my sister, Ilmin Kim, my parents-in-law, Myeong Joon Kim and

Nam Joo Hahn, and my sister-in-law, Min Sun Kim, for their universal support. Most of all, I was able to finish this dissertation since I have been given the greatest and deepest support from my wife, Min Jung Kim. She has always supported me with unbelievable love and patience.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this research is to develop a framework of a per-exemplar analysis with MFoM fusion learning for multimedia retrieval and recounting. As a large volume of digital video data becomes available, along with revolutionary advances in multimedia technologies, demand related to efficiently retrieving and recounting multimedia data has grown. However, the inherent complexity in representing and recognizing multimedia data, especially for large-scale and unconstrained consumer videos, poses significant challenges. In particular, the following challenges are major concerns in the proposed research.

One challenge is that consumer-video data (e.g., videos on YouTube) are mostly unstructured; therefore, evidence for a targeted semantic category is often sparsely located across time. To address the issue, a segmental multi-way local feature pooling method by using scene concept analysis is proposed. This scheme demonstrated benefits over conventional methods by constructing clip-level representations via average-based global pooling. The key idea of the framework is to utilize similarities between two videos in terms of various scene concepts and to improve a discriminative power by using kernelization techniques. In particular, the proposed method utilizes scene concepts that are pre-constructed by clustering video segments into categories in an unsupervised manner. Then, a video is represented with multiple feature descriptors with respect to scene concepts. Finally, multiple kernels are constructed from the feature descriptors, and then, are combined into a final kernel that improves the discriminative power for multimedia event detection.

Another challenge is that most semantic categories used for multimedia retrieval have inherent within-class diversity that can be dramatic and can raise the question as to whether conventional approaches are still successful and scalable. To consider such huge variability and further improve recounting capabilities, a per-exemplar learning scheme is proposed with a focus on fusing multiple types of heterogeneous features for video retrieval. While the conventional approach for multimedia retrieval involves learning a single classifier per

category, the proposed scheme learns multiple detection models, one for each training exemplar. In particular, a local distance function is defined as a linear combination of element distance measured by each features. Then, a weight vector of the local distance function is learned in a discriminative learning method by taking only neighboring samples around an exemplar as training samples. In this way, a retrieval problem is redefined as an association problem, i.e., test samples are retrieved by association-based rules.

In addition, the quality of a multimedia-retrieval system is often evaluated by domain-specific performance metrics that serve sophisticated user needs. To address such criteria for evaluating a multimedia-retrieval system, in MFoM learning, novel algorithms were proposed to explicitly optimize two challenging metrics, AP and a weighted sum of the probabilities of false alarms and missed detections at a target error ratio. Most conventional learning schemes attempt to optimize their own learning criteria, as opposed to domain-specific performance measures. By addressing this discrepancy, the proposed learning scheme approximates the given performance measure, which is discrete and makes it difficult to apply conventional optimization schemes, with a continuous and differentiable loss function which can be directly optimized. Then, a GPD algorithm is applied to optimizing this loss function.

CHAPTER 1

INTRODUCTION

Along with advances in multimedia technologies, video data are being generated and shared through the internet (e.g., YouTube and Facebook) at an unexpected pace. For example, on YouTube, approximately 72 hours of video are being uploaded every minute, and over 4 million hours of video are watched each month [1]. Accordingly, the demand related to retrieval, organization, and recounting of this huge amount of multimedia data has grown. However, in real-world problems, the inherent complexity in representing and recognizing multimedia data poses significant challenges. The research presented in this thesis contributes to developing a novel framework that successfully addresses such challenges.

In this thesis, the author mainly examines a task of multimedia event detection (MED), of which the goal is to search video recordings by the main event appearing in them. In such a context, a multimedia event is defined as a combination of complex human actions, processes, and activities that involve people interacting with other people and/or objects. These events are loosely or tightly organized and have significant temporal and semantic relationships with some overarching activities, e.g., *making a sandwich* or *attempting a board trick*. In addition, the author explores a task of multimedia event recounting (MER), in which the goal is to provide a user with a set of evidence to indicate the presence of a multimedia event in a video.

For the MED and MER tasks, this thesis assumes the usage of real-world consumer video data (e.g., videos on YouTube) that are usually of a large scale and unconstrained in many ways, including the temporal, spatial, and contextual aspects. Such data give rise to the following problems. First of all, they are mostly unstructured along the temporal axis. Therefore, evidence for a multimedia event is often sparsely located across time. For example, assume a video clip labeled with the *wedding ceremony* category. In the clip, all sub-events occurring on a wedding day, e.g., sunrise, interviews from friends, make-up for

the bride, and the wedding march, may all be recorded in a random sequence. However, only some sub-events can be highly correlated to the event category. Yet, because the video clip is temporally unstructured, capturing such highly correlated parts in a coherent manner is often not easy, e.g., it is difficult to directly apply conventional hidden Markov models (HMMs) to even detection. Another issue is that most multimedia events have inherently diverse within-category variations. For example, consider the *feeding an animal* category. This event category conveys a large variety of animal types from a dog to a giraffe. Their appearances and ways to feed them could largely differ. Therefore, it is questionable whether conventional techniques, which learn a detection model per class, can still be successful and scalable. Yet another challenge is that the quality of a multimedia retrieval system is often evaluated by domain-specific performance metrics that serve user needs. These metrics sometimes require complex formulations over simple precision or recall, involving rank ordering of retrieved results or a user-defined operating point. Solutions obtained with conventional learning methods that reduce simple classification errors, e.g., support vector machines (SVMs) [2], could obviously be not as consistent in performance measures as those obtained with by learning methods that directly optimize the target metrics.

1.1 Contribution of this Research

Considering the aforementioned core research issues, the author presents a novel framework for multimedia event detection and recounting. In particular, the proposed framework incorporates novel developments into a system with the following three major contributions.

- Addressing sparseness of discriminative evidence in temporally unstructured videos by using multiple feature descriptors with *scene concept analysis*.
- Capturing content variability within a multimedia event category and enhancing multimedia recounting capability by *per-exemplar learning*.

- Learning detection models that *explicitly optimize domain-specific performance metrics* that have been widely used for multimedia retrieval.

First, this thesis addresses the problem of representing temporally unstructured videos. A conventional method to represent this type of videos is to extract features across all frames (segments) in a clip and to average them into a single clip-level descriptor, as seen in [3, 4]. However, this method is likely to fail in dealing with the aforementioned sparseness of the discriminative video parts, by diluting them with other competing yet less discriminative ones. In contrast, the author proposes the use of multiple feature descriptors with scene concept analysis. The proposed method leverages upon segment-level (sub-clip) information and represents a video clip with multiple descriptors in which each of them is designed to relate to a specific scene category. Detail on the related work will be presented in Chapter 4.

Second, this work proposes a novel per-exemplar learning scheme that deals with issues regarding content variability within a multimedia event category. The diversity can be large, especially when it involves sophisticated concepts and activities. This situation is difficult to handle with the conventional thinking of one classifier per category technique, since diverse local characteristics of training samples are not likely to be reflected to a single global model. In contrast, the proposed scheme learns a local detection model per training exemplar. In this way, a test sample is retrieved according to the similarity with respect to various training exemplars provided by corresponding local models. In addition, MER functions for a retrieved test sample can be enhanced by referring to its sufficiently similar exemplars. This will be discussed in Chapter 5.

Finally, the author presents a learning scheme that explicitly optimizes two widely used performance metrics for multimedia retrieval, i.e., average precision (AP) and a weighted sum of the probabilities of missed detection and false alarms at a desired error ratio. These metrics require complex formulations, e.g., rank ordering for the former and a user-specific

operating point for the latter. The proposed learning scheme incorporates such formulations into approximating target metrics, while it can be considered as an extension to a recently proposed maximal figure-of-merit (MFoM) learning framework [5]. In particular, the proposed scheme is applied to combining base classifier outputs learned from multiple features in order to generate a final fusion score. More detail can be found in Chapter 6.

In all, the above three contributions are integrated into the proposed framework as follows: scene concept analysis for constructing feature descriptors, per-exemplar learning for designing a fusion classifier, and MFoM learning for computing contribution-weights of training exemplars to the final retrieval scores. In addition to the three major contributions, this thesis provides studies on a set of various feature types, along with effective methods to utilize them for the MED task with an extensive performance comparison among them. Moreover, the MER capabilities of the proposed framework are also discussed by using multiple features in various granularities.

1.2 Organization of the Dissertation

The remainder of the thesis is organized as follows. The related work to this study is summarized in Chapter 2, covering various techniques discussed in the presented research. In Chapter 3, the feature types used in this dissertation and their individual performance for multimedia event retrieval are presented. The three major contributions of the presented research are discussed in the following three chapters, respectively: scene concept analysis in Chapter 4, per-exemplar learning in Chapter 5, and explicit optimization of domain-specific performance metrics by MFoM in Chapter 6. In Chapter 7, the integrated system that utilizes the advantages of the above three components for MED and MER tasks is discussed. Finally, in Chapter 8, the proposed framework is summarized, and the future work is discussed.

CHAPTER 2

BACKGROUND AND RELATED WORK

With a focus on MFoM and per-exemplar learning with scene concepts, particularly for large-scale consumer video data, the proposed research is related to five areas of work. They are multimedia event detection and recounting (Section 2.1), local feature pooling for image and video retrieval (Section 2.2), exemplar-based local learning (Section 2.3), multimodal feature fusion (Section 2.4), and explicit performance metric optimization (Section 2.5). An overview on each of the five topics are given in the following.

2.1 Multimedia Event Detection and Recounting

In the past, available video data were most movies, TV broadcasts, or homemade videos. However, revolutionary advances in digital multimedia techniques have recently been witnessed. As a result, online services for sharing and archiving personal videos have become popular [1]. To satisfy the demand in processing such video data, research to develop techniques for efficient and effective video retrieval and recounting has been conducted in many areas, such as video shot detection [6], video classification [7], and multimedia event detection (MED) [8, 9].

In the early stages (in the early 2000's), research on video retrieval has been focused on the use of pre-defined lexicons of concepts, e.g., large scale concept ontology for multimedia (LSCOM) [10, 11] and MediaMill [12]. Although these lexicons typically cover a wide range of concepts, they fail to consider the fact that the appearance of a concept can vary from one event category to another. For example, people may dress differently, depending on the environment, e.g., different cultures and weather conditions. In addition, most existing work has used a very limited number of video examples in learning, since collecting and labeling a large-scale dataset are usually painful and expensive in terms of both time and human labor. Moreover, most early work oversimplified problems to constrained

datasets, such as news broadcasting videos that consisted of videos collected in controlled environments with clear backgrounds and little camera motion [13, 14]. However, it is difficult to extend such techniques developed within these constrained videos to YouTube-style consumer videos because of their unbounded properties in content, structure, length, and quality. Overall, despite many efforts to tackle problems in the early stages of research, most methods did not properly address the challenging aspects of consumer video analysis [15].

To address the aforementioned issues, the computer vision and multimedia processing communities have promoted research by simulating real-world environments. For example, the Columbia consumer video (CCV) dataset, which is of large-scale, was provided in [16]. The dataset contains 9,317 YouTube videos in over 20 semantic categories, which were collected with an extra care to ensure relevance to consumer's interest and originality of video content without post-editing. In [16], a set of precomputed features is also provided to the community for research use. Another good example is the University of Central Florida (UCF) YouTube action dataset [17, 18], which contains 11 action categories. This collection is also a good test bed for real-world problems due to a large variation in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, and illumination conditions.

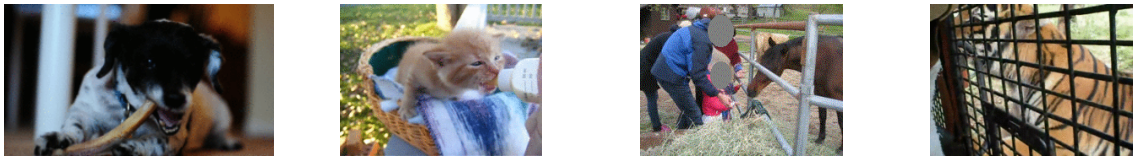
Recently, a series of evaluation campaigns has been organized by the Text Retrieval Conference (TREC), supported by the National Institute of Standards and Technology (NIST). TREC has arranged a technical session devoted to video data as TREC video (TRECVID) [19], and has provided numerous large-sized consumer video archives. Annual competitions with various challenging tasks in multimedia retrieval have been conducted in TRECVID. For example, approximately 140,000 video clips, with a total running time of 5,570 hours, and 30 classes of multimedia events were provided for the TRECVID 2012 MED task [20]. The dataset simulates various aspects of real-world problems. As an example, there are only ~150 positive training samples for each multimedia event category,

Table 1. The list of multimedia event classes for the TRECVID 2012 MED task.

ID	Event Name	ID	Event Name
E001	Attempting a board trick	E016	Doing homework or studying
E002	Feeding an animal	E017	Hide and seek
E003	Landing a fish	E018	Hiking
E004	Wedding ceremony	E019	Installing flooring
E005	Working on a woodworking project	E020	Writing text
E006	Birthday party	E021	Attempting a bike trick
E007	Changing a vehicle tire	E022	Cleaning an appliance
E008	Flash mob gathering	E023	Dog show
E009	Getting a vehicle unstuck	E024	Giving directions to a location
E010	Grooming an animal	E025	Marriage proposal
E011	Making a sandwich	E026	Renovating a home
E012	Parade	E027	Rock climbing
E013	Parkour	E028	Town hall meeting
E014	Repairing an appliance	E029	Winning a race without a vehicle
E015	Working on a sewing project	E030	Working on a metal crafts project

snowboard surfboard skateboard fingerboard

(a) E001-*Attempting a board trick*



dog cat horse tiger

(b) E002-*Feeding an animal*

church garden beach mid-eastern culture

(c) E004-*Wedding ceremony*

Figure 1. The diversity of visual content in the TRECVID 2012 MED classes.

which creates a huge imbalance in the number of positive and negative samples. The length and quality of video clips also differ drastically. Furthermore, because the dataset was collected from the internet, semantic concepts used in multimedia event detection, shown in Table 1, are complex in nature and consist of a number of human interactions among people and/or objects that are often loosely organized. Accordingly, this large within-category content variability poses significant challenges in categorizing them. Such variability within a multimedia event category is illustrated in Figure 1: (a) snowboard, surfboard, skateboard, and fingerboard scenes in E001-*Attempting a board trick*; (b) dog, cat, horse, and tiger scenes in E002-*Feeding an animal*; and (c) church, garden, beach, and Mid-eastern-culture scenes in E004-*Wedding ceremony*. The TRECVID datasets have become one of the most widely used multimedia corpora, and many state-of-the-art techniques have been developed and verified by using them. Overall system architectures of work in this area can be found in [21, 22, 23, 24, 25], which have been reported along with the TRECVID annual competitions.

2.2 Local Feature Pooling for Image and Video Retrieval

In the past, local feature pooling was studied mostly in the computer vision and multimedia processing communities for image classification. In [26, 27], a spatial pyramid pooling scheme was proposed to leverage the spatial layout of images into feature representation. This scheme works by placing a sequence of increasingly coarser grids over the feature space and separately applying bag-of-words (BoW) feature representations [28] at each level of grid. Such spatial pyramid pooling scheme showcased significant improvement on image scene categorization tasks, especially when a structured locale relationship among object components exists, e.g., the sky and the sea are located in the upper and lower regions of an image, respectively. On the other hand, in [29, 30, 31], a latent discriminative learning scheme was proposed, where image regions with more relaxed spatial relationship can be incorporated to recognize an object. As an example, this scheme can recognize a

human body by using local features in arm and leg regions, while their location can vary by postures. As for using multiple instances to detect an object, latent discriminative learning is closely related to multiple instance learning (MIL) [32].

In terms of taking advantage of deforming a sample to sub-regions and learning discriminative models, the proposed learning scheme, to be discussed in Chapter 4, is related to a latent-SVM scheme [29, 30]. However, the two schemes are sufficiently different since pooled features from all of the video regions are preserved in the proposed scheme, while image patches are compactly represented as confidence scores for a few of the most discriminative latent variables in the latent-SVM scheme [29, 30]. Furthermore, a latent-SVM scheme requires a prolonged search to determine the most discriminative latent variables. Therefore, applying it to a large-scale dataset is difficult because of the computational complexity, especially for the case that kernelization is required such as the MED task examined in this thesis.

Recently, research on local feature pooling has also been conducted for video retrieval problems. While an image scene or object can be modeled with sub-region grids or image patches, a video can be represented with sub-temporal regions or video segments. Recent work that utilizes local feature pooling for video retrieval includes [33], [34], and [35]. To recognize human activities, [33] models temporal structures of decomposable motion segments and learns a discriminative classifier for each of them. Then, recognition is made based on the quality of matching between the learned classifiers and temporal segments in a query sequence. While [33] showed promising performance in activity recognition, it is still most suitable for videos with considerably regularized structures, such as the Olympic sports activity dataset, provided in [33] along with the scheme. In contrast to [33], [34] tackles the problem of understanding the temporal structure of complex events in highly unstructured videos, by utilizing a conditional model that automatically discovers discriminative segments of video. In particular, it introduces latent variables over the frames of a video and assigning sequences of states that are most discriminative for a target event.

The potential challenge for [34] is that the scheme is associated with a large number of parameters, and accordingly, it often requires a large training dataset to learn such parameters. Moreover, non-linear kernelization techniques, which are known to be crucial for multimedia retrieval tasks, were not utilized in [34]. In [35], a scene aligned pooling (SAP) scheme was proposed based on the observation that a video clip is often composed of shots involving different scenes. This scheme decomposes video features into concurrent scene components, which are described by using a secondary image feature, e.g., GIST [36], and constructs classification models that are adaptive to different scenes. However, using the secondary feature type to construct scene clusters may introduce inconsistent scene alignment, especially when a considered feature type shows fairly different characteristics from the secondary feature. Furthermore, the SAP scheme is limited to image-based features only.

Among the previously related work about local feature pooling, [35] is related to the proposed feature pooling scheme, to be discussed in Chapter 4, in terms of using multi-way feature pooling. However, they are also fairly different because the proposed scheme does not require a secondary image feature, and accordingly, is more general in its application to various audio/visual feature types. In addition, the proposed scheme that utilizes kernelization techniques without L_1-normalization provides an improved discriminant power, when the kernels from features by multi-way pooling are combined.

2.3 Exemplar-based Local Learning

Next, the proposed per-exemplar learning, to be discussed in Chapter 5, builds upon past work in two areas. First of all, in terms of learning a local distance function around a training exemplar, it is related to association-based object recognition in images [37, 38, 39]. In [37], a local learning scheme that exploits local perceptual distance for image retrieval and classification was proposed. In the view of various image attributes, such as shape, color, and texture, this local learning scheme aims to address the large variation among images

within the same semantic category. In particular, a local distance function is learned for each training image as a combination of elementary distances between patch-based visual features. In addition, [38] proposed per-exemplar distance learning, which is suitable for object recognition tasks. In particular, it trains a local function per training exemplar to return interpretable distances, which can be analyzed in absolute terms. Furthermore, in [39], the concept of per-exemplar distance learning is extended to an exemplar-SVM scheme, where decision boundaries of local functions can be learned in a flexible way. By forcing a training exemplar to have a maximally attainable similarity, [39] showed enhanced capability to incorporate input from negative samples into the learning process.

Second, on the subject of learning localized discriminative functions by using neighboring samples, the proposed per-exemplar learning is related to a discriminative nearest-neighbor learning scheme [40, 41]. In [40], a locally adaptive form of nearest neighbor learning was proposed in an attempt to ameliorate a bias issue in high-dimensional feature spaces. In particular, [40] estimated an effective metric for computing neighborhoods by using a local linear discriminant analysis. Then, the local decision boundaries were determined from centroid information, and neighborhoods were shrunk in the directions orthogonal to these local decision boundaries. Thereafter, [40] argued that any neighborhood-based classifier can be employed on these modified neighborhoods. On the other hand, instead of deforming the distance metric, [41] proposed a k-nearest-neighbor SVM (KNN-SVM) method that finds neighborhoods close to a query sample. The KNN-SVM method preserves the distance function by learning a local SVM on the collection of neighbors.

Compared to the first category of research [37, 38, 39], the proposed per-exemplar learning scheme, to be discussed in Chapter 5, exploits discriminative elementary distance to identify the relevance of features per training exemplar. In particular, while [37, 38, 39] used generative elementary distance, e.g., feature-wise L_2-distance, the proposed scheme uses discriminative elementary distance generated by a base classifier learned from each feature type. This discriminative elementary distance is useful, especially when we need to

incorporate multiple features in high-dimensional spaces, by alleviating a bias issue with its compact representation. In addition, video retrieval tasks, examined in this thesis, are usually quite different from image retrieval tasks, studied in [37, 38, 39]. With an extensive usage of both visual and audio features, the proposed scheme provides a new perspective beyond relatively constrained image retrieval problems. Compared to the discriminative nearest-neighbor schemes [40, 41], the proposed scheme yields explicit and well-defined distance functions and provides a principled manner to compute confidence scores for new test samples, along with additional recounting capabilities.

2.4 Multi-modal Feature Fusion

The use of fusion to combine multi-modal features is crucial in multimedia event detection and recounting. Unlike speech or a text document, a video can convey multi-modal features, including acoustic/speech, spatial vision, and temporal dynamic information. The capability of fusion is particularly necessitated by the huge content variability in consumer videos. As an example, consider a birthday party video, captured in a dark room showing a cake with lighted candles. In such cases, evidence from only visual features may be too weak to strongly trigger a system to identify the video as a hit. However, audio features may provide strong evidence by capturing sound types such as a birthday song, laughter and clapping.

The benefits of fusion for multimedia retrieval have been demonstrated in the recent literature. For example, in [16], the CCV dataset was introduced with a benchmark system that uses SVMs as fusion classifiers. It showed that retrieval performance gradually improves when additional feature types are incorporated into the system by feature concatenation. In [42], for automatic categorization of videos, text-level label features, e.g., related videos, searched videos and text-based webpages, were fused by incorporating a manually designed semantic hierarchy. It showed the effectiveness of the fusion scheme

by extensive experiments on approximately 8,000 videos on YouTube. Moreover, fusion-based tag-recommendation methods were presented in [43, 44]. They demonstrated that the fusion of web tagging and audio-visual content in videos improves tag-recommendation qualities on YouTube videos. Recently, feature fusion for multimedia event detection has been widely studied by participants of the TRECVID tasks [9, 8, 45].

In the field of multimedia event detection and recounting, fusion can be largely categorized into two types: *early* and *late fusion*. An example of an early fusion method is concatenating multiple feature vectors into a large feature representation. Consequently, this early fusion method learns a fusion classifier by using all features jointly from the early stages [16]. However, consider cases that each feature is represented as a high-dimensional vector, such as a bag-of-words (BoW) representation with thousands codewords [28]. In such cases, assuming that numerous feature types are available, this concatenation might not be suitable for large-scale data because of both computational and memory-wise costs. Another example of an early fusion method is multiple kernel learning (MKL) [46]. The MKL scheme combines kernels constructed from multiple features into a fused kernel, by linear combination, weighted products, or both. The combined kernel is usually more discriminative than individual kernels. However, MKL does not systematically support the optimization of domain-specific performance metrics, and reported results are not always competitive [47].

On the other hand, in a late fusion method, a decision model is learned within a hierarchical approach. First, weak base classifiers are separately trained by using individual features. Then, outputs from the base classifiers, such as rankings, distance from a decision boundary, or loss functions, are collected and used in learning a final fusion classifier. For example, in [48, 49], a discriminative score fusion scheme, founded in model-based transformation (MBT), was proposed. The MBT fusion scheme can be regarded as supervised mapping from low- or intermediate-level feature space to high- or semantic-level space. Another example of a late fusion method is the use of boosting for fusion [47, 17]. In

terms of using outputs of base classifiers, the proposed research can be categorized into late fusion.

Numerous late fusion methods have been studied across the communities of multimedia processing, computer vision, and machine learning. Again, they can be grouped into three categories. One category performs score normalization before output scores from base classifiers are combined [50, 51, 52]. Normalizing scores is particularly necessitated when types of base classifiers and their learning procedures are fairly different across systems, and accordingly, the distribution of generated scores from the base classifiers is inconsistent. Under such circumstances, score normalization may improve the robustness of a fusion classifier. However, normalization schemes by most of these methods require expert knowledge, which are sometimes not available for unseen samples. In this thesis, it is assumed that base classifiers are designed in a consistent manner, and therefore, the proposed framework does not require sophisticated score normalization techniques.

Another category of late fusion applies fixed rules, e.g., a summation or product of base classifier scores with uniform weights, regardless of the actual distributions of the scores. In [53], various fusion rules were studied with extensive experiments. Recently, [54] reported that the geometric mean works as effectively as other sophisticated rules, despite its simple formation. Assigning different weights by cross validation prior to a combination of base classifier scores can also be categorized into this group. However, despite advantage of simplicity and non-dependency on expert knowledge, there may exist a chance to improve fusion performance by systemically learned fusion schemes over simple fixed rules.

The third category of late fusion attempts to systemically learn a fusion classifier to combine base classifier scores. For example, in [55], a fusion scheme learns weights by optimizing different target metrics with various regularization methods. In [56], confidence scores from base classifiers are collected in order to form a feature vector, and then a fusion classifier is learned by a sample-based approach. In [57], multiple localized fusion classifiers instead of a single fusion classifier are learned across multi-dimensional score

space in a local expert forest (LEF) learning scheme. The fusion scheme, to be presented in Chapter 6, also belongs to this category in terms of learning a fusion classifier in the space of base classifier scores. In contrast to other methods in this category, the proposed framework exploits a robust fusion learning scheme by addressing inconsistency between scores of seen training and novel test samples.

2.5 Explicit Performance Metric Optimization

Finally, in many pattern recognition problems, the success of learning algorithms is often evaluated by a domain-specific performance metric that simulates real-world user needs. In particular, specific performance metrics, such as the weighted ratios of precision and recall, false alarms, F-scores, or any combinations of these, count to measure the quality of the system and the potential user experience. For example, the precision of top-ranked retrieval results was used in [43, 16]; F_1 scores were used in [42]; and the ratio of 12.5:1 between the probability of missed detection and false alarms was used for the TRECVID MED task in [19]. However, most learning methods use training according to their own learning criteria, and not a preferred performance measure. This discrepancy could potentially create mismatches between training and testing conditions, and thus could likely yield suboptimal solutions.

Learning with explicit performance metric optimization has been studied mostly in the machine learning community, albeit sparsely. The proposed learning scheme, to be presented in Chapter 6, is based on efforts attempting to directly optimize a targeted performance metric. In particular, the proposed scheme introduces a continuous and differentiable objective function that simulates a discrete performance measure of interest. A good example of previous work in this area is the minimization of classification error rate (MCE) learning [58]. The MCE learning addresses the fact that, in many realistic applications, the distribution of features is rarely known, precisely. In particular, the MCE learning approximates a misclassification measure to a continuous function regarding classifier parameters

and directly minimizes the approximated measure. In [59, 5], the maximal-figure-of-merit (MFoM) learning was proposed to integrate more flexible performance metrics over a misclassification measure, such as accuracy, recall, precision, or F-scores. In particular, the MFoM learning incorporates any performance metric that can be formulated with the four essential components in the confusion table, i.e., true-positive, true-negative, false-positive and false-negative terms, in a differentiable loss function. Then, the MFoM learning optimizes the metric with advanced optimization techniques such as a generalized probabilistic descent (GPD) algorithm [60].

The learning scheme, to be presented in Chapter 6, can be considered as an extension of the previous research in the MFoM learning to optimizing the following performance measures: (1) average precision (AP) [61] and (2) a weighted sum of the probability of missed detection (P_{MD}) and false alarms (P_{FA}) at a preferred operating point [62]. These metrics are more sophisticated over simple error rates, and accordingly, have been widely used in evaluating multimedia retrieval systems while simulating user needs. Since they involve complex formulations, which are ranking ordering for the former metric and a desired operating point (ratio) for the latter one, it is difficult to directly apply the conventional MFoM learning. The novel extended MFoM learning scheme that incorporates such complex conditions was studied in [63, 64] as preliminary work for the research presented in this thesis.

Optimization of ranking performance measures such as AP to be discussed in Section 6.2, has been studied mostly in the machine learning and information retrieval communities. In most conventional approaches, a loss function is defined by incorporating pair-wise rankings, which are taken as sample instances for learning. Then, a classifier is trained to correctly order the pairs. For example, in Ranking SVM [65] and RankBoost [66], a surrogate loss function for a ranking measure is defined based on the pair-wise losses. In [67], a neural network model was proposed to optimize the expected value of pair-wise ranking metrics. In [68, 69], a pair-wise ranking is smoothed and used in formulating a

differentiable objective function that simulates the area under a receiver operating characteristic (ROC) curve (AUC-ROC) [70]. In addition, [71] proposed a pair-wise rank-based loss function, considering *good* and *bad* neighbors of an instance.

Although learning schemes by using pair-wise rankings have shown promising performance, an objective function of pair-wise learning is formalized to minimize errors in ordering individual sample pairs, rather than minimizing errors in ordering an entire set of samples. Moreover, it is often computationally too costly for practical uses. As an effort to address these issues, [72, 73] proposed list-wise methods, in which sample lists instead of pairs are adopted as learning instances. In addition, [74] proposed an efficient gradient computational approach to optimizing AP, based on the observation of AP values with respect to individual score changes. To address the computational complexity issues, the trade-off between accuracy and complexity for linear ranking functions was explicitly studied in [75]. In particular, a linear function along with a feature selection scheme showed substantially reduced online complexity. In addition, an early-exit scheme was proposed in [76], based on the context of decision tree ensembles. In this scheme, samples that do not appear to be relevant to a given query were not further evaluated in the learning process. Although [75, 76] have shown considerably reduced complexity, they are sufficiently different from the proposed learning scheme that uses entire training data, to be discussed in Chapter 6. In terms of considering AP as a function with respect to individual sample scores, the proposed scheme is motivated by [74]. However, while AP is approximated by using sparse sample points in [74], AP is approximated in a principled manner by using mathematical derivations around discontinuous points in the proposed scheme, to be presented in Section 6.2.

A weighted sum of P_{MD} and P_{FA} at a target error ratio was originally suggested by TRECVID for the MED tasks [62]. In particular, the task is evaluated by examining an operating point at the ratio of $P_{MD} : P_{FA} = 12.5 : 1$. While optimizing the metric has not yet been actively studied, it is closely related to the multi-objective programming (MOP)

scheme [77]. In the MOP scheme, a composition of two objective functions (first, the maximum likelihood of the model parameters from the in-domain data and, second, an appropriate representation of prior information obtained from a general purpose corpus) is considered and explicitly optimized for an application of the language model. The problem with multiple constraints could be solved by incorporating a Lagrange multiplier [78] into the objective function to be optimized for each constraint. The presented work related to optimizing a weighted sum of P_{MD} and P_{FA} at a target error ratio is motivated by this learning scheme, and is discussed in detail in Section 6.3.

CHAPTER 3

FEATURE CONSTRUCTION AND BASE CLASSIFIER EVALUATION

For successful multimedia event detection and recounting, this research makes use of a large set of audio/visual features to facility the ability of capturing salient information across diverse event classes. Feature types used in this research are briefly reported in [79]. In this chapter, these features are studied in detail. Furthermore, additionally discussing the design of base classifiers and investigating various kernel types. Many feature descriptors used in this work have been proposed in previous research; however, there still exist many open issues in terms of how to efficiently use these descriptors, especially for representing unstructured consumer videos. The presented feature representation schemes and experimental results will provide insight into such issues, acquired through this research. It is also noted that the features discussed in this chapter are used across all experiments reported in the following chapters.

3.1 Visual Features

In this work, a set of low-level visual features is used. They are mostly quantized by a codebook-based method. In particular, for the purpose of evaluating features and base classifier performance, this chapter uses a single clip-level histogram representation that is based on bag-of-words (BoW) models. In other words, feature descriptors are extracted from image/video patches in a training corpus, collected, and quantized to codewords that represents a corresponding visual feature. Then, an entire video clip is assumed to be a single instance and is represented as a histogram of the constructed codewords for each feature.

The list of low-level visual features used in this research includes 3-dimensional histogram of gradients HoG3D [80], GIST [36], color scale invariant feature transform (SIFT)

[81], independent subspace analysis (ISA) [82], a transformed color histogram (TCH) [81], and a set of visual features from [83] (called SUN09 in this work), including a histogram of gradients (HoG), a geometry texton histogram (GTH), a self-similarity measure, dense/sparse SIFT, local binary patterns (LBP), and a tiny image. They are constructed in a clip-level feature vector as the following schemes:

HoG3D: HoG3D [80] is a spatio-termpoal variant of a popular histogram of gradients (HoG) descriptor, and additionally captures motion information beyond the standard HoG. The rationale for including HoG3D features is to incorporate low-level motion and appearance signals into the system. The raw 300-dimensional HoG3D features are densely computed from videos at every 5th frames where the samples are drawn from resized videos for consideration of speed and storage. Each video is rescaled such that its largest dimension (height or width) becomes 160 pixels. Once HoG3D samples are collected, K-means clustering is employed to create a codebook of 1000 words from a random subset of samples from the training data only. Finally, an average histogram is built for every video clip to form an HoG3D bag-of-words (BoW) descriptor.

GIST: In order to exploit correlations between event types and scenes where events take place, the GIST feature [36] is incorporated. GIST features represent an image's content in particular spatial frequency bands, and has been shown to provide discrimination between different types of environments such as natural versus man-made, open (i.e. outdoor) versus closed (indoor), and the like. GIST features are extracted at every 10th frame of video clips. Base classifiers are trained using per-frame features. Finally, the scores across frames of clips are averaged to provide a single score for a clip, constituting a clip-level base classifier.

Color SIFT/TCH: Color SIFT and TCH, which are efficient at capturing color information in a video clip, are extracted by [81]. For both feature types a spatial pyramid histogram is applied to construct a feature vector. In particular, 1 global histogram and 3 spatial histograms pooled from the top, middle and bottom local sections are concatenated. For all histograms, 4,096 codewords are used.

Table 2. Properties of low-level visual features

	Property							
Feature	C	T	G	M	B	U	P	S
HoG3D	x	x	o	o	o	x	o	x
Gist	x	o	x	x	x	x	x	o
color SIFT	o	x	o	x	o	x	o	x
ISA	x	x	x	o	o	o	x	o
TCH	o	x	x	x	o	x	o	x
HOG*	x	x	o	x	o	x	o	x
GTH*	x	o	x	x	o	x	o	x
self-similarity*	x	x	x	x	x	x	o	x
dense SIFT*	x	x	o	x	o	x	o	x
sparse SIFT*	x	x	o	x	o	x	o	x
LBP*	x	o	x	x	o	x	o	x
tiny image*	o	x	x	x	x	x	x	o

*SUN09 features

ISA: Hierarchical spatio-temporal information can be captured by ISA. As originally proposed in [82], features are directly learned from video data in an unsupervised learning manner, unlike hand-designed local features, such as SIFT or HoG. The extracted features are clustered into 3,000 words, and a video is represented as a clip-level BoW feature.

SUN09: Various low-level features including HoG, GTH, dense/sparse SIFT, a self-similarity measure, LBP, and a tiny image, are extracted by using the code provided by [83]. The feature extraction is very slow and is defined for images instead of videos. Therefore, these features are extracted on down-sampled frames at a rate of 4 seconds per frame. The same codebook provided by [83] is used to form clip-level histograms (or spatial pyramid features for some features). The histograms are L_1-normalized to form the final frame-level features (each feature is treated separately). Then, the final clip-level features are computed by taking the component-wise average of the frame-level features.

In Table 2, various types of visual information introduced by each low-level feature are summarized. The visual information considered includes whether it involves color (C), texture (T), gradient (G), temporal (M), BoW (B), unsupervised learning (U), patch-based (P), and entire image scenes (S). As we can observe, each low-level feature can contain

unique visual information. For example, GIST performs well especially in capturing textures and image scene information in the frames of a video, while color SIFT and TCH provide color information in a video clip. The usefulness of features for multimedia event detection can vary according to the type of a target event class, while it does not necessarily follow intuition. The results of individual low-level features are reported in Section 3.4.

In addition to low-level visual features, high-level visual semantic information is also utilized by Object Bank (OB) features [84]. Compared to traditional scene-level concepts such as LSCOM [10, 11], OB features provide a semantic and descriptive understanding of visual scenes at the object level. The OB framework is arbitrarily expandable and open, which means that, regardless of the object classes, every object detector is trained in a generic framework and can be easily plugged into the main system. The current OB implementation incorporates detectors for 177 object classes and is one of the main large-scale object recognition system publicly available. These object classes are independent of event categories.

In this work, spatial pyramid layout information, originally suggested by [84], is discarded, because the variation of object locations within unconstrained videos is not regularized, and the resulting lower-dimensional representation helps the generalization capability during classifier training. In particular, first, the entire array of object detectors is applied to images at various scales. Then, their responses are recorded along with spatial layout information to form high-dimensional scene appearance descriptors. ObjectBank is the most computationally intensive among the visual features. Accordingly, to identify a set of key frames from each video clip, a change detection technique based on color histograms is applied, which takes approximately 100 hours on its own. Then, ObjectBank features are computed only on those key frames. Because ObjectBank is applied on a per-image basis, multiple ObjectBank features are agglomerated to produce a clip-level descriptor across frames. In this work, using max-pooling and average-pooling has been observed to provide good performance; hence, both max and average responses from each ObjectBank feature

dimension across key frames are recorded at the clip-level feature.

3.2 Audio Features

For audio features, first, a low-level audio feature is considered to capture the general audio information of a video. In particular, MFCC features are represented in a BoW feature. At every 10ms with a 25ms frame size, 32-dimensional MFCCs are extracted. Then, the frame-level features are quantized based on a codebook with a 1K size, using hard-assignment.

This research also includes developing a new audio feature that involves high-level audio semantics for MED tasks. A conventional way of exploiting audio semantics for MED is to use a set of pre-defined audio concepts [85, 24]. However, using a fixed set of audio concepts to perform event detection might not be suitable because consumer-level videos tend to be unconstrained and unstructured. As such, there exists a wider range of variability in audio signals. Alternatively, in this research, acoustic segment models (ASMs) [86] to understand a broader range of mid-level audio semantics by capturing diverse temporal structures within low-level audio signals are developed. ASMs build upon previous work such as fundamental speech sound units for speech recognition [87], which have been applied to music genre classification [88] and speaker recognition [89]. This approach is the first study of ASMs to MED by building bottom-up acoustic semantic words. In particular, unlike previous work that exploits temporal acoustic structures in particular domains, e.g., speech or music, the developed ASMs provide an extended framework for generic audio sound types.

In particular, ASMs are modeled as 3-state HMMs. They are trained with a set of "representative" audio segments for given multimedia event classes; in particular, 8 initial segments are manually chosen from an event class. For example, initial segments for *Birthday party* include singing a birthday song, cheering, laughing, and clapping, while those for *Getting a vehicle unstuck* include tire spinning, motor, and street noise. Then,

Viterbi decoding and Baum-Welch estimation are iteratively conducted in order to refine the models until they converge. The typical length of decoded segments is 100–200ms. Once ASMs are obtained, each audio clip is transformed into a BoW vector by considering the N-best Viterbi sequences with unigram and bigram statistics.

Once ASMs are obtained, each audio clip in multimedia material is transformed into a feature vector, treating each ASM as a basis of a vector space. To this end, ASM n-grams are calculated, obtaining *bag-of-sounds* vectors similar to BoW vectors in information retrieval. In particular, N-best Viterbi sequences are considered; then, the number of occurrences of each ASM is counted in the Viterbi sequences (*unigram*). In addition, the number of co-occurrences of two adjacent ASMs (*bigram*) is considered. In this work, co-occurrence counts not only for adjacent ASMs, but also for any pairs of ASMs located within a certain window are evaluated. More detail of modeling and learning of the proposed audio feature representation by ASMs can be found in [86].

3.3 Evaluation of Non-linear Kernels for Learning Base classifiers

In recent research regarding MED [21, 22, 23, 24, 25], it has been reported that using non-linear kernels can significantly improve the quality of MED systems beyond using a linear kernel. Most features used in this research are based on BoW representation or one of its variants. Therefore, the following 4 non-linear kernels, which can be applied to histogram-based features, are considered for evaluation: the chi-squared kernel (CSK), the Bhattacharyya kernel (BK) [90], the histogram intersection kernel (HIK) [91], and the negative geodesic distance kernel (NGDK) [92]. The formulations of the kernels are as follows:

$$k_{CSK}(x, y) = 1 - \sum_{i=1}^{D} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}, \tag{1}$$

$$k_{BK}(x, y) = \sum_{i=1}^{D} \sqrt{x_i y_i}, \tag{2}$$

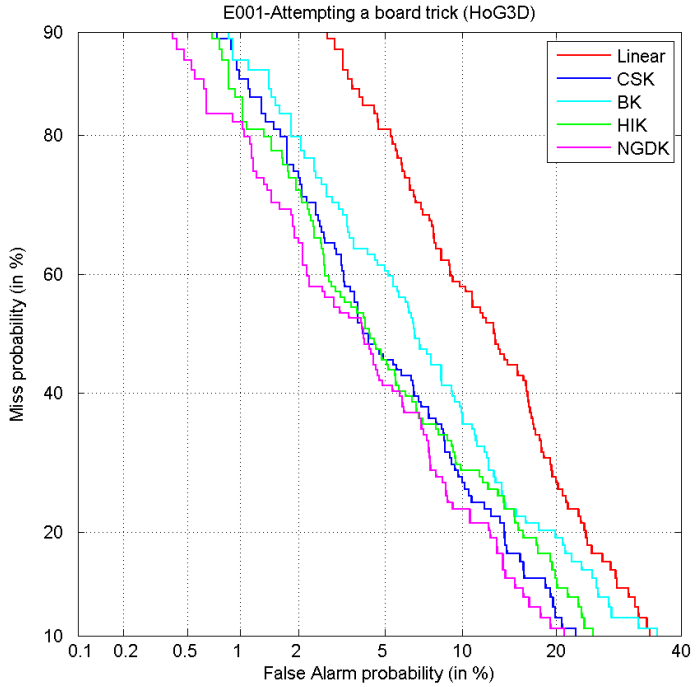
$$k_{HIK}(x, y) = \sum_{i=1}^{D} \min(x_i, y_i), \tag{3}$$

$$k_{NGDK}(x, y) = -2 \arccos\left(\sum_{i=1}^{D} \sqrt{\frac{x_i}{|x|}\frac{y_i}{|y|}}\right). \tag{4}$$

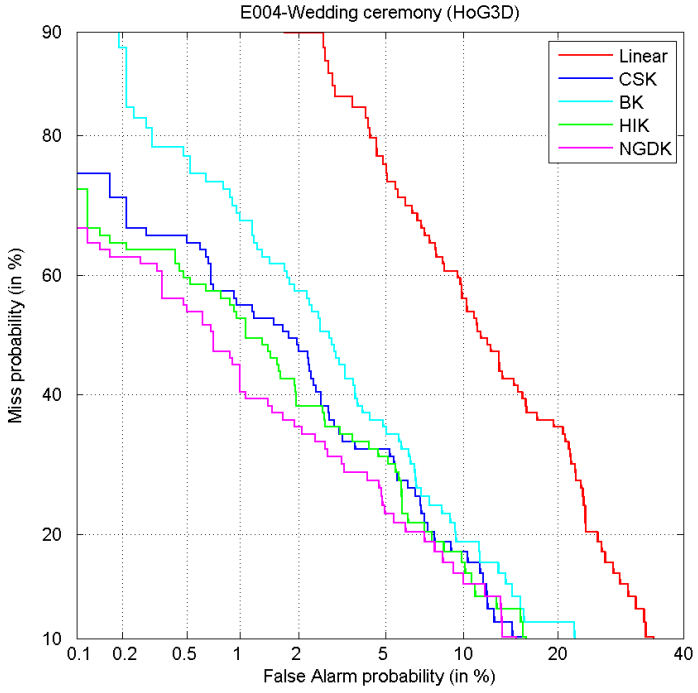
When base classifiers for each feature type are learned, this research uses kernelization techniques. In particular, base classifiers are learned and tested by a discriminative model, e.g., SVMs, in a kernel space. The performance of kernel types is illustrated in Figure 2 by using a detection error tradeoff (DET) curve [93] for the two exemplar classes (it is noted that similar performance has been shown in other event classes). In a DET plot, the bottom-left location of a lined curve implies the superiority of a measured system, and it is recommended as one of the metrics to evaluate multimedia event detection systems by TRECVID [62]. We can observe that all non-linear kernels significantly outperform the linear kernel, implying that using non-linear kernels is crucial to successfully perform multimedia event detection. Among non-linear kernel types, NGDK shows the best performance, followed by HIK. These two kernels are simple to apply and can be computed within a reasonable amount of time. Therefore, in this research, NGDK and HIK are mainly considered across the remaining experiments.

3.4 Evaluation of Feature Types

The base classifier performance over the 10 TRECVID MED '11 event classes is summarized in Table 3, evaluated by average precision (AP). To additionally evaluate the performance of kernel types, different kernel types are applied to a feature, where the kernel types are available. It is noted that some kernel types are not available based on the feature characteristics, e.g., NGDK cannot be applied to GIST, since this feature type involves negative values. The mean AP (mAP) over the 10 event classes is also illustrated in Figure 3, which clearly demonstrates overall feature performance and the effect of kernel types. We can observe that overall, the SUN09 feature with MKL shows the best performance. This



(a)



(b)

Figure 2. Comparison of kernel types in DET curves by using HoG3D on (a) E001-*Attempting a board trick* and (b) E004-*Wedding ceremony*: we can clearly observe that non-linear kernels significantly outperforms linear kernel (red), while NGDK (magenta) shows the best performance followed by HIK (green).

Table 3. Comparison of features by different base classifiers, evaluated in AP (%) for the 10 event classes. It is noted that the event-wise best AP is marked in bold. The best performance is achieved by different types of features for different event classes.

Base classifier	E006	E007	E008	E009	E010	E011	E012	E013	E014	E015	mAP
GIST HIK	3.3	1.7	22.4	4.5	2.3	5.7	10.7	3.3	5.6	2.2	6.2
HoG3D Linear	3.3	1.2	11.9	3.8	1.1	1.8	10.0	4.1	2.2	1.3	4.1
HoG3D HIK	12.6	7.4	42.9	11.9	5.0	6.6	11.6	28.8	30.1	10.7	16.8
HoG3D NGD	12.9	11.1	42.8	17.8	6.2	10.4	23.9	**30.5**	31.5	10.5	19.8
ISA HIK	9.3	10.6	38.8	14.6	4.0	6.9	13.7	27.0	15.6	4.3	14.5
CSIFT Linear	2.4	2.0	14.6	4.8	1.1	2.0	11.1	1.6	5.8	1.4	4.7
CSIFT HIK	13.8	18.7	47.3	19.4	6.7	6.6	**26.2**	25.2	33.2	13.1	21.0
CSIFT NGD	9.8	23.8	47.5	19.8	6.7	**13.7**	25.7	27.2	36.8	13.3	22.4
TCH Linear	1.8	1.7	8.3	4.0	0.8	1.2	9.0	1.4	3.6	2.3	3.4
TCH HIK	7.8	12.7	36.1	17.8	5.1	8.9	25.7	16.0	31.4	11.8	17.3
TCH NGDK	8.7	13.8	35.2	17.2	6.2	9.8	24.3	19.5	34.3	11.3	18.0
OB-Avg Linear	3.7	3.1	19.9	10.7	1.6	3.4	7.9	2.1	2.9	3.8	5.9
OB-Max Linear	7.2	8.2	13.1	9.5	1.9	6.6	10.0	4.0	7.1	3.5	7.1
OB-Avg HIK	7.9	10.9	32.7	19.8	6.2	7.5	16.6	15.4	18.3	10.0	14.5
OB-Max HIK	9.5	13.5	31.0	17.0	5.6	9.8	18.4	10.7	25.7	7.7	14.9
SUN09 MKL	15.7	**29.8**	**50.4**	**25.7**	**15.2**	13.3	23.3	25.3	32.2	**13.9**	**24.5**
MFCCs Linear	7.9	1.9	4.0	1.4	0.7	1.2	4.0	1.5	13.5	1.7	3.8
MFCCs HIK	17.5	3.1	14.0	4.4	1.4	1.5	7.2	3.1	22.4	6.3	8.1
MFCCs NGDK	**24.0**	3.6	17.3	7.9	1.6	6.4	9.8	2.3	42.9	13.6	12.9
ASM HIK	14.3	4.5	10.5	5.7	1.6	5.8	10.8	1.9	31.1	7.7	9.4
ASM NGDK	23.6	4.9	16.4	7.1	1.6	7.1	13.2	2.4	**43.6**	13.8	13.4

might be because SUN09 incorporates multiple visual features with various granularities.

By comparing the different types of features and kernels, we can draw an interesting observation. First, it has been observed that event detection performance is improved significantly by non-linear kernels versus linear kernels across all feature types. Between HIK and NGDK, NGDK shows consistent improvement compared to HIK. By this observation, for the remaining experiments of this research, HIK and NGDK are mainly considered in modeling base classifiers, while NGDK is preferred only if it is available.

Second, from the base classifier results in Table 3, we can also observe that the best performance can be achieved by different feature types for different event classes. For example, for E006-*birthday party* and E014-*Repairing an appliance*, the audio features (MFCCs and ASM) show fairly strong performance, while the remaining visual features

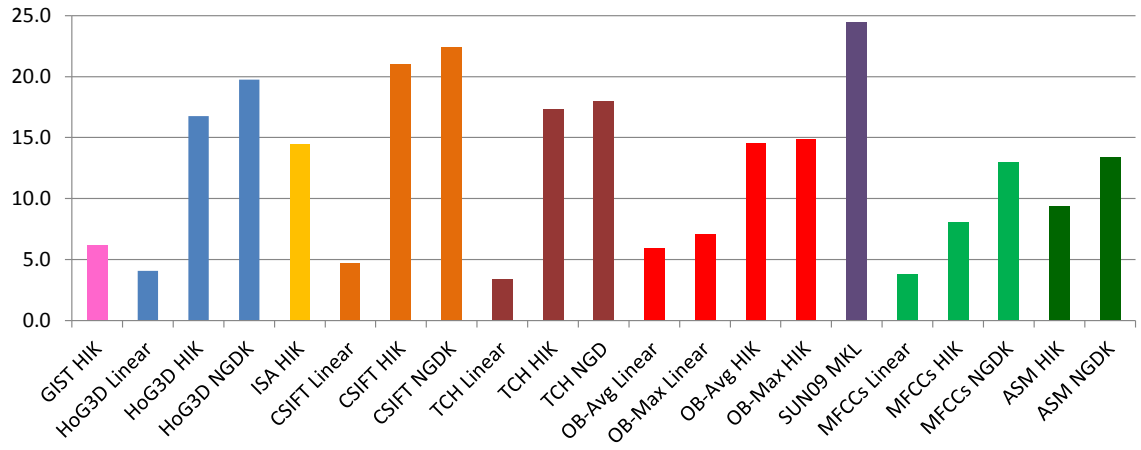


Figure 3. Comparison of features by different base classifiers, evaluated mAP (%) over the 10 event classes. Among kernel types, NGDK outperforms linear kernel and HIK. Overall, SUN09 MKL shows the best performance.

show better performance for the other event classes. It is not surprising since for E006-*birthday party*, cheering by guests or a birthday song can be strong evidence, and there exists large variation in visual information, e.g., a video can be recorded in a dark room or at party place. For E014-*Repairing an appliance*, the most discriminative cue for detecting the event class is human speech since most samples are instructional video clips. Furthermore, it is found that most features are complementary. In other words, when various features are combined in a fusion method, performance for multimedia event detection consistently improves. We will discuss this idea further in Chapter 6.

In addition, it is observed that low-level features show surprisingly competitive performance compared to high-level features. In terms of quantitative performance, there does not seem to be any significant advantage of high-level visual or audio features (OB and ASM).

CHAPTER 4

SCENE CONCEPT ANALYSIS FOR SPARSE EVIDENCE

Detection of complex events on unconstrained real-world videos (e.g., YouTube) is a challenging problem. Most complex events (e.g., *birthday party* and *board trick)* exhibit large within-category variations, and videos frequently consist of multiple segments exhibiting different and evolving contents that include not only a mixture of contents closely related to events, but also temporal clutters such as caption screens or irrelevant contents arbitrarily stitched-in by users.

For example, in Figure 4, the visual content of segments in a video clip labeled with the *Attempting a Board Trick* class is illustrated. The first segment (seg_00) is rendered in gray-scale, and accordingly, color information that might be useful for detecting sky or snow regions is lost. The second, third, and last segments (seg_01, seg_02, and seg_13, respectively) consist of black-screen scenes, which are possibly not distinctive for the *Attempting a Board Trick* class. Actual board-riding scenes appear at the remaining segments (seg_03~seg_12); however, visual content still varies among these video segments, in both temporal and spatial aspects, e.g., the visual stream is paused at seg_08 and seg_09, while only audio is being played. It is clear that the discriminative power significantly differs among video segments with different scenes. However, addressing such sparseness of evidence with a conventional approach, which uses a single feature representation, e.g., a video-level BoW representation, is challenging.

Many reported successful retrieval systems (e.g., [4, 94]) for unconstrained videos share the common idea of constructing clip-level representations via average-based global pooling. For each feature type, a bag-of-words (BoW) descriptor (or variation) is built per video by pooling across the entire video. These globally pooled features work well, although the fact that these methods do not exploit detailed segment information leaves room for further research, which recent efforts have begun to address, such as [35, 34].

Figure 4. The scene variance of video segments in a clip.

In this work, a multi-way local pooling (MLP) approach is presented, which uses detailed segment-level information and boosts performance beyond the globally pooled descriptors. The overall scheme is illustrated in Figure5. The approach builds multiple descriptors per video, where each descriptor is designed to relate to one of the pre-built scene concepts. These scene concepts can be understood as rough themes interchangeably appearing as segments in videos. From an input video, a separate descriptor is built per scene concept by accumulating features from segments that are local (or similar) to the represented scene concept. The rationale behind the MLP strategy is partly inspired by the recently introduced theory of local pooling [95], which shows that pooling features similar in multi-dimensional input space separately improve representational power, and classification accuracy. In addition, it has been observed that the frequency of segment-to-concept assignments provides a unique signature indicating the importance of each scene concept in describing a video sample. Accordingly, the proposed approach intentionally avoids normalization on each descriptor, which is in contrast to [95, 35].

Consider the example video of *board trick* in Figure 5, which consists of title screens at both ends and actual snowboarding segments in the middle. First, there are a set of scene concepts discovered by clustering segment-level features[1], which include concepts such as

[1]For these results, (uncolored) HoG3D feature [80] is used.

caption/titles, a moving object on a smooth background, and cube-2shaped large objects[2]. It is worth noting that the proposed method is general and can be applied to various audio-visual features developed for multimedia videos. Accordingly, scene concepts with motion or audio patterns can be discovered as well, depending on the characteristics of the underlying features. For example, it can be seen in Figure 5 that, not only image-based concepts but also motion-related concepts are discovered. Then, each per-concept descriptor is built by pooling features from different parts of the input video using soft-assignment, based on similarity between segments and scene concepts.

For classification, this work adopts the intersection kernel (IK) SVMs [91] to build classifiers. In particular, the kernel between a pair of videos is computed by combining per-concept kernels computed for every scene concept. It is important to note that the per-concept kernel values on frequently assigned kernels tend to be high and vice-versa, due to the captured frequency information. Accordingly, if certain scene concepts are poorly represented in exemplar videos, they will contribute to kernel values in a limited way. In addition, this work explores an alternative strategy to combine kernels using multiple kernel learning (MKL), e.g., [96]. In essence, it is plausible that certain scene concepts are more discriminative, even though they are rarely represented in exemplar videos, or vice-versa. The use of MKL provides an opportunity to learn discriminative weights for kernel combination.

The idea of multi-way pooling has been developed in [95], but it is only applied to low-level raw visual feature descriptors for image recognition. This work is extended to video recognition at a higher-level granularity of segments. Recent work that characterizes videos at the segment-level includes [33], [34], and [35]. To represent complex activities, [33] identifies distinctive temporal segments (i.e., sub-actions) along with the temporal structures between them. Although the temporal structures allow certain flexibility, [33] is still most suitable to videos with fairly regularized structures (e.g., Olympic dataset).

[2]These scene concepts are manually named *a posteriori*.

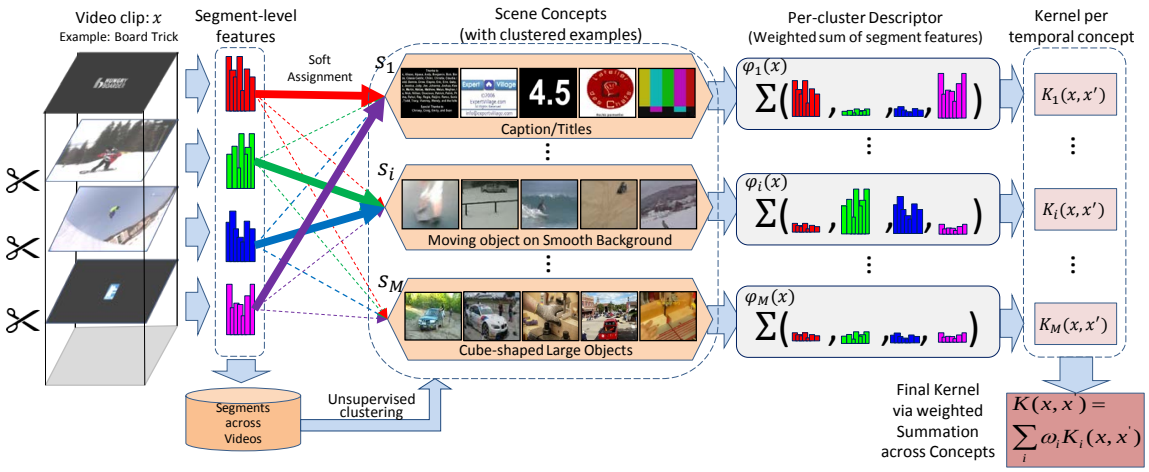


Figure 5. Illustration of the Proposed Representation: a video clip consists of multiple segments. Each segment-level feature is pooled multi-way into different descriptors based on their similarity and the corresponding pre-constructed scene concepts. Then, kernelization is separately applied per descriptor. The final kernel combines multiple kernels and provides an improved discriminant power.

In [34], a discriminative recursive hidden segmental Markov model is proposed to cope better with less regularized temporal structures in consumer videos. However, the potential challenge for [34] is that the model is associated with a large number of parameters, which requires a large training dataset. In the closely related work of [35], features from images in videos are pooled into different scene clusters, guided by the secondary GIST [36] feature. Although using a secondary feature might be necessary to incorporate extremely sparse feature types (e.g., sparse SIFT), it renders this method applicable to image-based features only, and exploring a unified feature pooling method (without a secondary feature) for more general feature types is necessary. In contrast, the proposed method is simpler and more general because it is more widely applicable to diverse audio-visual features beyond image-based ones, and can utilize audio/temporal concepts. In contrast to [35], the proposed approach also explores MKL variations to combine kernels across different scene concepts.

4.1 Multi-way Local Pooling

The multi-way local pooling (MLP) method uses multiple descriptors instead of a single descriptor given a feature type for a video clip, and then attempts to improve discriminant

power using kernelization techniques. The key idea is to quantify and utilize similarities between two video samples with respect to various scene concepts, especially in unstructured consumer video data, where it is difficult to apply conventional temporal models, e.g., HMMs.

The overall scheme is illustrated in Figure 5. First, we divide a video clip into video segments and represent each segment with a given feature type. Then, every video segment is soft-assigned to scene concepts. These scene concepts are pre-constructed by unsupervised clustering from all segment-level feature descriptors in training data, and thus represent broad categories covering entire training corpus, e.g., caption/title, moving object on smooth background, or cube-shaped large object. A large assignment value of a video segment to an existing concept indicates that they are highly correlated, and vice-versa. Soft-assignment is important because it can substantially alleviate the arbitrary space partitioning built by unsupervised clustering of segments. Using this soft-assignment, segment-level feature descriptors in a video are combined with different weights for each scene concept. In other words, if we have M scene concepts, M video-level feature descriptors are constructed in a way that a highly correlated video segment to a corresponding scene concept contributes more. In a casual sense, the newly constructed video-level feature descriptors can be considered to be projections of a video clip toward corresponding scene concepts. After multiple descriptors are constructed with respect to scene concepts, kernelization is separately applied. Thus, when a similarity kernel is used, e.g., IK, a kernel function measures similarity of video samples with respect to each corresponding scene concept. Finally, multiple kernels are combined to a final kernel to provide an improved discriminant power for video recognition. For brevity, the detailed derivations below are based on BoW features, although it can be generalized to other representations.

In detail, let $x = \{x^i | \, x^i \in R^D, \, 1 \leq i \leq n\}$ be a training video sample, where x^i is a D-dimensional BoW representation for the i-th segment, and n is the number of total segments in a video sample x. It is assumed that scene concepts are already available by collecting all

of the video segments from the training corpus and clustering them in the D-dimensional feature space by an unsupervised k-means scheme. Then, the proposed approach uses centroids of the clusters as scene concepts that compactly describe the segment types in the video. Let $S = \{s_j | \, s_j \in R^D, \, 1 \leq j \leq M\}$ be a set of M scene concepts, which are represented as D-dimensional vectors. Each D-dimensional feature descriptor of a video x with respect to the j-th scene concept is formulated as a weighted-BoW representation $\varphi_j(x)$ computed across the entire video segments $\{x^1, x^2, \cdots, x^n\}$, with corresponding soft-assignment weights as

$$\varphi_j(x) = \frac{1}{n} \sum_{i=1}^{n} \omega_j(S, \, x^i) \cdot x^i, \tag{5}$$

where n is the number of video segments in a video sample x, and $\omega_j(\cdot)$ is a soft-weight assignment function between a corresponding scene concept and a video segment. While the choice for the soft-weight assignment is flexible, we have adopted the following variant of the Gaussian function, which has shown superior performance across our experiments with diverse features:

$$\omega_j\left(S, \, x^i\right) = \exp\left[-\frac{\left\{d\left(s_j, x^i\right)\right\}^2}{\alpha}\right], \tag{6}$$

where α is a positive parameter that controls the sensitivity on the distance $d\,(\cdot)$ between a centroid and a sample point. For a distance measure, the negative geodesic distance (NGD) that provides an effective distance measure on BoW features [92] is used, defined as the following:

$$d\left(s_j, x^i\right) = -2 \arccos\left(\sum_{k=1}^{D} \sqrt{\frac{s_{j,k}}{|s_j|} \frac{x_k^i}{|x^i|}}\right). \tag{7}$$

The varying weights of video segments to a scene concept make the contributions of the video segments differ in constructing the weighted-BoW representation for a specific scene concept. For example, in Figure 6, the soft-assignment values for the video segments in Figure 4 to 50 scene concepts are illustrated. Each column indicates a video segment with a 50-dimensional assignment vector. The brighter the element is of an assignment vector,

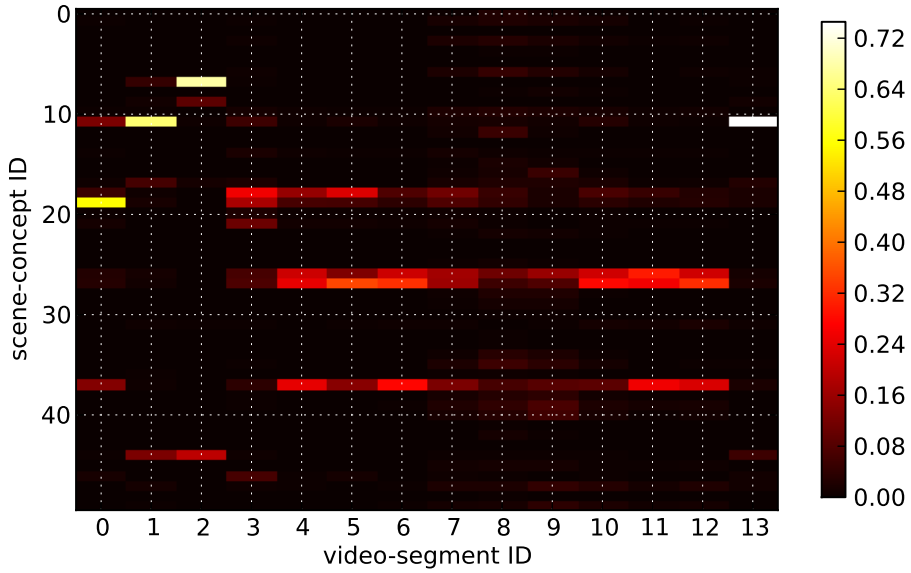


Figure 6. The soft-assignment vectors to 50 scene concepts for the video segments in Figure 4. The assigned values to a scene concept varies across the video segments.

the more a video segment contributes to a corresponding scene concept, and vice-versa. In a casual manner, the weighted-BoW representation $\varphi_j(x)$ can be considered as a projection of a video sample x to the j-th scene concept space.

4.1.1 Comparison to SAP

According to the SAP method proposed in [35], one way to utilize such multiple feature representations from scene concepts is concatenating the features into a large $(n \times D)$-dimensional feature vector $\psi(x)$, formulated as

$$\psi(x) = \{\varphi_1^{'}(x), \varphi_2^{'}(x), \cdots, \varphi_n^{'}(x)\}, \tag{8}$$

where $\varphi_i^{'}(x) = \varphi_i(x)/\|\varphi_i(x)\|$ is a variant of a feature representation for the i-th scene concept $\varphi_i(x)$ by using L_1-normalization. However, considering that most state-of-the-art methods in multimedia retrieval incorporate a large number of code words, the dimension of the concatenated feature vectors can be extremely large. Therefore, learning a detection model on the concatenated feature vectors may not be feasible for a large-scale dataset because of both computational and memory-wise costs. Furthermore, since feature representations for scene concepts are L_1-normalized in a segment-wise manner, we may not take full

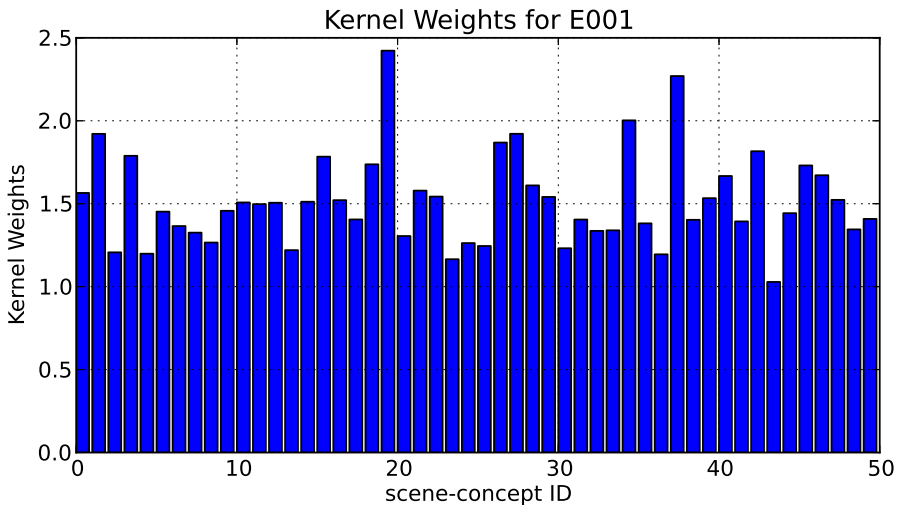


Figure 7. The discriminative weights for 50 scene concepts learned by MKL with L_2-regularization.

advantages of the effects from soft-assignment. For example, consider a video clip that is loosely correlated to a specific scene concept s_i across entire video segments. In this case, since all segments in the video clip will have small soft-assignment values to the scene concept s_i, $\|\varphi_i(x)\|$ should small, and accordingly, the effects by the feature representation of $\varphi_i(x)$ for the scene concept s_i become trivial compared those from other strongly correlated scene concepts. However, by segment-concept-wise L_1-normalization, the effects from all scene concepts become comparable, raising a question that advantages of multi-way pooling are still fully implemented in the system.

4.1.2 Feature Combination by Kernelization

In contrast to constructing the large vector by concatenating multiple feature representations for scene concepts, the proposed method performs combining the multiple feature representations by kernelization techniques without normalization. For a kernel type, the popular IK is selected, along with the desirable property of not involving normalization during kernel computation. The usefulness of IK is presented in Section 3.3 (IK is a slight variant of HIK, which assumes that a feature vector is not L_1-normalized). It is noted that the constructed BoW feature $\varphi_j(x)$ for a scene concept s_j is not L_1-normalized. In this

way, the IK $K_j(x, x')$ between a video x and another video x' is determined by not only the similarity between the distribution of $\varphi_j(x)$ and $\varphi_j(x')$, but also $\|\varphi_j(x)\|$ and $\|\varphi_j(x')\|$, which reflect their assignment frequency and correlation to a scene concept s_j. In other words, even if $\varphi_j(x)$ and $\varphi_j(x')$ show similar distributions, $K_j(x, x')$ might be small if one or both samples are loosely correlated to a scene concept s_j. Then, a final kernel $K(x, x')$ is constructed as a linear combination of the multiple kernels constructed for multiple scene concepts as following:

$$K\left(x, x^{'}\right) = \sum_{j=1}^{n} \beta_j K_j\left(x, x^{'}\right), \ \forall j, \beta_j \geq 0, \tag{9}$$

where β_j is a non-negative weight on the j-th kernel. We applied both equal weights and those learned by MKL, and their results are reported in the following section. For example, in Figure 7, the learned weights for 50 scene concepts by MKL with L_2-regularization are illustrated. It can be observed that the learned weights significantly vary.

4.2 Experiments and Analysis

The proposed method was evaluated on TRECVID 2011 MED corpus [20], which is a challenging large-scale consumer video dataset. The dataset provides an excellent test-bed for a real-world unconstrained video retrieval problem. It consists of 13K training and 32K test samples with 10 annotated test event classes. The number of positive and negative samples is highly imbalanced; e.g., there are only about 150 positive samples for each class in both training/test sets. For each event class, we trained SVMs in a one-vs-all manner across our experiments and report average precisions (APs) or mean AP (mAP) across all ten events as metrics. For training protocol, the approach in a recent study [34] was followed for fair comparison, across experiments.

The proposed approach is extensively compared with other strong baseline methods and/or state-of-the-art methods for both visual and audio features. For the results, (visual) HoG3D [80] with 1,000 codewords and (audio) MFCCs with 1,024 codewords are used,

Table 4. Retrieval results w.r.t. varying number of scene concepts on HoG3D, in mAP (%).

# of SCs	1	20	40	60	80	100
mAP	7.00	9.43	9.75	9.74	9.75	9.72

respectively. For the length of video segments, a clip was regularly divided into fixed 2-second-length segments, which is found to work well across feature types. Although the use of variable-length segments based on techniques such as shot detection may be interesting, it is beyond the scope of this study, and we could still verify the benefits of the proposed method with the aforementioned video segments. The parameter α in Eq. (6) was set to be $\alpha = 0.3\pi^2$, which is found through cross validation.

4.2.1 Robustness Against the Number of Scene Concepts

The first result analyzes the sensitivity of the proposed method against the number of scene concepts (SCs). Table 4 summarizes the mAP results across all ten classes with respect to varying number of scene concepts on HoG3D. It is noted that using only one scene concept is equivalent to using a conventional kernelized SVM (KSVM). It can be observed that significant improvement (relatively 39.3%) can be achieved as the number of scene concepts is increased to 40. Beyond 40, the performance stabilizes, showing the desirable property that the proposed approach is relatively immune to over-fitting even when large number of SCs are used, which can be credited to the proposed distance and soft-weighting schemes. Although the number of optimal scene concepts may differ by features types, event types, and datasets, it is found that 30–50 SCs are generally sufficient to acquire the benefits of the proposed method. For the remainder of the experimental results, 40 scene concepts have been used.

4.2.2 Comparison with the State-of-the-art Methods

The main experimental results on 32K test samples using visual HoG3D features are summarized in Table 5. The performance of Chance (i.e., random) is very low due to the imbalance between positive and negative samples. For MLP approaches, two variations

Table 5. Comparison in AP(%) among the baseline systems including state-of-the-arts, and the proposed MLP methods. For each row, the best result is marked in bold. Overall, both MLP-EQ and MLP-MKL consistently outperformed baselines, showing notable improvement in mAP (illustrated in Fig 8 for more clarity).

event ID	Chance	LSVM	Niebles	Tang	LSAP	KSVM	KSAP	KSAP-H	MLP-EQ	MLP-MKL
E006	0.54	1.97	2.25	4.38	3.95	6.08	4.24	4.73	6.34	**6.74**
E007	0.35	1.25	0.76	0.92	2.88	2.87	2.86	2.26	**3.01**	2.98
E008	0.42	6.48	8.30	15.29	17.31	20.75	22.33	22.99	**31.16**	30.87
E009	0.26	2.15	1.95	2.04	4.33	6.25	5.36	**7.61**	7.54	7.50
E010	0.25	0.81	0.74	0.74	1.31	1.43	1.14	1.34	2.11	**2.34**
E011	0.43	1.10	1.48	0.84	1.94	2.29	2.57	2.65	**4.07**	3.86
E012	0.58	5.83	2.65	4.03	7.43	8.44	7.08	8.7	10.63	**11.13**
E013	0.32	2.58	2.05	3.04	9.78	9.44	9.33	10.43	**15.57**	15.25
E014	0.27	1.18	4.39	10.88	5.25	10.00	9.79	11.89	14.81	**14.84**
E015	0.26	0.92	0.61	**5.48**	1.54	2.49	2.02	2.4	2.25	1.82
mAP	0.37	2.43	2.52	4.76	5.57	7.00	6.67	7.50	**9.75**	9.73

using equal kernel weights (MLP-EQ) and those learned by generalized MKL [96] (MLP-MKL) are reported. The compared approaches used in the experiments include linear SVM (LSVM), KSVM, Niebles [33], Tang [34], and linear/kernelized SAP (LSAP/KSAP) [35]. It is noted that, for direct comparisons, we reproduced the results of Niebles and Tang from [34] by using the same quantized features and training/test protocol, and also re-implemented LSAP/KSAP using the same GIST[36] feature as a secondary image feature and parameters suggested by the authors [35]. In addition, a variant of KSAP (denoted as KSAP-H) is also evaluated, in which scene clusters are constructed by using the same pooled feature (not a secondary image feature, suggested by [35]), i.e., HoG3D in this experiment. For KSVM, KSAP, and KSAP-H, the same IK used in the proposed approach is applied. Across all compared methods, the same HoG3D features have been used.

As shown in Table 5, the proposed approach consistently outperforms the compared systems for most event classes. In particular, the proposed MLP approaches show significant improvement of (relatively) 30% on average beyond KSAP-H, which is found to be the best baseline system. Such improvement was achieved by our soft-assignment scheme based on NGD distance and kernel combination without concept-wise normalization. I has

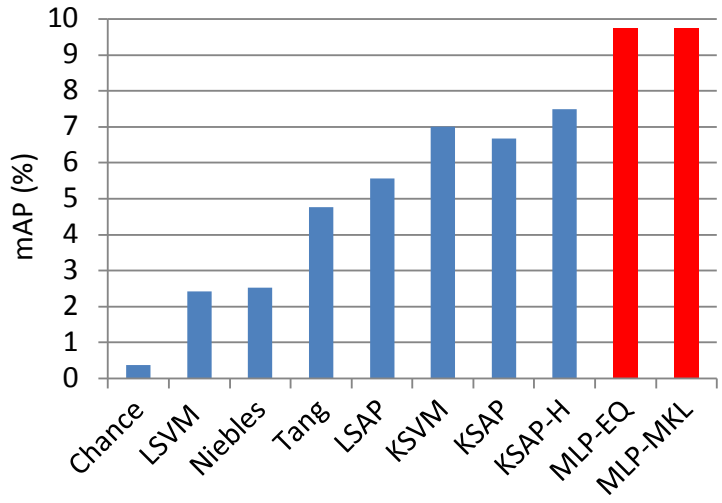


Figure 8. Comparison in mAP(%) among the baseline systems. The proposed methods, MLP-EQ and MLP-MKL, marked in red, show significant improvement over the various baseline systems, including the state-of-the-art methods.

also been observed that KSAP-H outperforms KSAP. This implies that the use of a consistent feature in constructing SCs can improve the quality of SCs, when compared to the use of a secondary image feature, especially for densely extracted feature types (it is noted that the HoG3D feature is densely extracted, while features used in [35] are sparsely extracted). In addition, it can be observed that KSVM outperformed the other latest methods without kernelization (Niebles, Tang, and LSAP), which suggests that the use of kernelization is one of the critical techniques for successful event detection.

In Table 6, the proposed method was also compared by using audio MFCC features against two baselines (LSVM and KSVM). Other baseline systems are not included because the non-image features can not be incorporated or they did not show better performance than KSVM. In this experiment, about 2% of test videos without audio was excluded from both training and test data, which makes the performance by Chance slightly different from Table 5. It is clear that the proposed MLP methods provide advantages in audio as well, showing on average 14.9% improvement (relative) over KSVM. These results show that the proposed framework is fairly general and can yield benefits across different feature modalities.

Table 6. Results in mAP(%) using MFCCs (MLP with audio features).

	Chance	LSVM	KSVM	MLP-EQ	MLP-MKL
E006	0.55	2.08	10.69	11.72	**12.01**
E007	0.35	0.79	1.82	**2.89**	2.78
E008	0.42	1.04	6.56	**7.19**	7.05
E009	0.26	0.54	1.33	2.29	**2.45**
E010	0.26	0.47	0.77	1.81	**1.98**
E011	0.40	0.85	1.59	2.12	**2.34**
E012	0.60	1.32	5.50	6.45	**6.67**
E013	0.32	1.22	2.50	2.87	**2.91**
E014	0.25	3.54	24.63	**25.70**	24.40
E015	0.23	0.64	4.38	5.21	**6.11**
mAP	0.36	1.25	5.98	6.83	**6.87**

Among the proposed methods, MLP-EQ and MLP-MKL showed comparable results across all of the event classes, with slight improvement by one or the other, depending on event classes. The surprising effectiveness of MLP-EQ can be attributed to the following reasons: (1) individual kernels constructed for corresponding scene concepts are already weighted by the assignment frequency, which seems to capture most discriminative information; and (2) when underlying features are constructed effectively with little redundancy, equally weighted kernels has been shown to have comparable performance to MKL [97]. These results suggest that, for time-sensitive applications, the use of MLP-EQ alone can be a good approach at the loss of minor accuracy for some classes.

4.2.3 Video Categorization by Scene Concept Weights

In addition, it has been found that an assignment vector toward scene concepts can be a robust and compact representation of videos, which can be used to measure similarity among video samples. For example, Figure 9 illustrates clustering video samples by scene concept assignment vectors given them. Each column in Figure 9 indicates an assignment vector of a video clip, which is generated by averaging scene concept assignments across all segments in the clip. More precisely, for a video x and a set of scene concepts S, a

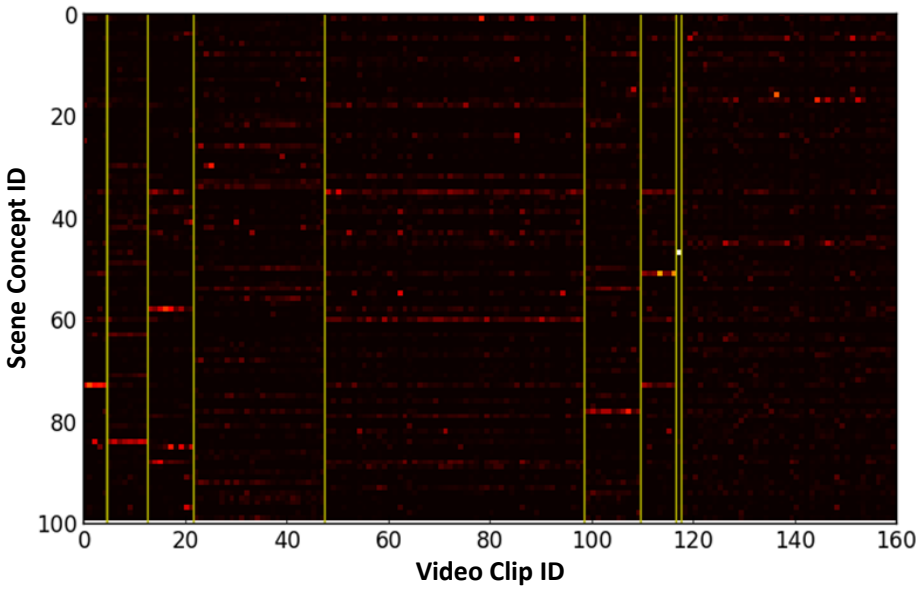


Figure 9. Assignment vectors regarding scene concepts of videos in E001-*Attempting a board trick*: we can clearly observe patterns in the vectors, and videos can be categorized, while they are all labeled as the same event class.

weight vector $\overline{\Omega}_{S,x}$ is formulated as

$$\overline{\Omega}_{S,x} = \frac{1}{n}\sum_{i=1}^{n}\overline{\omega}\left(S,\ x^i\right), \tag{10}$$

where x^i is the i-th segment of a video x; n is the number of video segments; and $\overline{\omega}\left(S,\ x^i\right)$ is a soft-weight vector, discussed in Eq. (6). The number of scene concepts used in this example is 100, where scene concepts are represented in row-wise. The red color (bright) in an assignment vector indicates strong correlation between a video clip and a corresponding scene concept. The video clips represented in Figure 9 are all labeled as '*Attempting a board trick*'; however, we can clearly observe some patterns among the assignment vector of them. The yellow lines indicate divisions of such patterns by k-means clustering (it is noted that video clips are ordered by categories for better presentation). For example, the first group of videos has strong correlation with the 73th scene concept, while the second group of videos shows fairly strong response to the 84th scene concept.

In Figure 10, categorization of videos in E001-*Attempting a board trick*, by the assignment vectors in Figure 9 are illustrated. Each column indicates a categorization of videos.

Figure 10. Categorization of videos in E001-*Attempting a board trick*, by the assignment vectors presented in Figure 9.

As we can see, the categorization provides reasonable division of video samples. For example, videos in each category contain similar contents: riding a wakeboard on water in the first category, attempting a skateboard trick on stairs in the second category, snowboarding in plain snow region in the third category, and jumping with a skateboard in the street in the forth category.

It has been found that these vector representations are useful to improve both multimedia event detection and multimedia recounting systems. The utilization of this information will be more discussed in Chapter 7, for an integrated system that takes advantages of MLP with scene concepts.

4.3 Summary

In this chapter, a novel multi-way feature pooling approach is presented to address the problem of complex event detection on unconstrained videos, especially to capture relevant contents effectively from varying contents embedded within temporal structures. For this purpose, the proposed method constructs multiple descriptors with respect to pre-constructed scene concepts. The extensive experiments on the challenging TRECVID 2011 MED dataset demonstrate the usefulness of the proposed method, showing promising performance against strong baselines and the state-of-the-art methods.

CHAPTER 5

MULTIMEDIA EVENT DETECTION BY PER-EXEMPLAR LEARNING

Most semantic categories used for multimedia retrieval have inherent within-class diversity. For example, consider a video search for the concept class "wedding ceremony". Across different cultures, both their looks (visual) and music (audio) are fairly different. The diversity can be dramatic, which raises the question as to whether conventional approaches (e.g., [16, 3]) that learn a single classifier per category can still be successful and scalable as the number of concepts and diversity increase. Furthermore, when blackbox methods such as SVMs are used, recounting the search results or an in-depth analysis of the underlying training data has been challenging. Recounting here refers to the ability to automatically explain to users why some results are retrieved at all, and what particular characteristics triggered them to be returned as results. This process is a core high-level challenge that the multimedia community needs to address.

In this chapter, a retrieval technique based on per-exemplar fusion associations, initially studied in [98], is proposed, as a solution to address the aforementioned challenges. The proposed approach involves training samples as exemplars, and learns localized per-exemplar distance functions centered around each sample. In this way, all of the diversity within the training data is maintained in a straightforward manner. For a new sample, each local distance function only associates itself with samples that are sufficiently similar. Thus, the notion of retrieval is re-defined as an association problem where test data with a relatively high ratio of positive associations are retrieved.

In particular, the per-exemplar learning method is designed to incorporate and fuse multiple types of heterogeneous features, with an emphasis on video retrieval problems. Overall, the resulting learning architecture can be understood as a non-parametric variant of late-fusion approaches where discriminative per-feature base classifiers are used. In detail,

an association between two samples is established by a set of distances across different feature types. Furthermore, for every training exemplar, the relevance of each feature is measured based on its discriminative power around its neighborhood. This is particularly useful because some features may be more relevant for certain exemplars. For example, imagine a birthday video clip recorded in a dark room with a crowd singing a birthday song. It is crucial to learn that the audio (and not visual appearance) is the main relevant feature, and similar samples are discovered mostly based on audio. It is shown that the per-exemplar relevance of each feature as well as the importance of each exemplar, can be automatically analyzed and incorporated to achieve competitive retrieval accuracy.

In addition, it is shown that the proposed method enables a rich set of recounting or summarization capabilities. Due to the nature of association-based retrieval, it is straightforward to identify the exemplars that actually trigger the retrieved data. In addition, the existing knowledge related to the relevant features can be transferred from the exemplars to the target data to describe them. For example, if a large number of exemplars with a metadata tag (e.g., dynamic motion or rock music) are associated with a clip, the metadata can be used to automatically describe the new data. Furthermore, the relevance of each feature dimension can be used to indicate the core evidence considered during the retrieval process. For example, users can easily understand that a particular result has been retrieved due to its audio and/or visual evidence.

The empirical usefulness of the proposed method is evaluated on a challenging real-world dataset, where competitive retrieval accuracy is demonstrated to match or exceed other conventional approaches. Furthermore, the aforementioned novel recounting aspects of the proposed method are highlighted through qualitative analysis.

5.1 Per-exemplar Similarity

In this work, each training sample is regarded as an exemplar, and a local distance function is defined for each exemplar to measure similarities of neighboring samples to the exemplar.

It is assumed that there exist various types of features available, which represent a video sample, including spatial/temporal vision and audio as discussed in Chapter 3. Let $F = \{f_i | 1 \leq i \leq N\}$ be a set of N types of those features, where each feature represents a video sample in a form of a high-dimensional vector.

5.1.1 Local Distance Function

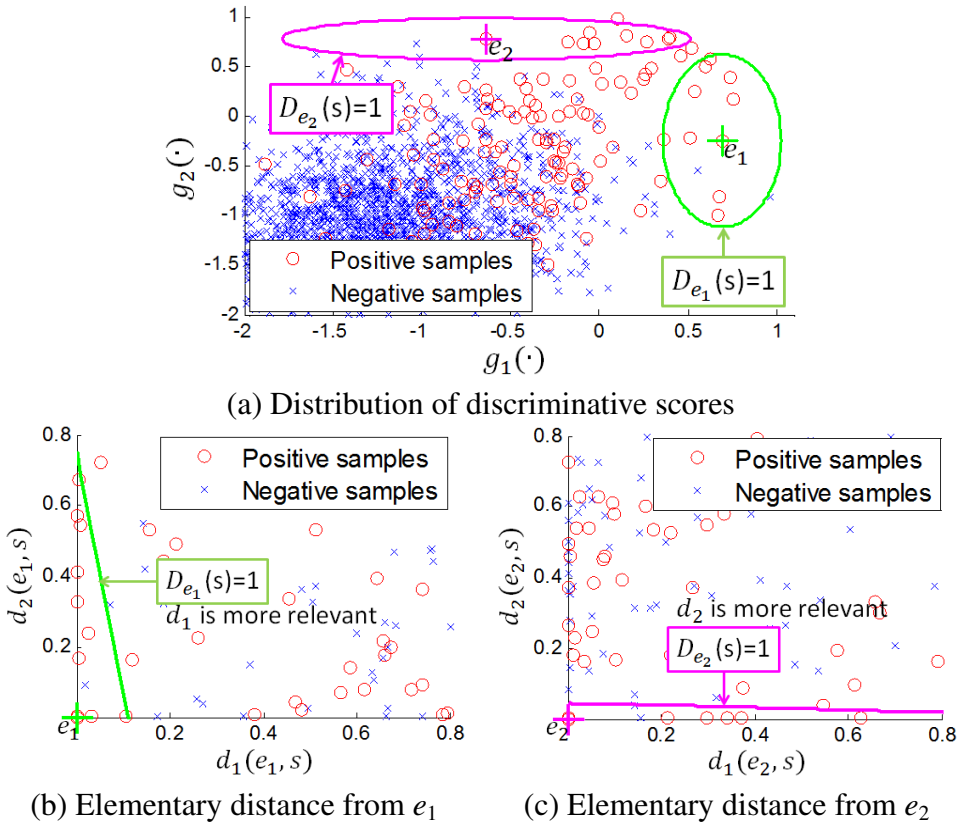
Given a set of multiple features, a simple way to define a local distance measure between samples is concatenating all features, which are high-dimensional, to a single big feature vector (after normalization in each feature), and applying a conventional distance measure, e.g., L_2-distance. However, this method is likely to fail in meaningfully measuring distance among neighboring samples because: (1) the concatenated feature vector may be sparse (consider BoW feature vectors) in the extremely high-dimensional space, and accordingly, distance from a sample point to each other can be equally too far, and (2) despite normalization, the distribution of distance in each feature is not predictable, where a distance measure can be dominated by a few features with large dynamic ranges.

One possible solution to address these concerns is defining a local distance measure as a function of element distances that are derived from each feature types. In particular, the local distance $D_e(s)$ from an exemplar e to a test sample s can be measured by aggregating a set of feature-wise elementary distances $d_i(e, s)$ computed for each f_i. Here, a feature-wise elementary distance $d_i(e, s)$ is pre-constructed in each feature type; detail of an elementary distance used in this work will be discussed in the following sections. The local distance $D_e(s)$ is represented as a linear combination of these elementary distances, as follows:

$$D_e(s) = \sum_{i=1}^{N} \omega_i(e) \times d_i(e, s) = \langle \omega(e) \cdot d(e, s) \rangle, \tag{11}$$

where $\omega_i(e) \geq 0$ denotes the relevance weight for the i-th feature and the corresponding elementary distance $d_i(e, s)$.

Accordingly, each per-exemplar distance function is characterized by a $1 \times N$ parameter vector $\omega(e)$. With a higher value of the weight $\omega_i(e)$, the local distance function is more



(a) Distribution of discriminative scores

(b) Elementary distance from e_1

(c) Elementary distance from e_2

Figure 11. The distribution of discriminative scores (a), and the elementary distance spaces centered around two different exemplars (b) and (c), respectively. Each exemplar has its own local distance function with different relevance weights, where the iso-local distance lines with the different slopes in (b) and (c) appear as the iso-local distance ellipses with the different ratios in (a).

heavily influenced by the similarity in the i-th feature, and vice-versa. The non-negative condition for weights is imposed to ensure that larger elementary distances always lead to larger overall aggregate distances and maintains the notion of distance, which is advocated also in [37, 38].

Concretely, the objective of this method is, for each exemplar e, learning the weight vector $\omega(e)$ in Eq. (11) in a discriminative learning scheme by taking neighboring samples around the exemplar e as training data.

5.1.2 Discriminative Elementary Distance

Among many possible choices for an elementary distance measure, in the proposed per-exemplar fusion approach, a discriminative distance measure is employed. In particular, in

measuring distance between two samples, confidence scores estimated by a discriminative model, e.g., SVMs, which is previously learned by using a corresponding feature type, are used. Therefore, in this scheme, distance among samples is affected by how far they are from a decision boundary. We observed that the proposed approach using the discriminative elementary distance can improve the retrieval accuracy, showing results superior to other approaches using generative elementary distance such as L_2 distance as studied in [37, 38], especially when a feature vector is high-dimensional and extremely sparse, e.g., bag-of-words (BoW) features with thousands codewords.

By using the elementary distance based on discriminative scores, we could acquire more discrimination power than using generative distance such as L_2 distance, We can also take advantage of kernelization in training base classifiers, which often shows significant improvements in many multimedia applications. Moreover, the compact representation by discriminative scores can provide a robust approach to fuse multiple types of heterogeneous features by transforming them into a common vector space.

First, for each feature type f_i, we learn a discriminative base classifier per concept in a one-vs-all manner. Then, we derive an elementary distance $d_i(e, s)$ from the scores output by the discriminant function $g_i(\cdot)$ learned from the i-th feature. ($g_i(\cdot)$ measures the confidence of its input matching a target class based on feature type f_i.)

Specifically, the squared difference between discriminative scores is applied:

$$d_i(e, s) = |g_i(e) - g_i(s)|^2 . \tag{12}$$

Hence, with Eq. (12), a local distance between an exemplar e and a sample s is measured, which is mapped from a feature space to a discriminative score space by the base classifiers. With Eqs. (11) and (12), a local distance function, which is a linear combination of elementary distances, appears as an ellipsoid centered around the corresponding exemplar in a discriminative score space.

Figure 11 geometrically illustrates the local distance functions $D_{e_1}(s)$ and $D_{e_2}(s)$ of the two exemplars e_1 (green) and e_2 (magenta). For clarity, only two types of features

are considered in this example. In Figure 11(a), positive (red 'o') and negative (blue 'x') samples are scattered in a two dimensional discriminative score space by their confidence measures from discriminant functions $g_1(\cdot)$ and $g_2(\cdot)$ learned from different features. We can observe that positive and negative samples are separated with some overlaps. At the top right corner, samples have high scores from the both discriminant functions $g_1(\cdot)$ and $g_2(\cdot)$. On the other hand, we can still see samples with a high score from only one discriminant function, for example, e_1 from $g_1(\cdot)$, and e_2 from $g_2(\cdot)$; in other words, the remaining type of score does not seem reliable. The proposed method attempts to learn a local distance function in a way of suppressing the effect of a relatively unreliable score type, i.e., a loosely related elementary distance.

Figures. 11(b) and (c) are drawn in the elementary distance space defined in Eq. (12) from the two exemplars e_1 and e_2, respectively. The exemplars are located at the origin in Figures. 11(b) and (c), where $d_i(e, e) = 0$. Each axis in the figures is transformed from the corresponding axis in Figure 11(a) by Eq. (12), while the mapped value is non-negative. In this example, we gave the exemplar e_1 a higher relevance weight for the elementary distance $d_1(\cdot)$ than $d_2(\cdot)$, while the exemplar e_2 had a higher relevance weight for the elementary distance $d_2(\cdot)$ than $d_1(\cdot)$. According to the definition of the local distance function in Eq. (11), points that have equal distance from an exemplar can be drawn as a line, e.g., $D_{e_1}(s) = 1$ in Figure 11(b) and $D_{e_2}(s) = 1$ in Figure 11(c), while its inclination is decided by a weight vector on elementary distance measures around the exemplar. It is clear that the iso-local distance line $D_{e_1}(s) = 1$ of the exemplar e_1 in Figure 11(b) is steeper compared to $D_{e_2}(s) = 1$ of e_2 in Figure 11(c). In addition, we can observe that the iso-local distance lines in the elementary distance spaces in Figures. 11(b) and (c) appear as the ellipses with the different ratios and radiuses in Figure 11(a). This illustration provides insights to the influence of a weight vector that differs by an exemplar. A higher relevance weight makes the local distance more heavily influenced by the similarity in the corresponding feature; for example, the iso-local distance ellipse of e_2 contains samples which share more similar

scores from the second feature ($g_2(\cdot)$-axis) than the first feature ($g_1(\cdot)$-axis).

In this manner, we can determine which feature is more relevant to the similarity with a given exemplar than other features. This is one of the key aspects of the proposed per-exemplar learning algorithm, and examples for video recounting will be discussed in Section 5.3.

5.2 Retrieval by Local Distance Function

In the following sections, the approach towards learning relevance weights in Eq. (11) for elementary distance measures, provided by multiple features, is presented. At a high-level, relevance weights of an exemplar e are learned to assign small distance to neighboring samples with the same class with e, while assigning large distance to all the competing samples. In particular, a discriminative learning scheme is applied, where only neighboring positive samples and all negative samples are considered as training data. In this way, we can avoid interruption in learning a local function by other remaining positive samples that might be fairly different from a target exemplar due to huge within-class variability. In addition, it is shown how the learned local distance functions can be combined to estimate the classification probabilities for test samples.

5.2.1 Learning Feature Relevance Weights

In this approach, the set of feature relevance weights for an exemplar e belonging to a particular concept class C is learned by an iterative discriminative neighborhood analysis. During learning, we incorporate the set of the most similar K positive examples, which is denoted as $S_e(C, K)$, and all the available negative examples $\overline{S(C)}$. Accordingly, training samples for each exemplar is different by a set of neighboring positive examples, and can be denoted as $S_e = S_e(C, K) \cup \overline{S(C)}$. The use of only the nearest positive samples is to ensure that localized relevance can be learned, differently per exemplar. Here, the set of positive nearest samples $S_e(C, K)$ are found based on the distance function $D_e(\cdot)$ whose

parameters $\omega(e)$ are iteratively updated to maximize discrimination among training samples. Accordingly, the learning process can be understood as the simultaneous estimation of the weight vector $\omega(e)$ and the localized training subset $S_e(C,K)$. This overall iterative optimization problem is formulated as the following max-margin learning problem with hinge loss functions with respect to the non-negativity constraint imposed by the definition of the local distance function $D_e(\cdot)$:

$$\{\omega'^{*}(e), S_e^{*}\} = \mathrm{argmin}_{\omega'(e),S_e} f(\omega'(e), S_e) \tag{13}$$

$$\begin{aligned} f(\omega'(e), S_e) &= \tfrac{1}{2}\|\omega'(e)\|^2 \\ &\quad + c_1 \textstyle\sum_{j\in S_e(C,K)} \xi_j + c_2 \sum_{j\in\overline{S(C)}} \xi_j \\ s.t.\ \forall i,j &: \left\langle \omega'(e)\cdot d'\left(e, s_j\right)\right\rangle \geq 1-\xi_j, \xi_j \geq 0, \omega_i(e) \geq 0, \end{aligned} \tag{14}$$

where the two extended vectors $\omega'(e) = [\beta_e; \omega(e)]$ and $d'(\cdot) = [1; d(\cdot)]$ are used to consider a bias term, and constant parameters $c_1$and c_2 control the effect of loss terms from the K most similar and competing samples, respectively. Assuming that the ratio of positive and negative samples affects error counts by corresponding class samples, $c_1$and c_2 are set as

$$c_1 : c_2 = \#of\ positive\ samples : \#of\ negative\ samples. \tag{15}$$

Given a training subset S_e, minimizing Eq. (14) is a conventional convex programming problem, which can be solved by a quadratic programming (QP) method such as SVM with the additional non-negative constraints, $\forall i:\ \omega_i(e) \geq 0$. In this work, an approximated implementation to solve14 is used, in a form of a variant of the conventional C-SVM [99]. In particular, the non-negativity constraint is imposed at every iteration of the SVM solver, by setting negative values as 0. This approximation to impose non-negativity constraints while learning SVMs has been studied in [100] and worked well in the proposed method.

Data: $\omega'(e), S_e$

Result: $\{\omega'^*(e), S_e^*\} = \text{argmin}_{\omega'(e), S_e} f(\omega'(e), S_e)$

$k = 0$;

$\omega'^k(e) = \overline{1}$;

while $k < maxIter$ **do**

  update S_e^k;

  $\left\{\omega'^{k+1}(e), S_e^{k+1}\right\} = \text{argmin}_{\omega'^k(e), S_e^k} f\left(\omega'^k(e), S_e^k\right)$;

  if $S_e^k(C, K) = S_e^{k+1}(C, K)$ **then**

    break;

  end

end

return $= \left\{\omega'^k(e), S_e^k\right\}$

Algorithm 1: Learning a local disance function

Then, we can solve Eq. (13) iteratively, as illustrated in Algorithm 1. First, an initial relevance weight vector is set to uniform weights as $\omega'^0(e) = [1, \cdots, 1]$. Then, at every k-th iteration, the current training subset S_e^k can be found based on a current weight vector $\omega'^k(e)$. By using S_e^k and $\omega'^k(e)$ as input of Eq. (13), the next training set and weight vector S_e^{k+1} and $\omega'^{k+1}(e)$ are estimated. This routine is repeated until the set of the K most similar positive samples $S_e(C, K)$ converges, or it reaches maximum iterations allowed. Here, the value of K can affect the computing time of the algorithm. For example, if K is large, the chance that$K$ samples converge becomes low, and accordingly, the number of iterations becomes large. On the other hand, if K is fairly small, $S_e(C, K)$ estimated during iterations is likely to swing among disjoint sets of K samples. Therefore, it is important to set an appropriate value of K. Moreover, a varying value of K according to the distribution of samples around a target exemplar could be helpful. Imagine the cases that neighboring samples are densely and sparsely distributed. If neighboring samples are densely distributed, a small K might be helpful to generate a robust set of neighboring positive samples; on the other

hand, if neighboring samples are sparsely distributed, a large K will be effective to avoid possible overfitting. However, this varying size of neighboring positive samples is beyond the scope of this work and will be remained for a future work. In this work, an appropriate value of K is estimated by cross-validation, which still showed the benefits of the proposed algorithm.

Solving Eq. (13) can be considered as a process of finding a decision boundary between similar neighbors with the same class as an exemplar e and all competing samples with different class labels. After acquiring the optimal relevance weights $\omega(e)$, we divide it by the bias term β_e to normalize so that the value of a learned local distance function at the decision boundary becomes 1. Then, samples within the boundary are defined as 'associated' samples to the exemplar e. Due to the normalization, distances to all associated samples are $[0, 1)$. It is noted that although a decision boundary is learned by using K neighboring positive samples, the number of positive samples inside the boundary is not guaranteed to be K. In other words, according to a margin learned by SVMs, it can vary. In addition, since a decision boundary learned in Eq. (14) is characterized by neighboring training samples of each exemplar, the number of associated test samples may also differ for each exemplar, while a conventional k-NN method keeps the same number of associations across all samples.

5.2.2 Probability Estimation for Retrieval

In this work, we design the probability $\hat{P}(C|s)$ of a test sample s being in a class C to be estimated based on the general form shown in Eq. (16), where the set of all exemplars that built associations with the test sample s is denoted by A_s, while a subset of them which are positive samples of class C is denoted by A_s^C.

$$\hat{P}(C|s) = \frac{\epsilon + \sum_{e \in A_s^C} h(D_e(s))}{\epsilon + \sum_{e \in A_s} h(D_e(s))} \tag{16}$$

Above, the influence function $h(\cdot)$ represents a class of arbitrary functions that decrease with respect to the distance between an exemplar e and the a target sample s. Intuitively,

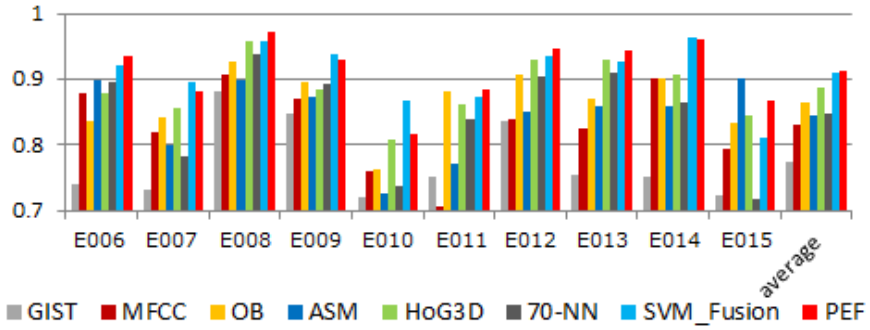
Eq. (16) states that the probability is estimated as the sum of influence by the positive examples, divided by the total amount of influence by all associations. The term ϵ provides smoothing prior which is helpful when the number of associations are sparse or heavily skewed. We found that the proposed distance-based probability model can improve the retrieval accuracy for reasonable choices for h, delivering results superior to the conventional approaches using $\hat{P}(C|s) = |A_s^C|/|A_s|$ which is employed by methods such as k-NN.

5.3 Experiments and Analysis

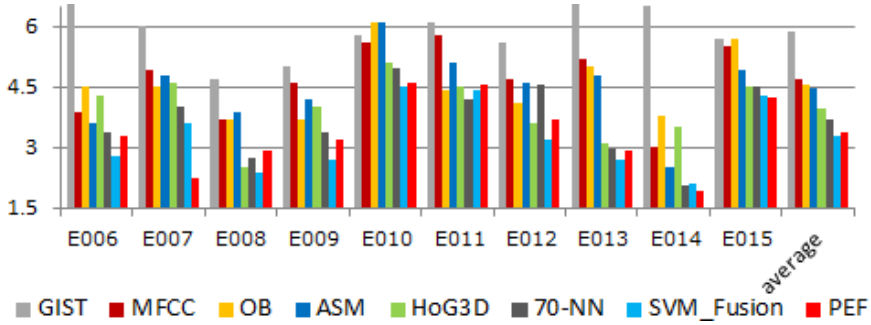
To assess the proposed method, we conducted experiments on a challenging real-world video dataset. For our experiments, we employed the TRECVID 2011 multimedia event detection (MED) data [20]. The MED data provides an excellent test-bed for real-world video retrieval and recounting problems due to its large size and diversity. It consists of consumer videos on the Internet. Accordingly, huge within-class content variability poses significant challenges and fusion can improve retrieval. It consists of 13K training and 32K test video clips labeled with 10 event classes and a pure negative class. The 10 event classes are follows: *Birthday party* (E006), *Changing a vehicle tire* (E007), *Flash mob gathering* (E008), *Getting a vehicle unstuck* (E009), *Grooming an animal* (E010), *Making a sandwich* (E011), *Parade* (E012), *Parkour* (E013), *Repairing an appliance* (E014), and *Working on a sewing project* (E015). For each event class, the training data is substantially imbalanced: there are ~150 positive training samples on average per class, while there are more than 11K pure negative training video clips in total. In the test data, which consists of 32K clips, there are ~120 positive examples (0.4 percent) for each class on average, and approximately 31K videos were pure negative.

5.3.1 Features and Discriminative Distance

To capture diverse information from videos, total of 5 different features were computed. They include 3 visual and 2 audio features: HoG3D bag-of-words (BoW) [80], Object Bank (OB) [84], GIST [36], MFCC BoW, and acoustic segment models (ASMs) BoW



(a) Comparison of AUC (higher is better)



(b) Comparison of $P_{FA}(\%)$ at TER (lower is better)

Figure 12. Performance comparison for video retrieval by different approaches including base classifiers without fusion, k-NN, fusion by a single SVM, and per-exemplar fusion. Two metrics are used, including (a) area under curve (AUC), and (b) P_{FA} at TER.

[87]. For each feature, standard SVMs are individually learned as base classifiers in a one-vs-all manner for each event class. Then, they are used on test data to generate scores which are used as basis to measure per-feature discriminative distance between samples.

5.3.2 Video Retrieval Performance and Comparison

The proposed per-exemplar fusion (PEF) algorithm for video retrieval is evaluated and compared with other methods including per-feature base SVM classifiers, and two alternative fusion methods: (1) k-nearest neighbors (k-NN) and (2) a standard SVM fusion. For k-NN, a score on a test clip was computed by the proportion of the positive neighbors among k neighbors, where we used $k = 70$ (found by cross validation). NNs are found by unweighted Euclidean distances based on discriminative distances, which is a special case of our approach. For SVM fusion, a single SVM fusion classifier was trained using score features formed by concatenating all available discriminative base classifier scores. Other

non-discriminative distances have been studied, but, their results were inferior. Accordingly, the direction has not been further pursued, which is omitted for brevity.

Two performance measures were adopted for our evaluation: (1) area under ROC curves (AUC), and (2) the probability of false alarms (P_{FA}) at a target error ratio (TER) of P_{FA} over the probability of missed detections (P_{MD}) as

$$S_\tau = \tau \times P_{FA} + P_{MD} \text{ s.t. } P_{MD} : P_{FA} = \tau : 1. \tag{17}$$

A specific TER can capture an aspect of user experience regarding the tolerance they are willing to assume between a missed detection and a false alarm. The TER was set to be $P_{FA} : P_{MD} = 1 : 12.5$ in this work. Note that, because positive samples constitute only 0.4 percent of the test data, the per-sample mis-classification cost is still ~16 times higher for a positive sample than a negative one. In terms of implementation detail, the following parameters were used for PEF, based on the analysis of data label imbalance and cross-validation: $\{K, \alpha, \epsilon, c_1, c_2\} = \{5, 1.5, 20, 1\}$. For the influence function in Eq. (16), the following variant of Gaussian function has been used for the result reported in this work:

$$h\left(D_e\left(s\right)\right) = \exp\left[-\alpha\left\{D_e\left(s\right)\right\}^2\right] \tag{18}$$

A summary of the classification results for the 10 test classes are shown in Figure 12. For the two metrics, it can be observed that PEF and SVM fusion consistently outperform all base classifiers, while k-NN shows degradation in AUC. Between PEF and SVM fusion, we observe comparable classification performances: PEF (0.9138) is slightly better than SVM fusion (0.9089) in AUC, but SVM fusion (3.27%) is slightly better than PEF (3.36%) in P_{FA} at TER. Given additional advantages provided by PEF, such as recounting capabilities, the top-end video retrieval performance is appealing.

5.3.3 Qualitative Analysis and Multimedia Recounting

In addition to favorable retrieval performance, PEF provides notable advantages regarding recounting, which is enabled by the association-based retrieval scheme. We can look into

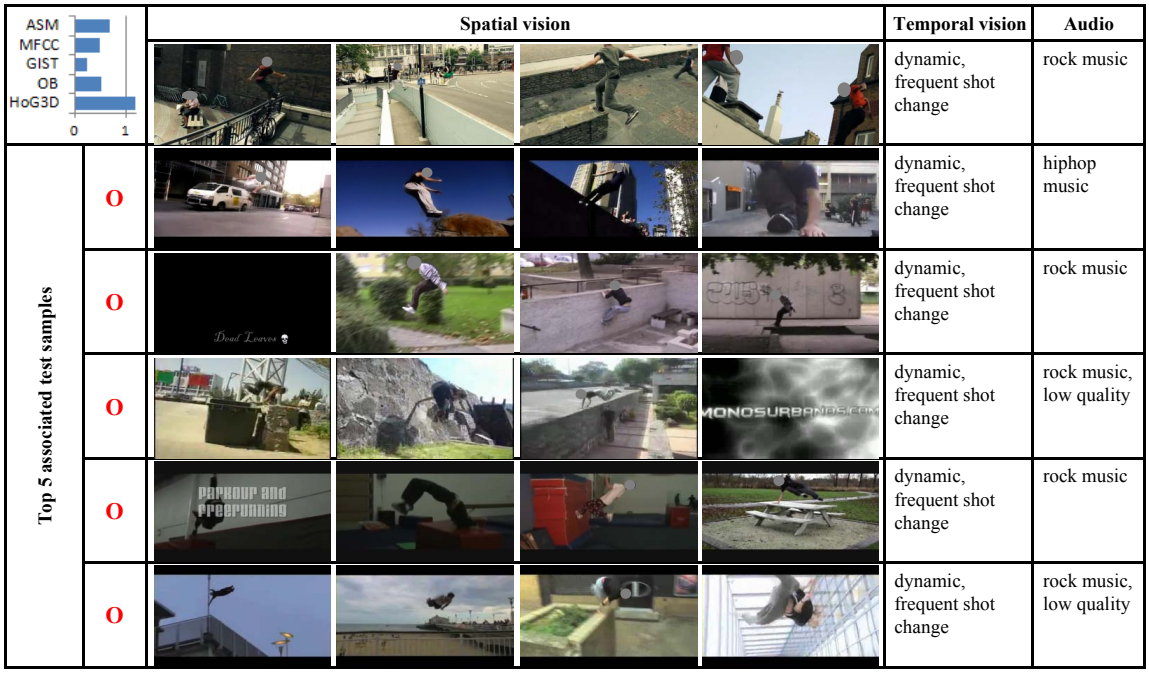


Figure 13. Example of video recounting for *Parkour*: HoG3D is most significant for associations with the given training exemplar, which contains fast movements of objects and frequent shot changes. The consistent music sound type is also notable.

the learned relevance weights of per-exemplar local distance functions and obtain insights about the core characteristics of the exemplar and their associations. In general, samples are associated when highly weighted features are similar to the exemplar. Four examples (*Parkour*, *Grooming an animal*, *Changing a vehicle tire*, and *Flashmob gathering*) are shown in Figure 13-16 where the learned relevance weights are visualized for the training examples at the top, along with the top-ranked associated test examples below. The test examples that share identical labels with the exemplar are marked by circles, otherwise, by crosses. Additionally, frames from each video clip are shown, along with manually marked visual and audio characteristics.

Parkour: Figure 13 illustrates an example video for *Parkour*. The top row indicates that the given training exemplar will associate with samples that have high correlation for HoG3D followed by ASM. The top 5 associated test samples indeed show significant temporal dynamics, i.e., jumping, running fast, tumbling, and etc. It can be observed that the potential automatic recounting of those examples by transferring both temporal vision

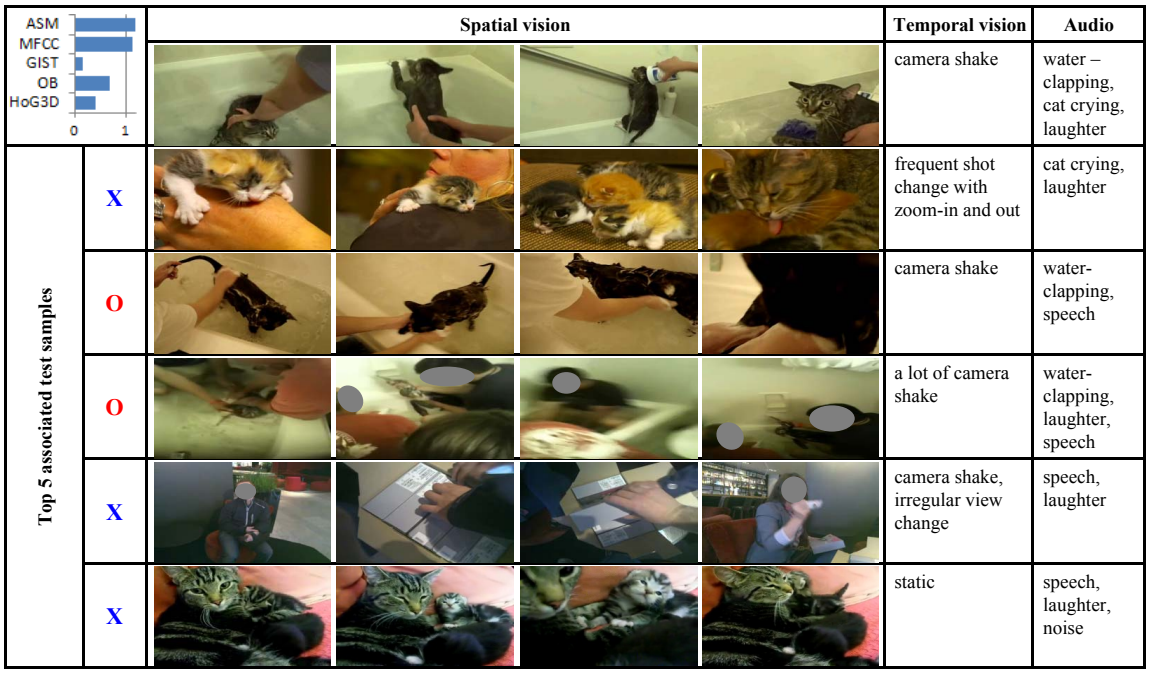


Figure 14. Example of video recounting for *Grooming an animal*: Both audio features are significant, triggered by water-clapping sound for associations with the given training exemplar, which includes water-clapping and laughter sound types.

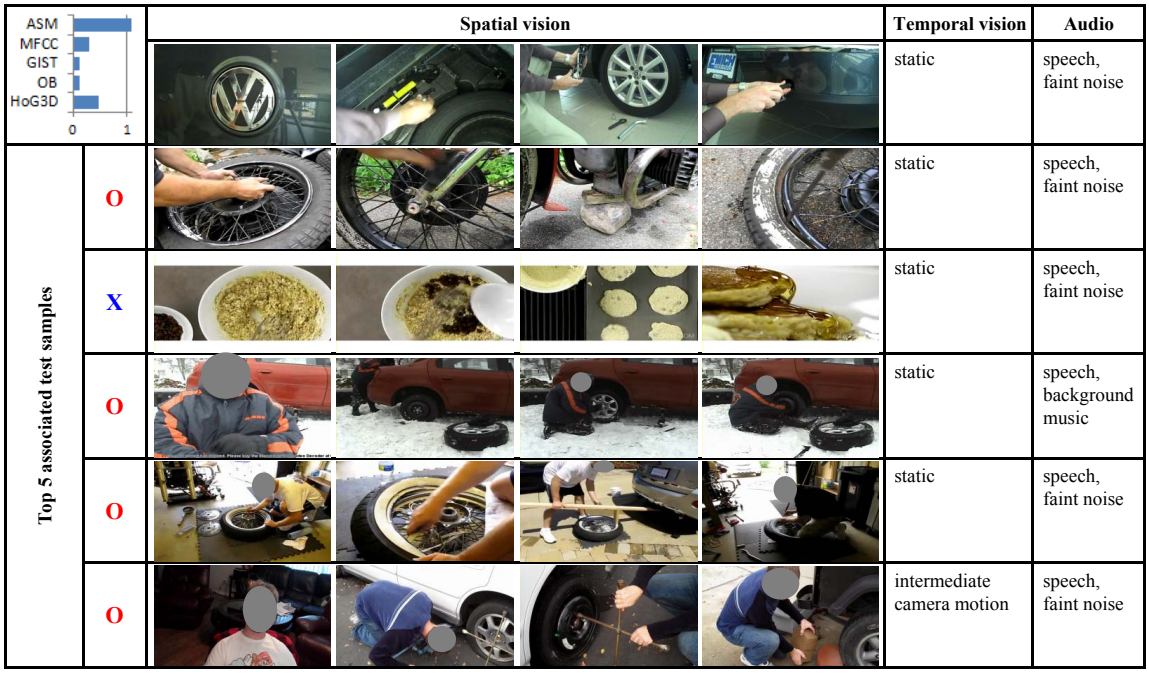


Figure 15. Example of video recounting for *Changing a vehicle tire*: ASM is most relevant to associations with the given training exemplar. For the top 5 associations, speech and faint noise are observed. HoG3D is also significant among visual features, where circular objects and static camera motion can be captured by the dense spatio-temporal gradient descriptors.

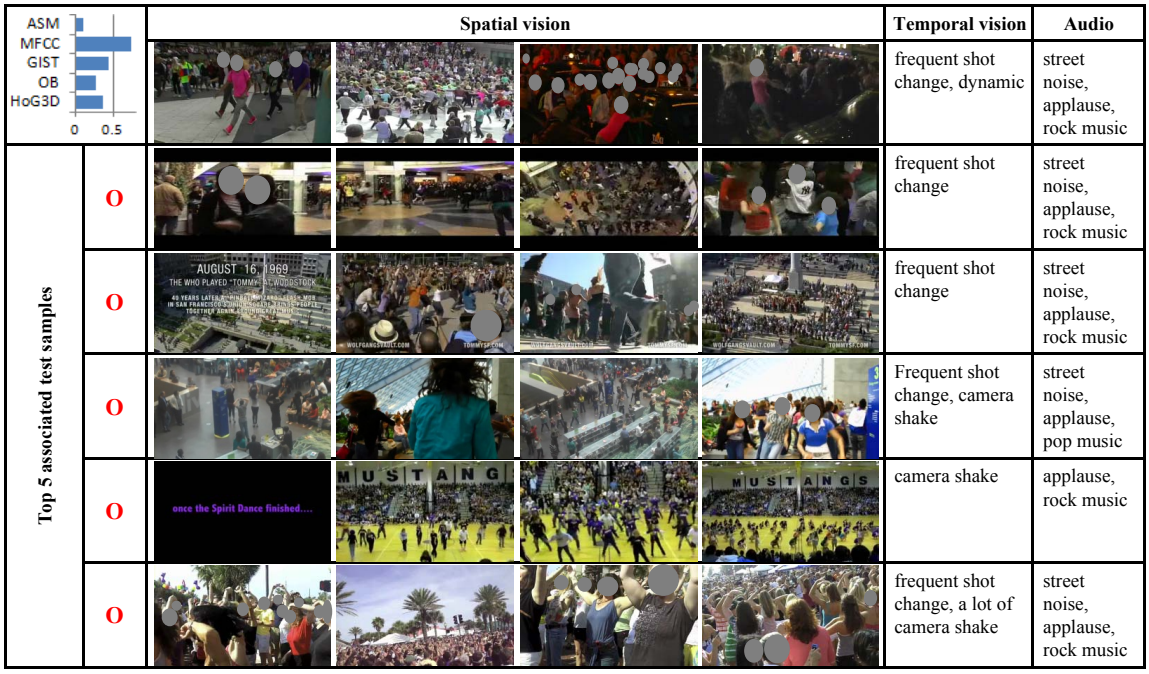


Figure 16. Example of video recounting for *Flashmob gathering*: MFCC is most significant. Street noise, applause and loud music are commonly observed. All the visual features are fairly relevant, which capture a crowd and dynamics in temporal vision.

and audio characteristics of the exemplar will be fairly accurate, regardless of the diversity in the data.

Grooming an animal: Another example for *Grooming an animal* is illustrated in Figure 14. In this exemplar, audio evidence such as water-clapping, cat crying, speech, and laughter are strong, and it can be observed that the corresponding audio features are highly weighted. The third associated sample contains very blurry spatial vision due to camera shakes; however, it could be still accurately retrieved based on audio evidence. While the accuracy of associations is limited in terms of the class label, it is notable that even the incorrect results share similar audio properties. Among visual features, OB is the most important, and it can be seen that associated examples indeed contain similar objects such as cats and hands.

Changing a vehicle tire: Figure 15 illustrates an example video for *Changing a vehicle tire*. In this exemplar, we can clearly observe circular objects at top associated test samples. It is interesting that although the second associated test sample (row 3) is not

involving the target event class, it contains big circular objects in the middle of the scenes, and accordingly, is associated to the training exemplar. It is largely because HoG3D that captures spatial texture is largely weighted. ASM feature is also important because across the exemplar and associated test samples, human speech is strong evidence to relate these video clips.

Flashmob gathering: In Figure 16, the most important evidence to associate the exemplar and retrieved test samples is MFCC that is efficient to capture general audio information across video samples. We could observe that the video samples contain similar audio sound including street noise, applause, and loud music. It is interesting that ASM, which is also capable to capture various audio signatures, is lowly weighted. This might be because between the ASM and MFCC features, MFCC is more efficient to capture such general audio information occurring in the entire videos. The other visual features are also fairly relevant that capture complex textures in the video scenes and dynamics in temporal vision.

5.4 Summary

We presented our novel PEF method for video retrieval and recounting. Our approach incorporates novel schemes to learn and use discriminative local distance functions to associate with examples that share similar properties on core feature channels. Our experimental results on a large consumer video archive is promising: (1) PEF shows favorable retrieval results comparable to competitive alternatives, (2) Furthermore, PEF provides substantial advantages towards understanding core characteristics of each exemplar and automatic tagging of associated samples and retrieval results, which straightforwardly leads to detailed recounting.

CHAPTER 6

EXPLICIT PERFORMANCE METRIC OPTIMIZATION BY MAXIMAL FIGURE-OF-MERIT LEARNING

In many machine learning problems, the success of the learning algorithms is often measured by domain-specific performance metrics that simulate real-world needs or user experience. Such domain-specific performance metrics can be largely categorized into two types. One metric type is based on the classification of competing samples as belonging to a particular class. For example, the precision of top-ranked retrieval results is used in [43, 16]; a weighted sum of the probabilities of missed detection and false alarms is preferred in the TRECVID multimedia event detection (MED) task [20]; and the F_1-score between precision and recall is used in [42]. In these performance metrics, a learning system is evaluated by a true-false classification at a specific operating point. On the other hand, by referring to multiple operating points, the other metric type evaluates the ranking performance of a learning system according to the relevance of retrieved samples to a given query, which has been emphasized in text-document or image/video retrieval systems. For example, in "Google search," orders of top ranked results among numerous test samples can be regarded as more important than the true-false classification. In recent years, average precision (AP) and mean average precision (MAP) have also been widely used in measuring such ranking performance (e.g., TRECVID high-level feature (HLF) extraction tasks [101]).

Most conventional learning approaches attempt to optimize their own learning criteria, as opposed to domain-specific performance measures (e.g., support vector machines (SVMs) learn model parameters that maximize soft margins by incorporating hinge loss). This discrepancy may introduce a mismatch between training and testing conditions, and

may potentially yield sub-optimal solutions. As an effort to address this issue, a maximal figure-of-merit (MFoM) learning framework is proposed in [5]. In an MFoM framework, the four basic elements in the confusion matrix, i.e., true-positve, false-positive, true-negative, and false-negative terms, are approximated and combined to form a continuous and differentiable objective function that simulates a target performance metric. Then, an advanced optimization technique, such as a generalized probabilistic descent (GPD) algorithm [60], is used to explicitly optimize the given performance metric. In this chapter, the idea of MFoM is extended to the optimization of two challenging performance metrics that have become popular in evaluating the qualities of multimedia retrieval systems, namely AP and a weighted sum of the probabilities of false alarms (P_{FA}) and missed detection (P_{MD}) at a target error ratio.

This chapter is organized as follows: first, the numerical formulation of MFoM learning frameworks is presented in Section 6.1. Then, the optimization of AP by using an efficient gradient-based approach is discussed in Section 6.2, with experimental results for an application of automatic image annotation. In Section 6.3, the optimization of a weighted sum of P_{MD} and P_{FA} at a target error ratio is discussed, along with a robust feature fusion method for multimedia event detection.

6.1 MFoM Learning Framework

The proposed learning task is formulated within a discriminative framework. Let $T = \left\{(x, y) \,|\, x \in R^D,\, y \in C\right\}$ be a set of training data, where x is a D-dimensional sample and y is a class label belonging to one of two competing classes $C = \{C_+, C_-\}$, i.e., positive and negative. In this work, a 1-vs-all detection problem is focused for clarity, although a multi-class extension is straightforward [5].

Let $d(x; \Lambda) \in (-\infty, \infty)$ be a *anti-class confidence function* which indicates the confidence that a sample x belongs to the positive class, C_+, where a large positive value corresponds to a low confidence and vice-versa. Given $d(\cdot)$ and Λ, the following decision rule is

applied for the test data:

$$\begin{cases} \text{Label } x \text{ as } C_+, & \text{if } d\,(x;\Lambda) < 0 \\ \text{Label } x \text{ as } C_-, & \text{otherwise.} \end{cases} \tag{19}$$

The goal is to learn the parameters Λ such that the target metric is minimized.

The core ideas of our MFoM-based learning approach are two-fold. First, we exploit the fact that most custom target metrics and their sub-components, such as P_{MD} and P_{FA}, can be expressed as a combination of the four sub-metrics from a confusion matrix: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Second, a target metric that is based on discrete error counts—making it difficult to applying conventional optimization techniques—is approximated with a parameterized continuous and differentiable loss function $L\,(T;\Lambda)$.

In particular, the four sub-metrics are approximated as continuous functions using (truncated) sigmoid functions $\sigma\,(\cdot)$, which approaches one for high confidence for a positive class C_+, or approaches zero otherwise. In detail, the four approximated sub-metrics are expressed as follows:

$$\widehat{TP} = \sum_{(x,y)|y\in C_+} \{1-\sigma(d(x;\Lambda))\}, \tag{20}$$

$$\widehat{FN} = \sum_{(x,y)|y\in C_+} \sigma(d(x;\Lambda)), \tag{21}$$

$$\widehat{FP} = \sum_{(x,y)|y\in C_-} \{1-\sigma(d(x;\Lambda))\}, \tag{22}$$

$$\widehat{TN} = \sum_{(x,y)|y\in C_-} \sigma(d(x;\Lambda)), \tag{23}$$

where the sigmoid function is parameterized by a positive constant α as follows:

$$\sigma(d(x;\Lambda)) = \frac{1}{1+\exp\{-\alpha\cdot d\,(x;\Lambda)\}}. \tag{24}$$

In all, it can be observed that the loss function $L(T;\Lambda)$ is expressed as a function of the distance function $d(x;\Lambda)$.

The anti-class confidence function $d_i(x;\Lambda)$ should be designed to reflect the difference of confidence measures for a corresponding class and competing classes, which is computed from learned class-wise discriminant functions $g_i(x;\Lambda)$, as follows:

$$d_i(x;\Lambda) = -\left\{g_i(x;\Lambda) - \overline{g}_i(x;\Lambda)\right\}. \tag{25}$$

Above, $\overline{g}_i(x;\Lambda)$ is the class-wise anti-discriminant function, which represents an aggregate discriminant score for competing class models. In a general form, it can be defined as a power mean of the competing discriminant functions, as follows:

$$\overline{g}_i(x;\Lambda) = \log\left[\frac{1}{|\overline{C}_i|}\sum_{j\in\overline{C}_i} exp\left\{g_j(x;\Lambda)^{\eta}\right\}\right]^{1/\eta}, \tag{26}$$

where $\overline{C}_i$ represents the set of the competing classes against C_i, $|\cdot|$ is a cardinality of a set, and η is a positive constant controlling the aggregation behavior of Eq. (26). In particular, when $\eta \to \infty$, $\overline{g}_i(x;\Lambda) = \max_{j\in\overline{C}_i} g_j(x;\Lambda)$. In the special case of a binary classification problem, the class anti-discriminant function is simply formulated as a negative of $g_i(x;\Lambda)$ which greatly simplifies Eq. (25).

The choice of $g_i(\cdot)$ can be any differentiable discriminant functions. For example, a linear discriminant function (LDF) can be used as

$$g_i(x;\Lambda) = g_i(x;\Lambda_i) = \sum_{j=1}^{D}\omega_{ij}x_j + \omega_{i0}, \tag{27}$$

where $\Lambda_i = [\omega_{i0},\ldots,\omega_{iD}]$.

Finally, the optimal parameter Λ_{opt} that minimizes $L(T;\Lambda)$ is learned by advanced optimization tools such as the generalized probabilistic descent (GPD) [60], where the GPD conducts iterative descent steps with varying learning rate κ_t as follows (refer to [60] for more detail):

$$\Lambda_{t+1} \leftarrow \Lambda_t - \kappa_t \cdot \frac{\partial L(T;\Lambda_t)}{\partial\Lambda} \tag{28}$$

In all, there are three steps needed for the MFoM framework to be properly used for problems at hand. First, an appropriate parameterized class-confidence function $g(x;\Lambda)$

needs to be defined. The class of linear discriminant functions (LDF) is used in this work; but, in general, any parameterized function can be used [5, 102] such as a kernelized discriminant function [102]. Second, a good mapping function needs to be designed to precisely or approximately simulate the target metric. Finally, an effective constant α which controls the slope of the sigmoid function needs to be selected. The larger α is, the more accurate the approximations in Eq. (24). However, the smaller α is, the smoother the overall approximation in Eq. (51), which facilitates learning when the dataset is small or severely unbalanced. In practice, however, we observed that the choice of α affects convergence speed, rather than accuracy, for most datasets with reasonable sizes.

6.2 Optimization of AP

In an MFoM framework, a learning scheme that approximates a discrete target metric with a parameterized continuous and differentiable loss function has been well studied in [59, 5, 64]. However, unlike performance metrics studied in [59, 5, 64], which are based on discrete error counts, AP is based on the orders of sample scores and cannot be smoothed by approximating the four sub-metrics in the confusion matrix.

In this section, the problem optimizing AP is formulated in an MFoM learning framework. In particular, by assuming AP is a function of individual sample scores, the author proposes an MFoM learning framework that explicitly optimizes AP (MFoM-AP). It is shown that AP behaves like a staircase function with respect to an individual sample score, under the condition where other scores keep their current values. Then, by using a combination of sigmoid functions, the staircase-like AP function is approximated to a continuous and differentiable form. Finally, the gradient of AP is formulated by using a chain rule, and model parameters are estimated by using a GPD algorithm along with adjustment of bias terms. Since the proposed algorithm does not use conventional pair-wise ranking functions, which can be computationally too expensive for practical uses especially in large-scale data, we achieved significantly reduced learning time. To verify the usefulness of the proposed

method, experiments are conducted on two challenging image-retrieval datasets, the Corel 5k and TRECVID 2005 HLF datasets, while the usage of the algorithm in developing a fusion system for multimedia event detection will be discussed in Chapter 7.

6.2.1 Multiple Sub-classes for the Negative Class

In this work, although we assumed a binary classification class, we used multiple sub-classes for the negative class for enhanced discrimination power. In real-world applications, since it is usually necessary to incorporate imbalanced data, in which the negative sample space is much broader then the positive sample space, dividing C^- into several sub-classes could give enhanced discrimination power against the positive class. Therefore, the extended class set $C = \{C^+, C_1^-, \ldots, C_N^-\}$ is used, where $C_1^-, \ldots, C_N^-$ denote the negative sub-classes. For the set of model parameters $\Lambda = \{\Lambda^+, \Lambda_1^-, \ldots, \Lambda_N^-\}$, the class separation function for a sample x is re-defined as

$$d(x;\Lambda) = -g^+(x;\Lambda) + \overline{g}(x;\Lambda), \tag{29}$$

where $g^+(x;\Lambda)$ indicates a class confidence function for a positive class C^+, and $\overline{g}(x;\Lambda)$ represents an anti-class confidence function for the positive class, defined a

$$\overline{g}(x;\Lambda) = log\left[\frac{1}{N}\sum_{i=1}^{N} exp\left(g_i^-(x;\Lambda)\right)^\eta\right]^{1/\eta}, \tag{30}$$

which is a variant of a geometric average among confidence to all competing classes [5]. The positive parameter η controls the aggregation behavior of Eq. (30); in particular,

$$\lim_{\eta\to\infty} \overline{g}(x;\Lambda) = \max_{i\in N}\left\{g_i^-(x;\Lambda)\right\}. \tag{31}$$

The benefits of using a set of multiple negative sub-classes are well studied in [103].

6.2.2 Complexity of Pair-wise Rankings for AP Optimization

One possible method to approximate AP is incorporating pair-wise rankings by a similar manner studied in [68, 69]. Let a ranking score of a sample x be as following:

$$s(x) = -d(x;\Lambda), \tag{32}$$

of which a large negative value corresponds to a high confidence level. We introduced this redundant definition to keep consistent notations with previous research in MFoM learning. Assuming that the numbers of positive and negative samples are M_p and M_n, respectively, we can sort ranking scores of samples given by a classifier. Let $S = \{s_i^+, s_j^- | 1 \leq i \leq M_p, 1 \leq j \leq M_n\}$ be the set of sorted sample scores, where s_i^+ and s_j^- denote the i-th highest ranking score among positive samples and the j-th highest ranking score among negative samples, respectively. In [104], by introducing a pair-wise ranking function, AUC-ROC is formulated as the normalized Wilcoxon-Mann-Whitney (WMW) ranking statistics. In a similar manner, AP can also be represented as following:

$$AP = \frac{1}{M_p} \sum_{i=1}^{M_p} Prec(i) \tag{33}$$

$$= \frac{1}{M_p} \sum_{i=1}^{M_p} \frac{i}{\sum_{j=1}^{M_n} I(s_j^- - s_i^+) + i}, \tag{34}$$

where $Prec(i)$ denotes the precision computed on sorted sample scores when an operating point is the i-th positive score s_i^+, and $I(\cdot)$ is an indicator function that represents pair-wise ranking as

$$I(s_j^- - s_i^+) = \begin{cases} 1, & \text{if } s_j^- - s_i^+ > 0 \\ 0, & \text{otherwise.} \end{cases}. \tag{35}$$

To achieve a smooth overall loss function $L(T; \Lambda)$, previous research in [68, 69] tried to approximate the discrete pair-wise ranking $I(\cdot)$ by using a parameterized sigmoid or polynomial function [68] as

$$S(s_j^- - s_i^+) = \frac{1}{1 + e^{-\beta(s_j^- - s_i^+)}}, \tag{36}$$

$$R_1(s_j^- - s_i^+) = \begin{cases} \left\{ -\left(s_j^- - s_i^+ - \gamma \right) \right\}^p, & \text{if } s_j^- - s_i^+ < \gamma \\ 0, & \text{otherwise.} \end{cases}, \tag{37}$$

where β, γ, and p are positive parameters, and a Gaussian kernel-based or sigmoid-like kernel density function [69] (refer to [69] for more detail).

Although these approaches successfully design a smooth overall loss function $L(T;\Lambda)$ and have shown promising performance, $(M_p \times M_n)$ pair-wise ranking functions still need to be computed; assuming that most real-world problems convey a large number of training samples, the use of pair-wise rankings requires considerable computation and maybe not suitable for practical uses. This complexity issue is a major concern of this work, and we propose an efficient method based on approximation of AP gradients, discussed in the following sections.

6.2.3 AP as a Staircase Function

As discussed in Section 6.2.2, optimization of AP with pair-wise rankings requires considerable computation, which would prohibit practical uses. In this section, AP is formulated as a function of individual sample scores. It is noted that formulating AP as a function of individual sample scores has been already discussed in [74]; however, in this work, the concept is further developed based on the observation that AP behaves like a staircase function with respect to a change of individual sample scores. This observation led us to develop an explicit optimization algorithm that uses the approximation of AP gradients, discussed in Section 6.2.4.

The proposed approach assumes that AP is a function of scores for all positive and negative samples, given by a classifier, as

$$AP = f(s_1^+, s_2^+, \ldots, s_{M_p}^+, s_1^-, s_2^-, \ldots, s_{M_n}^-). \tag{38}$$

Then, by using the partial gradients of AP with respect to a positive and negative score, $\frac{\partial AP}{\partial s_i^+}$ and $\frac{\partial AP}{\partial s_j^-}$, the proposed method formulates the gradient of AP regarding model parameters ω with a chain rule as following:

$$\frac{\partial AP}{\partial \omega} = \sum_{i=1}^{M_p} \frac{\partial AP}{\partial s_i^+} \frac{\partial s_i^+}{\partial \omega} + \sum_{j=1}^{M_n} \frac{\partial AP}{\partial s_j^-} \frac{\partial s_j^-}{\partial \omega}. \tag{39}$$

However, AP is a discrete function with respect to order changes of sample scores. The partial gradients $\frac{\partial AP}{\partial s_i^+}$ and $\frac{\partial AP}{\partial s_j^-}$ are not available and need to be approximated to continuous and differentiable forms to adopt advanced optimization techniques, such as a GPD algorithm.

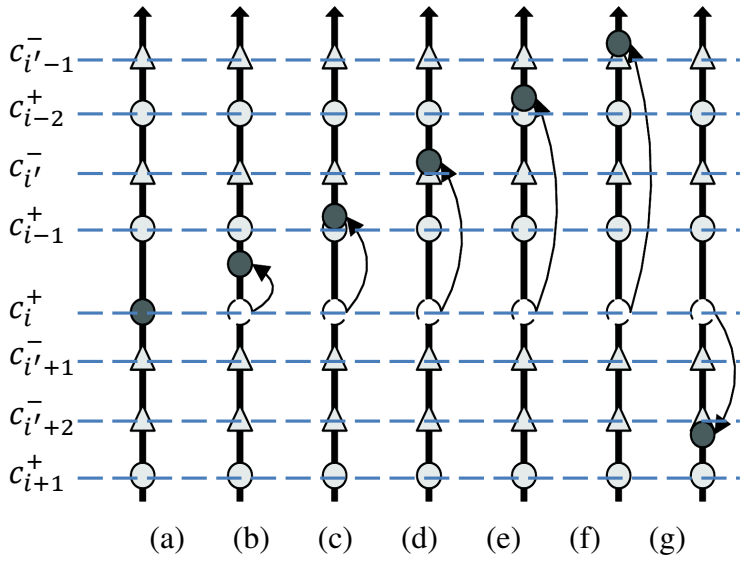


Figure 17. A change of one particular positive score s_i^+. Scores are sorted in descending order: positive scores and negative scores are marked as circles and triangles, respectively. A value of the positive score s_i^+, marked as a dark circle, is assumed to change from the current value c_i^+, while other positive and negative scores keep their current values.

The approximation scheme for the gradient of AP is based on the observation that the AP function behaves like a staircase function with respect to a positive or negative score. To observe this staircase-like behavior of the AP function, it is assumed that a value of the i-th positive score s_i^+ can change, while other positive and negative scores keep their current values. This assumption is illustrated in Figure 17, where all sample scores are sorted in descending order, and positive and negative samples are marked as circles and triangles, respectively. Let the dark circle in Figure 17 be the i-th positive score s_i^+. The current values of the positive and negative sample scores are denoted as c^+ and c^-, e.g., the (j)-th negative value has its current value as c_j^-. Then, the condition that all positive and negative scores, except the i-th positive score s_i^+, keep their current values can be defined as following:

$$D_i^+ = (s_u^+ = c_u^+, s_v^- = c_v^- | 1 \le u \le M_p, 1 \le v \le M_n, u \ne i). \tag{40}$$

Now, we examine how the change of the i-th positive score s_i^+ affects an AP value. If the i-th positive score s_i^+ increases marginally so that an order change between itself and a neighboring score does not occur, as illustrated in Fig 17-(b), AP will keep the current value

as $AP(s_i^+ = c_i^+|D_i^+)$. If the positive score s_i^+ increases a bit further, as illustrated in Figure 17-(c), the order between itself and the (i-1)-th positive score s_{i-1}^+, which is a higher-ranked positive score than s_i^+, will occur. However, in this case, AP will not change again since the order change between s_i^+ and s_{i-1}^+ results only exchanging the values of the precision $Prec(i)$ and $Prec(i-1)$ in Eq. (33), and the sum of these values does not change. Next, imagine the case illustrated in Figure 17-(d), in which the order between the i-th positive score s_i^+ and its nearest-higher-ranked negative score, denoted as $s_{i'}^-$, changes. Unlike the above cases, AP increases to $AP(s_i^+ = c_{i'}^-|D_i^+)$ since the pair-wise ranking function $I(s_{i'}^- - s_i^+)$ in Eq. (33) becomes zero from one by the order change. In other words, an order change between a positive and negative scores makes a discrete step of the AP value. In Figure 17-(e), another order change between the positive scores s_i^+ and s_{i-2}^+ is considered; similar to the case illustrated in Figure 17-(b), the AP value is still equal to $AP(s_i^+ = c_{i'}^-|D_i^+)$. With a further increase of the i-th positive score s_i^+, as illustrated in Figure 17-(f), the order change between the i-th positive score s_i^+ and the next negative score $s_{i'-1}^-$ occurs so that AP increases to $AP(s_i^+ = c_{i'-1}^-|D_i^+)$ since the pair-wise ranking $I(s_{i'-1}^- - s_i^+)$ additionally becomes zero from one, resulting another step in the AP value. On the other hand, with a decrease of the i-th positive score s_i^+, the order change between s_i^+ and $s_{i'+1}^-$ occurs, illustrated in Figure 17-(g). We can observe a step, downward in this case, in the AP value to $AP(s_i^+ = c_{i'+2}^-|D_i^+)$ since the pair-wise ranking $I(s_{i'+1}^- - s_i^+)$ becomes one from zero.

The AP function with respect to the change of one particular positive score can be summarized as follows: (1) since AP is based on ranking of sample scores, if any order change among scores does not occur, AP keeps its value; (2) moreover, order changes among positive sample scores do not affect the AP value; and (3) when order changes between positive and negative scores occur, the AP value jumps up or down. In Figure 18, this behavior of the AP function is illustrated, which is similar to a staircase function. With the current positive score value $s_i^+ = c_i^+$, the AP value is equal to $AP(s_i^+ = c_i^+|D_i^+)$. As the positive score

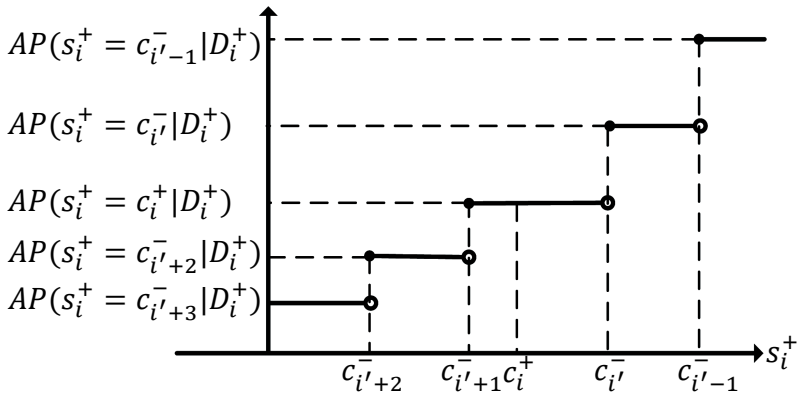


Figure 18. The staircase-like AP function according to the change of one particular positive sample $AP(s_i^+|D_i^+)$ around its current value c_i^+.

s_i^+ changes, the AP function has a discrete step at every point where the positive score becomes equal to the current value of a neighboring negative score, $c_{i'+2}^-$, $c_{i'+1}^-$, $c_{i'}^-$, or $c_{i'-1}^-$. In a similar manner, it can be easily shown that the AP function with respect to the change of one particular negative score s_j^- also behaves like a staircase function, but moves in the opposite direction.

6.2.4 AP Optimization through Approximated Gradients

As discussed in the previous section, the AP function with respect to the i-th positive score $AP(s_i^+|D_i^+)$, illustrated in Figure 18, is a staircase-like function, and accordingly, discrete and not differentiable—making it difficult to apply conventional optimization techniques. The core idea of the proposed method is approximating the discrete staircase function $AP(s_i^+|D_i^+)$ to a continuous and differentiable form and applying a chain rule to acquire approximated gradients of the AP function.

Assuming that two neighboring steps mostly affect the approximated gradient of a staircase function, the proposed method uses a combination of approximated functions around the two neighboring steps, where $s_i^+ = c_{i'}^-$ and $s_i^+ = c_{i'+1}^-$. Among many options, in this work, we used parameterized sigmoid functions in formulating the smoothed staircase

function $\widehat{AP}(s_i^+|D_i^+)$ as follows:

$$\widehat{AP}(s_i^+|D_i^+) = \frac{\sigma_1\left(s_i^+\right) \times \left(s_i^+ - c_{i'+1}^-\right) - \sigma_2\left(s_i^+\right) \times \left(s_i^+ - c_{i'}^-\right)}{c_{i'}^- - c_{i'+1}^-}, \quad \left(c_{i'+1}^- \leq s_i^+ \leq c_{i'}^-\right), \tag{41}$$

where $\sigma_1\left(s_i^+\right)$ and $\sigma_2\left(s_i^+\right)$ denote the two sigmoid functions that approximate the two neighboring steps. Using other functions in approximating a step function has previously been studied, e.g., a polynomial differentiable function [68] and Gaussian or sigmoid-like kernel density functions [69]; investigating those functions are beyond the scope of this work. In a casual manner, the smoothed staircase function $\widehat{AP}(s_i^+|D_i^+)$ is formulated as an internal division of the values of the two neighboring sigmoid functions. In particular, the sigmoid functions are defined as

$$\sigma_1\left(s_i^+\right) = \frac{A_1 - A_2}{1 + exp\left[-\alpha_1\left(s_i^+ - c_{i'}^-\right)\right]} + A_2, \tag{42}$$

$$\sigma_2\left(s_i^+\right) = \frac{A_2 - A_3}{1 + exp\left[-\alpha_2\left(s_i^+ - c_{i'+1}^-\right)\right]} + A_3, \tag{43}$$

where A_1, A_2, and A_3 as following:

$$A_1 = AP(s_i^+ = c_{i'}^-|D_i^+), \tag{44}$$

$$A_2 = AP(s_i^+ = c_{i'+1}^-|D_i^+), \tag{45}$$

$$A_3 = AP(s_i^+ = c_{i'+2}^-|D_i^+), \tag{46}$$

making the approximated AP function continuous within the given interval of $c_{i'+1}^- \leq s_i^+ \leq c_{i'}^-$, between the two neighboring negative scores. At the boundaries, where only one neighboring step exists, i.e., $i' < 1$ or $i' + 1 > M_n$, the proposed approach uses only one part of the sigmoid functions. In a similar way, the smoothed staircase function with respect to a particular negative score $\widehat{AP}(s_j^-|D_j^-)$ can also be formulated, of which the derivation is trivial and omitted for brevity.

Finally, the gradient of AP with respect to classifier parameters can be formulated with a chain rule as

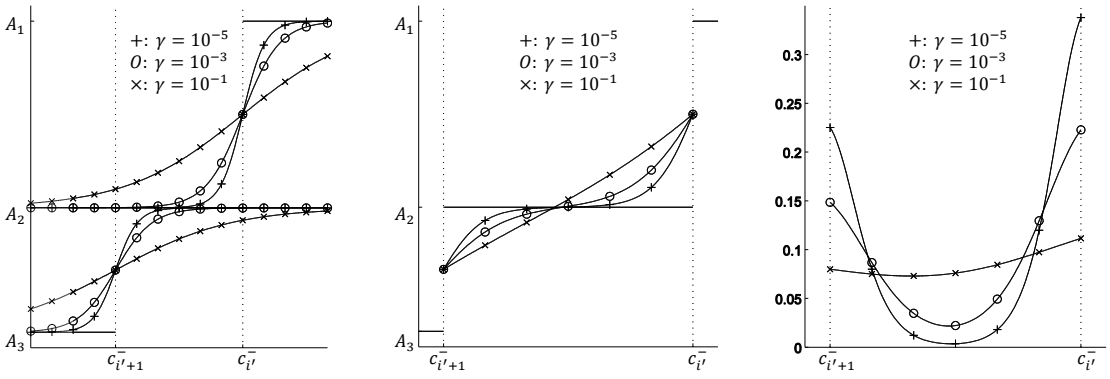


Figure 19. The effect of the parameter γ to the sigmoid functions approximating the AP function: (a) approximation of the two neighboring steps, $\sigma_1\left(s_i^+\right)$ and $\sigma_2\left(s_i^+\right)$, (b) approximated staircase function $\widehat{AP}(s_i^+|D_i^+)$, and (c) approximated gradient $\partial\widehat{AP}(s_i^+|D_i^+)/\partial s_i^+$.

$$\frac{\partial AP}{\partial \omega} \approx \sum_{i=1}^{M_p} \frac{\partial \widehat{AP}\left(s_i^+|D_i^+\right)}{\partial s_i^+} \frac{\partial s_i^+}{\partial \omega} + \sum_{j=1}^{M_n} \frac{\partial \widehat{AP}\left(s_j^-|D_j^-\right)}{\partial s_j^-} \frac{\partial s_j^-}{\partial \omega}, \tag{47}$$

where ω indicates the model parameters. It is noted that the approximated partial gradient with respect to a positive score $\partial\widehat{AP}\left(s_i^+|D_i^+\right)/\partial s_i^+$ always has a positive value, since $\widehat{AP}(s_i^+|D_i^+)$ is an increasing function. On the other hand, the approximated partial gradient with respect to a negative score $\partial\widehat{AP}\left(s_j^-|D_j^-\right)/\partial s_j^-$ always has a negative value, since $\widehat{AP}(s_j^-|D_j^-)$ is a decreasing function. With this approximation, we need to calculate only $(M_p + M_n)$ gradients, and the computational complexity is significantly reduced, when compared to $(M_p \times M_n)$ gradients in pair-wise approaches.

The proposed approximation approach relies on appropriate parameter selection in Eqs. (42-43), namely α_1 and α_2 in the sigmoid functions. Small α_1 and α_2 approximate the AP function to a linear-like function. On the other hand, large α_1 and α_2 make the approximated AP function steep around the two neighboring steps. The smoothness parameters α_1 and α_2 could be set with constant values, which might introduce inappropriate learning-window widths for varying intervals between neighboring scores. On the other hand, optimal values for the parameters may be found through analytic approaches by learning good values in

the training phase, e.g., automatically setting the smoothness through Parzen kernel (window) width estimation [105]. In fact, a more complex scheme of dynamically varying α_1 and α_2 during learning can be beneficial. The investigation of diverse detailed learning strategies is beyond the scope of this work, so we focus on illustrating the proposed algorithm with following parameter selection, which has shown competitive performance in our experiments.

In this work, we defined the parameters of the sigmoid functions α_1 and α_2 as a function of another parameter γ normalized by an interval between neighboring scores as the following:

$$\alpha_1 = \alpha_2 = \frac{\ln(1/\gamma - 1)}{c_{i'}^{-} - c_{i'+1}^{-}}, \ (0 < \gamma < 0.25). \tag{48}$$

The parameter γ controls the smoothness of the sigmoid functions, and its effect is illustrated in Figure 19. It can be observed that, as γ increases, the gradient of the AP function becomes large around the neighboring steps and small away from the steps; thus, an optimization process is mainly affected by sample scores near to the steps.

6.2.5 Parameter Estimation with Bias-term Adjustment

Unlike evaluation metrics based on true-false decision making, e.g., precision, recall, and F-score, the ranking performance, such as AP, does not rely on a learned decision boundary. Instead, AP is defined by orders of scores from positive and negative samples, i.e., it is decided by relative differences among the scores rather than their absolute values. This implies that we can keep the same ranking performance in a training stage, while adjusting bias terms by subtracting or adding a uniform value to the scores.

We observed that, although bias terms do not affect the AP value, they can affect the stability of our learning process. In Figure 20-(a), the effect of a bias term is conceptually illustrated with a simple binary model; two hyperplanes, *Model-A* and *Model-B*, are considered, which have same model parameters β, but differ only by a bias term β_0. For a given set of samples, denoted as *Data* in Figure 20-(a), the two hyperplanes generate the same

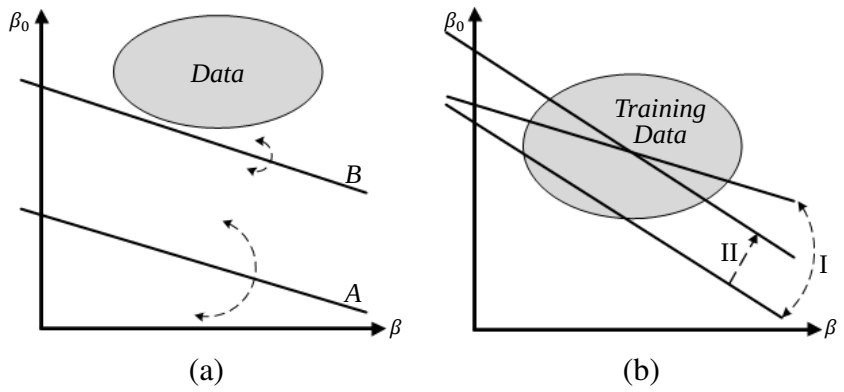


Figure 20. (a) Models with only difference by β_0 and (b) Two-step parameter estimation:(I) Update using approximated $\partial AP/\partial\beta$ and (II) Adjustment of β_0.

ranking statistics. However, assuming that a confidence score is computed by the distance between an instance and a hyperplane, *Model-B* requires higher values of parameter updates in a training stage to result the same amount of score change compared to *Model-A*. On the other hand, small parameter updates in *Model-A* could result the large fluctuation of scores, making it difficult for test performance to converge, especially when the distribution of training and test instances is different. Moreover, *Model-B* is more robust in terms of ranking statistics than *Model-A*, since normalized scores from *Model-B* are more discriminative than those from *Model-A*. With this observation, we adopted a two-step method for parameter estimation, which includes adjustment of constant terms, as follows:

- Step I. The entire parameter set is updated by a GPD algorithm with the approximated gradient of AP in Eq. (47). Specifically, we used the inexact line search algorithm [106] to find an appropriate learning rate.

- Step II. $\Delta\omega$ is computed to maximize F_1 by the following decision rules: positive decision is made if $d(x;\Lambda) + \Delta\omega < 0$, and negative decision is made, otherwise. Then, adjust a constant term for the positive class as

$$\omega_0^+ \leftarrow \omega_0^+ - \Delta\omega. \tag{49}$$

This two-step method is conceptually illustrated in Figure 20-(b). With this manner, we also

expect that the proposed algorithm can additionally provide reasonable decision results if they are necessary as well as ranking performances.

6.2.6 Experiments and Analysis

The proposed MFoM-AP learning algorithm for AP optimization is analyzed on an automatic image annotation (AIA) task using two challenging datasets. One is the Corel 5k dataset, which contains 5,000 images–4,500 training and 500 test samples. The dataset is labeled with 374 semantic concepts; in this work, 36 classes with at least 100 positive training samples are evaluated. The other one is the TRECVID 2005 dataset, which is a large-size dataset including 61,901 keyframes from 137 video clips of broadcast news. The keyframes were labeled with 39 classes defined by the LSCOM-Lite annotation set [10]. Across the experiments, the same class-identification numbers arranged in [107] are used. 70% of the keyframes were randomly selected as a training set and the others are used as a test set. We set $\gamma = 10^{-5}$ in Eq. (48).

For the both experiments, we used color and texture features based on dense grids, similar to those used in [49]. In particular, we divided an image into 8x8 dense grids. Then, from each grid, we extracted a 12-dimensional color histogram, by using mean and variance of RGB and LAB, and a 12-dimensional texture feature, by using energy of log Gabor filter. The extracted features are separately quantized to color and texture codewords, an image feature vector is constructed with a bag-of-word (BoW) representation. Since we used 64 codewords for each color or texture feature and considered unigrams and bigrams, the dimension of the feature vector for each feature is ${}_{65}C_2$=2,080, considering an edge as another codeword. Then, we applied latent semantic indexing (LSI) [108], and the two features are fused in an early fusion manner by concatenating them. The dimension of the final feature vectors was decided by cross-validation, resulting 1,200- and 1,270-dimensional feature vectors for the Corel 5k and TRECVID 2005 datasets, respectively.

Table 7. Comparison of MFoM-AP, pair-wise-AP and MFoM-F_1 in mean AP of 36 semantic classes for the Corel 5k dataset.

	MFoM-AP	pair-wise-AP	MFoM-F_1
Training	0.7921	0.7292	0.6478
Test	0.4283	0.4214	0.3925

6.2.6.1 Experiments on Corel 5k dataset

To verify the usefulness of the proposed MFoM-AP in improving AP, we compared its performance with those obtained with a pair-wise AP optimization scheme (pair-wise-AP) and an MFoM learning method optimizing F_1 (MFoM-F_1), by using the same confidence scores defined in Eq. (32). In particular, for pair-wise-AP, we applied a sigmoid function to approximate pair-wise rankings in Eq. (34), as studied in [68]. The experimental results of mean average precision (MAP) over the 36 semantic classes in the Corel 5k dataset are summarized in Table 7. Overall, MFoM-AP showed the best MAP performance (0.4283), followed by pair-wise-AP (0.4214), and then MFoM-F_1 (0.3925).

In Figure 21, examples of iterative performance by the proposed method are illustrated. It is noted that training and test performance are drawn in separate scales for clarification. We have observed that training performance is gradually improved and converged as iterations proceed. Compared to training performance, test performance shows shaking behaviors, especially at early iterations. This is because the number of test data are much smaller than that of training data (one tenth in our experiments), and their ranking changes can result large variation in AP values. Nevertheless, we can observe that the test performance is stabilized as iterations proceed, and the final results show reasonable performance through the iterations.

Since AP is largely affected by top ranked samples, we have also observed top retrieved results in training and test data. Top 10 retrieved results for the *grass* and *bear* classes are illustrated in Figure 22 and 23, respectively. It is interesting that although missed detections are found in the test data, they are very similar to one of the top retrieved results in training data. For example, in Figure 22, the second and third test samples correspond to the fifth

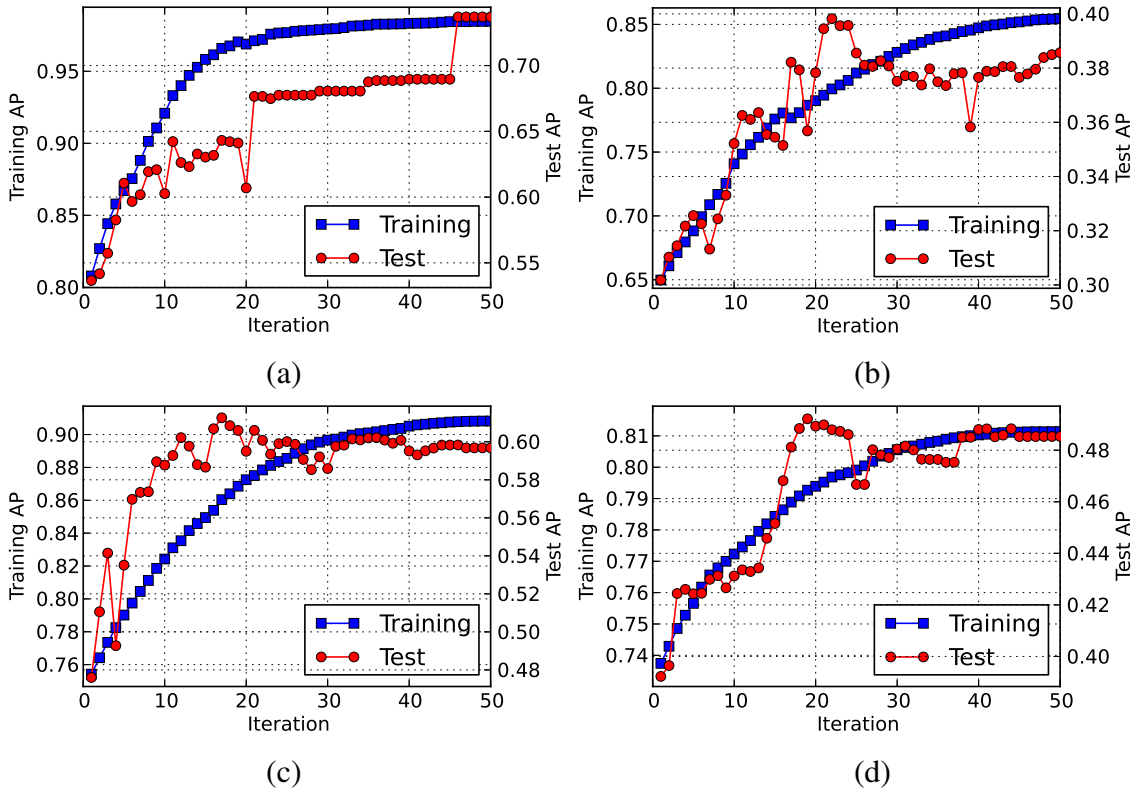


Figure 21. Examples of iterative performance by MFoMAP on semantic classes in the Corel 5k dataset: (a) sun, (b) grass, (c) bear, and (d) snow. Training and test performance are drawn in separate scales.

and second training samples, respectively, which can also be seen near misses. These near misses showcase the benefit of the proposed method that retrieves test samples considerably similar to top-ranked training samples.

For MFoM-AP, we need to perform an additional sorting routine, which can be done in $O\left\{(M_p + M_n)\log(M_p + M_n)\right\}$, and approximation process is more complex: two sigmoid functions for MFoM-AP vs. one sigmoid function for pair-wise-AP. However, the number of approximation computations significantly decreases by using MFoM $(M_p + M_n)$ instead of pair-wise-AP $(M_p \times M_n)$. For example, given 200 positive and 4.3k negative samples, we need 4.5k computations of gradient approximation with MFoM-AP, but 860k computations of pair-wise ranking approximation with pair-wise-AP. This reduced number of approximation computations largely affects the learning time compared to the above mentioned additional cost since deriving approximated gradients involves large computations,

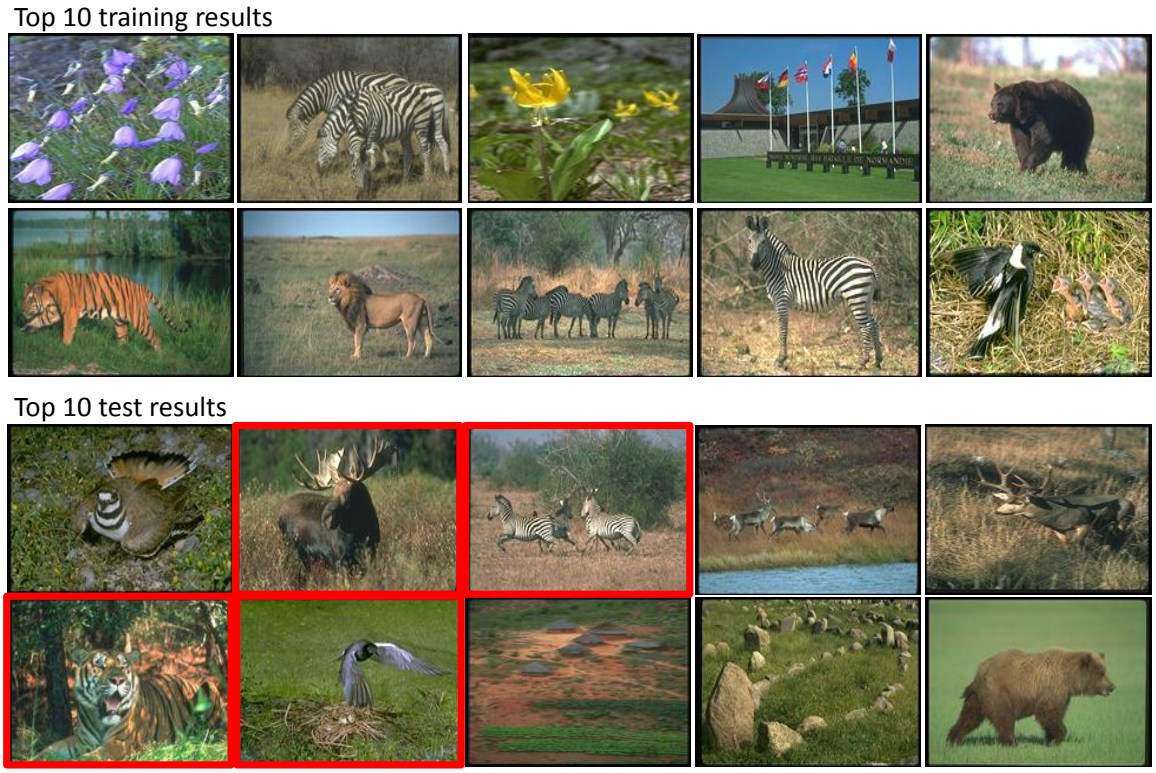


Figure 22. Top 10 results for the *grass* class in the Corel 5k dataset. The results are sorted from top-left to bottom-right, while missed detections are marked with red boxes.

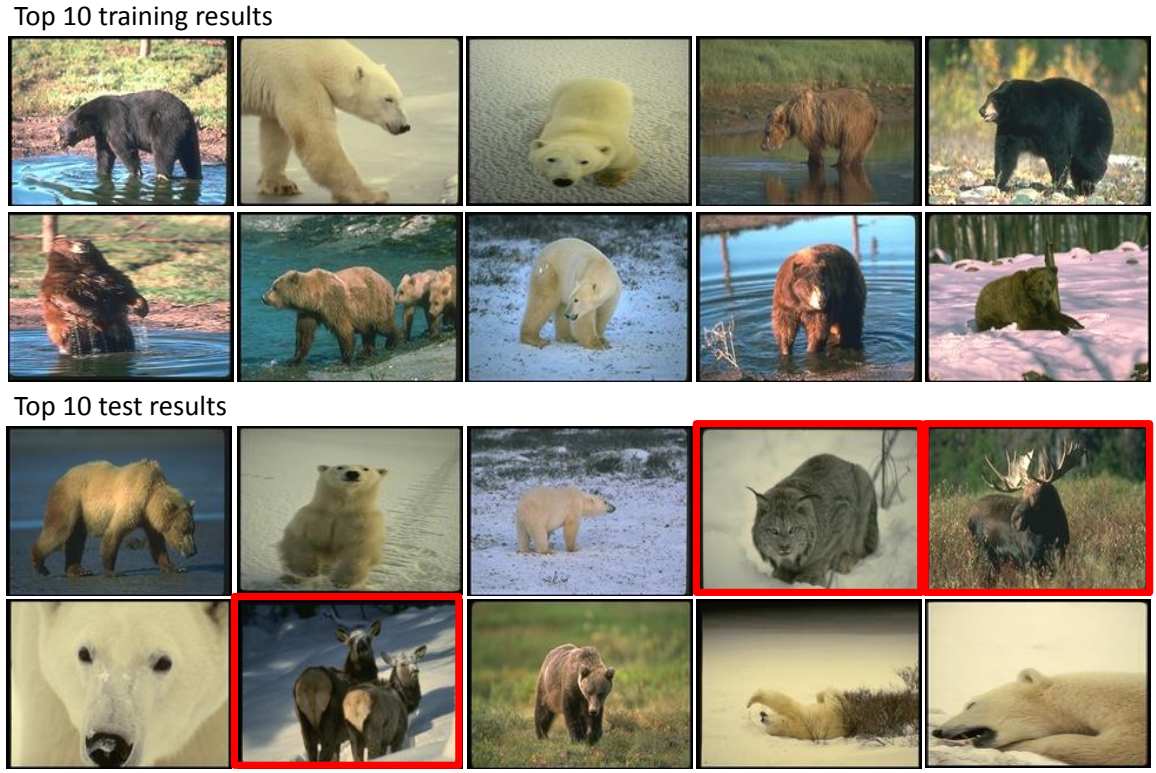


Figure 23. Top 10 results for the *bear* class in the Corel 5k dataset. The results are sorted from top-left to bottom-right, while missed detections are marked with red boxes.

Table 8. Learning time ratio of MFoM-AP to pair-wise-AP on the Corel 5k dataset.

Semantic class	# of Pos.	# of Neg.	Learning time ratio (pair-wise-AP / MFoM-AP)
sun	101	4399	43.8
grass	446	4054	132.3
bear	198	4302	80.4
snow	267	4233	83.6
Average of the 36 classes	255.4	4244.6	103.2

e.g., the dimension of feature vectors is usually high such as 1,200 for our experiments on the Corel 5k dataset.

Therefore, although MFoM-AP and pair-wise-AP showed comparable results on the Corel 5k dataset, we achieved remarkably reduced learning time–approximately 103.2 times faster in average–by using MFoM-AP. In Table 8, the learning time ratio of MFoM-AP to pair-wise-AP is shown for some selected semantic classes on the Corel 5k dataset. It showed a tendency that the efficiency of the proposed learning method is more improved as the number of positive training data increases; in other words, positive and negative data are more balanced.

6.2.6.2 Experiments on TRECVID 2005 dataset

For the TRECVID 2005 dataset, only MFoM-AP and MFoM-F_1 are compared because of the large size of the dataset and the above mentioned complexity issue. Table. 9 shows the training and test results of MFoM-AP and MFoM-F_1 in detail. We can observe that MFoM-AP consistently outperforms MFoM-F_1 in AP. The MAP of MFoM-AP among the 39 concepts were improved by 18.5% from 0.5356 of MFoM-F_1 to 0.6346 for the training data, and by 5.8% from 0.4039 to 0.4274 for the test data. The reason why the improvement rates differs much between training and test stages is that AP can be easily improved in a training stage especially for the concepts with the small number of positive samples, since AP is largely influenced by top-ranked scores. However, these big improvements could not be directly applied to the test performance, since the number of positive samples for those

Table 9. Comparison of MFoM-AP and MFoM-F_1 in AP for the TRECVID 2005 dataset.

		MFoM_AP		MFoM_F_1				MFoM_AP		MFoM_F_1	
con#[1]	pos#[2]	trn	tst	trn	tst	con#	pos#	trn	tst	trn	tst
1	2245	.717	.631	.635	.572	21	4056	.526	.426	.494	.404
2	10466	.847	.802	.820	.792	22	1443	.437	.304	.379	.273
3	581	.804	.613	.626	.547	23	433	.508	.211	.342	.166
4	172	.825	.326	.626	.299	24	1967	.491	.359	.439	.345
5	1104	.486	.196	.395	.185	25	93	.782	.211	.600	.188
6	2434	.554	.493	.514	.473	26	393	.722	.460	.529	.448
7	5174	.920	.883	.886	.871	27	1903	.710	.622	.584	.580
8	16768	.789	.781	.768	.753	28	393	.657	.294	.471	.276
9	4840	.473	.404	.455	.391	29	422	.527	.115	.383	.094
10	476	.616	.251	.471	.232	30	3339	.534	.392	.504	.378
11	4238	.473	.402	.462	.382	31	216	.526	.188	.436	.162
12	658	.689	.435	.560	.414	32	504	.476	.087	.346	.078
13	3066	.470	.379	.440	.369	33	300	.784	.250	.480	.193
14	5626	.651	.598	.619	.563	34	4612	.356	.286	.366	.294
15	241	.761	.266	.549	.235	35	1060	.462	.260	.381	.236
16	4772	.470	.407	.442	.394	36	857	.531	.298	.422	.267
17	1304	.639	.482	.543	.479	37	374	.505	.154	.361	.137
18	6539	.620	.643	.605	.591	38	760	.818	.643	.697	.609
19	27596	.924	.912	.911	.907	39	406	.725	.267	.413	.283
20	32922	.945	.938	.937	.932	MAP		.635	.536	.427	.404

[1]Class-identification number in TRECVID 2005 dataset

[2]The number of positive samples in the training data

concepts is relatively too small to cover the large variation of the test data set. We found that 12 concepts, #1, #3, #15, #22, #23, #25, #29, #31, #32, #33, #35, and #36, showed significant improvements as more than 10% in a test stage, while some concepts like #17, #19, #20, and #34 showed improvements less than 1%.

AP is considered equivalent to the area under the precision-recall curve (AUC-PR). The PR curves of some example classes in the TRECVID 2005 dataset are illustrated in Figure 24. It is notable that the curve of MFoM-F_1 abruptly drops in Figures. 24-(b) and (d). The breakpoints are the operating points, learned to maximize the F_1 metric by MFoM-F_1. Although MFoM-F_1 can outperform MFoM-AP in the F_1 measure at these operating points, it is clear that MFoM-AP outperforms MFoM-F_1 in AUC-PR. This result can give

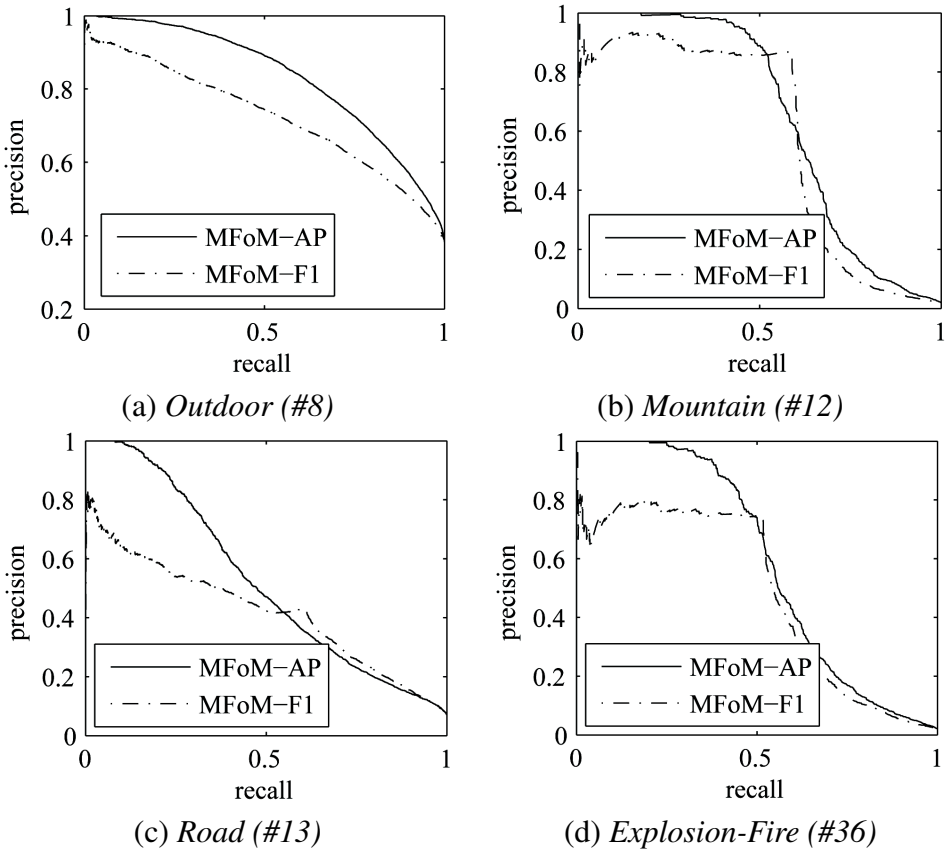


Figure 24. Precision-recall curve. Obviously, MFoM-AP outperforms MFoM-F_1 in AUC-PR.

a good explanation as to why learning schemes minimizing classification errors cannot guarantee optimizing ranking-performance metrics.

6.3 Optimization of P_{MD} and P_{FA} at a Target Error Ratio

In real-world retrieval tasks, the performance metrics that capture user desires can differ widely from application to application. For example, for a 'Google search', the important metric may be precision of the top-N, for small N (N = 10 to 50, say). For a statistical analysis problem, on the other hand, recall may be the most important factor. In general, a large class of these metrics can be thought of as the weighted combinations of P_{MD} and P_{FA} at a particular operating point.

In this section, the weighted sum of P_{MD} and P_{FA} at a particular ratio is mainly considered. Concretely, the goal is:

$$\text{Minimize } S_\tau = P_{MD} + \tau \times P_{FA} \quad \text{s.t.} \quad \frac{P_{MD}}{P_{FA}} = \tau. \tag{50}$$

In the following, the approach with regards to this particular metric is explained. For example, $\tau = 12.5$ is used for our experiments on TRECVID 2011 MED dataset [20]. However, we note again that the learning method is more general, and can be easily applied to other metrics such as rankings, F_1 or average precision [5, 63].

To optimize the metric in Eq. (50), a standard scheme is to learn a model with its own learning objective functions (e.g. error rates) and adjust its detection threshold until the desired ratio of $P_{MD}/P_{FA} = \tau$ is met where the metric S_τ will be computed. With this approach, however, there is no guarantee that the learning procedure will focus on improving performance at particular operating points. The proposed solution described in the following sections provides a principled approach to achieve such a goal.

In the MFoM learning framework, discussed in Section 6.1, the overall loss function $L(T;\Lambda)$ is formulated from approximate sub-metrics $\widehat{(.)}$ in Eqs. (20-23), using a mapping function $f(\cdot)$ as follows:

$$S_\tau \approx L(T;\Lambda) = f\left(\widehat{TP}, \widehat{FP}, \widehat{TN}, \widehat{FN}|\Lambda\right) \tag{51}$$

The role of the mapping function f is to reconstruct the loss function L accurately from sub-metrics. In fact, if the given target metric is a simple combination of sub-metrics, a precise mapping f is possible; e.g., for the F_1 metric where $F_1 = 2TP/(2TP + FN + FP)$. In some cases, however, the loss function may involve complex conditions such as the ratio constraint in Eq. (50), which needs approximation. This issue is further discussed in the following section.

6.3.1 Strategies for Complex Target Metric Approximation

In this section, we present how a good mapping function f in Eq. (51) can be designed to yield an accurate continuous loss function $L(T;\Lambda)$ for a given target metric, with focus on

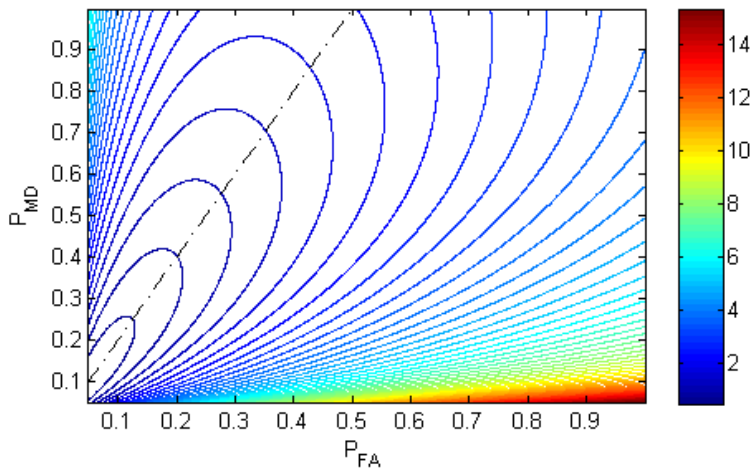
the example metric introduced in Eq. (50).

For cases where complex target metrics prohibit the use of precise mapping function f, the proposed method is to approximate the target metric as a combination of simpler sub-functions. This usually involves a set of parameters Γ which control the relative weights of sub-functions. Optimal values for Γ may be found through analytic approaches by minimizing the divergence between the resulting approximation f and the given target metric. On the other hand, good values for Γ can be found through cross validation as well. In fact, a more complex scheme of dynamically varying Γ during learning can be beneficial. For example, in Eq. (50), an optimal value for Γ may differ according to varying values of P_{MD} and P_{FA} during learning steps. The investigation of diverse detailed learning strategies is beyond the scope of this work, so we focus on illustrating these ideas on a concrete example below.

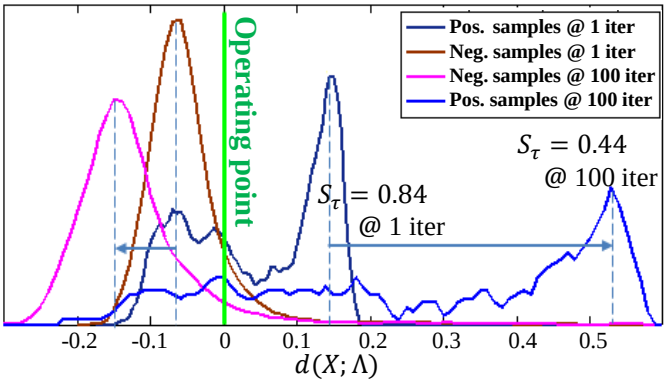
For the example target metric in Eq. (51), a linear sub-function for weighted error rate $\left[\widehat{P_{MD}} + \tau \times \widehat{P_{FA}}\right]$ can be incorporated in a straightforward manner where the approximations $\widehat{P_{MD}}$ and $\widehat{P_{FA}}$ are set to be equal to $\widehat{FN}$ and $\widehat{FP}$ (in Eqs. (21) and (22)) divided by the total number of positive and negative samples respectively. In addition, our mapping function should be designed to prefer user-specified target ratio τ between P_{MD} and P_{FA}. To enforce such a ratio constraint, we include a sub-function $R(\tau,\, P_{MD}/P_{FA})$ which monotonically increases loss with respect to the difference between a target ratio τ and the exhibited ratio $\widehat{P_{MD}}/\widehat{P_{FA}}$. By incorporating both terms with a weighting parameter Γ, the loss function $L(T;\Lambda)$ that approximates Eq. (50) is finally defined as:

$$L(T;\Lambda) = \left[\widehat{P_{MD}} + \tau \times \widehat{P_{FA}}\right] + \Gamma \times \left[R\left(\tau,\, \widehat{P_{MD}}/\widehat{P_{FA}}\right)\right] \tag{52}$$

With small Γ, learning focuses more on minimizing the error rate; however, the learned model is less likely to show a desired target error ratio τ, since the minimum value of the weighted error rate could be derived by reducing P_{FA} and sacrificing P_{MD}, especially when τ is large. On the other hand, with large Γ, learning will focus more on meeting target error ratio, and less on decreasing error rates. In this work, Γ is set to a fixed constant by



(a)



(b)

Figure 25. (a) Iso-contour curves of the loss function $L(T;\Lambda)$ defined in Eq. (52) when $\tau = 2$ and $\gamma = 1$. The dashed straight line corresponds to a iso-ratio $P_{MD}/P_{FA} = 2$.
(b) Distribution of the confidence function $d(x;\Lambda)$ after 1 and 100 iterations in MFoM learning for positive and negative samples when $\tau = 5$. As expected, false positives are suppressed more than false negatives, resulting in an error ratio of 4.68**.**

searching through cross-validation; this has shown promising results.

Among many options for the ratio constraint approximation term R, we found the following form to work well and used it in this work:

$$R\left(\tau,\ \widehat{P_{MD}}/\widehat{P_{FA}}\right) = \left\{\log(\tau) - \log\left(\frac{\widehat{P_{MD}}}{\widehat{P_{FA}}}\right)\right\}^2 \tag{53}$$

The logarithmic squared form used above provides a computational advantage in that overall gradients can be easily computed as a sum of two terms (i.e., the gradients of P_{MD} and P_{FA}), avoiding the complications potentially caused by the direct use of division P_{MD}/P_{FA}. Furthermore, it is found that the use of logarithmic functions provides a balancing effect

which alleviates unintended severe dominance of the target ratio error term during optimization.

To showcase the quality of the approximation in Eq. (52), Figure 25(a) illustrates the iso-contour curves of the loss function, along with the dashed line which corresponds to the ratio constraint for the case of $\tau = 2$. It can be clearly seen that the designed loss function is correlated with and declines towards the iso-ratio line. This implies that the minimum value of the loss function defined in Eq. (52) can be found near the iso-ratio line and left-bottom of the plot through the gradient descent procedures given in Eq. (28).

More in detail, the behavior of the proposed approach during learning is our MFoM learning framework in depicted in Figure 25(b). It plots the values of class-confidence function $d(x;\Lambda)$ for positive and negative samples for the 1st and 100th iterations when $\tau = 5$ (i.e., when a desired operating point is $P_{MD}/P_{FA} = 5$). Observe that as learning proceeds, the overall error rate (i.e., the weighted sum of P_{MD} and P_{FA}) has been decreased as larger separation between positive and negative classes is achieved. More interestingly, for a fixed threshold of zero for the operating point, the ratio of P_{MD} to P_{FA} approaches the desired value of 5, guided by the constraint term in Eq. (53).

6.3.2 Fusion Framework

The fusion-based video retrieval architecture used in this work is formulated within the *late fusion* paradigm, e.g., [47]. By late fusion, we mean that scores are computed independently by multiple base classifiers, one per feature type, and fusion classification is conducted on the base classifier scores. The MFoM approach to learn the fusion classifier parameters is used, which simultaneously optimizes target performance metrics explicitly. In particular, a specific training approach for late fusion systems is used, described below, which turns out to be crucial to maintaining performance on novel test data.

The overall architecture for discriminative score fusion is illustrated in Figure 26(a), where three separate data flows are shown, for proxy base classifier training (blue dashed), fusion classifier training (green dashed), and test phase (solid red) respectively. During the

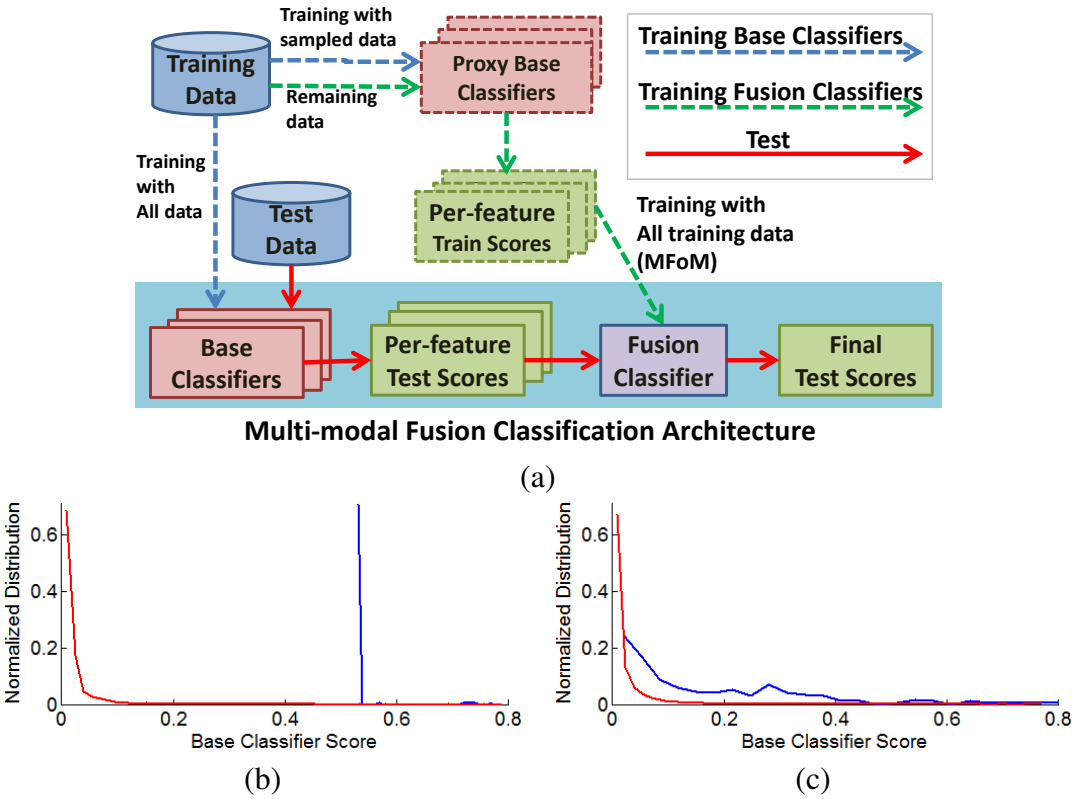


Figure 26. (a) The proposed discriminative score fusion framework, with separate data flows for training and test phases: (b)-(c) Comparison between score distributions from a base classifier on (b) training data seen during learning the base classifier and (c) unseen test data; Blue and red lines indicate distributions of positive and negative samples, respectively. There exists inconsistency between scores in (b) and (c); scores in (b) are unrealistically accurate, and not suitable to be used to train a fusion classifier.

test phase, the classification system at the bottom in Figure 26(a) applies base classifiers on test data to produce per-feature test scores, e.g., audio and video independently. These base classifiers are trained *a priori* using all available training data. Then, these scores are concatenated and used as an input vector to a fusion classifier which produces a single final score.

During training, each base classifier is trained in a one-vs-all manner as well, and is used to generate a single score for the target class. For base classifiers, we used SVMs [99] [1] and their estimated probabilities as base classifier scores.

[1] In theory, MFoM can be used for the training of base classifiers. The hierarchical joint learning is beyond the scope of this work and will be omitted for clarity.

For a fusion classifier, we used MFoM learning scheme and adopted LDF as our class-confidence function in Eq. (19) as $g(x;\Lambda) = \sum_j \omega_j x_j + \omega_0$, where x is the score vector from base classifiers. Accordingly, MFoM systematically learns the weights for each score dimension for the target class, while explicitly optimizing the desired performance metric. This way, the fusion classifier becomes confident when multiple base classifier scores are high, and vice-versa.

In particular, during the training phase, our system divides training data into sets where they are used separately to train proxy base classifiers and a fusion classifier, which is a crucial factor to maintain performance on novel data. By proxy base classifiers, we mean temporarily constructed base classifiers which are learned from a subset of available training data. Then, remaining training data are fed into these proxy base classifiers and per-feature train scores are generated, which are then used as inputs to learn fusion classifiers.

In detail, it is tempting to apply the base classifiers shown at the bottom left of Figure 26(a) on the training data to produce base classifier outputs to be used as training data for the fusion classifier. However, this approach fails to learn an accurate fusion classifier. The reason is that the base classifier has already seen all the training data, accordingly, the generated outputs are unrealistically accurate. For example, Figure 26(b)-(c) show score distributions generated by a base classifier on already seen training data and unseen test data respectively. In particular, Figure 26(b) shows the scores by a base classifier on training data, which are separated very cleanly. Then, Figure 26(c) shows more realistic spread-out score distribution on unseen test data. Because these two distributions are distinct, a fusion classifier learned from the unrealistically accurate scores shown in Figure 26(b) is unlikely to perform well on novel data. Our solution is illustrated in Figure 26(a) as dashed training flows. In detail, training data is divided into N subsets and proxy base classifiers are learned with $(N-1)$ subsets, then used to generate scores on a remaining subset. This procedure is repeated N times to generate scores for entire training data. This way, we can obtain more realistic base classifier outputs to be used to train a fusion classifier. This strategy is

particularly beneficial for datasets with a small number of positive training samples, e.g., the TRECVID MED dataset, since it provides a way to use available training data fully for fusion training.

To improve the performance of the fusion classifier further, we have investigated the use of additional non-target base classifier scores as inputs for 1-vs-All fusion classifiers, and observed consistent improvement in the final fusion classification. For example, we can incorporate the output by a base classifier trained for *Birthday party* for the training of a fusion system for the target class of *Wedding*. In this scheme, our fusion classifier uses $(M \times K)$-dimensional discriminative scores as its inputs, where there are K features and M base classifiers available. Abstractly, base classifier outputs can be regarded as supervised mappings from low-level feature space to score space, and a fusion classifier as a mapping from scores to a confidence score for a target class. The improvement is expected to be obtained because a fusion classifier systematically incorporates the correlation among event classes. Negative correlation as well as positive correlation could be helpful to acquire more discriminant power, i.e., high probabilities of outdoor event classes infer low confidence on indoor event classes. Figure 31 illustrates the learned model parameters of fusion classifiers for the 10 test event classes from TRECVID 2011 MED. The detail of this experimental results, in addition to the comparison of performance with and without the use of non-target scores illustrated in Figure 27 is described in Section 6.3.3.

However, it is also noted that this approach assumes reliability of base classifiers; in other words, additional usage of scores from unreliable base classifiers such as random perturbation could harm fusion learning. Feature selection on the score space would be an interesting topic and left for extended works.

6.3.3 Experiments and Analysis

To measure the usefulness of the proposed approach, we have applied our video retrieval framework on two challenging large-scale consumer video datasets including TRECVID 2011 MED dataset [20] and Columbia Consumer Video (CCV) dataset [16]. Both datasets

have been collected from video sharing websites and the visual contents are unconstrained as most YouTube videos. Both the size and complexity of the datasets are beyond other alternatives such as YouTube Sports [17] or Hollywood datasets [109]. For example, clips were frequently captured under unconstrained lighting and camera motion, exhibiting diverse degrees of encoding artifacts, and heavily edited by owners through shot stitching.

For comparison purposes, the proposed methods is compared against other fusion methods based on logistic regression (LR) and linear SVM, which are standard fusion techniques [47, 42]. In addition, final fusion results are compared with and without non-target base classifier scores, as discussed in Section 6.3.2. For all experiments, performance measure in the form of Eq. (50) has been used, with different values of τ. For the training of comparative approaches, we have assigned the weights equal to τ to positive samples. Operating points were selected on the training performance curves where the specified ratio τ is satisfied. Finally, the performance metrics are computed at the selected operating points.

6.3.3.1 Experiments on TRECVID 2011 MED dataset

The first experiment used the TRECVID 2011 multimedia event detection (MED) corpus [20], which provides an excellent test-bed for real-world video retrieval problems due to its large size (45K video clips) and huge inter- and intra-class content variability. For the MED task, there are 10 annotated event classes: *Birthday party* (E006), *Changing a vehicle tire* (E007), *Flashmob gathering* (E008), *Getting a vehicle unstuck* (E009), *Grooming an animal* (E010), *Making a sandwich* (E011), *Parade* (E012), *Parkour* (E013), *Repairing an appliance* (E014), and *Working on a sewing project* (E015). For each class, there are 150 positive training samples on average, and there are more than 11K purely negative (i.e., not from any of the classes) training video clips. As is clear from the list, the event classes are extremely varied. Moreover, the exemplars within each event class are also extremely varied. The duration of each clip varied significantly (on average, the duration is about 4 minutes); short clips lasts only tens of seconds, while long videos are more than 1 hour. In the test data of 33K clips, there are 120 positive examples (0.4 percent) for each class on

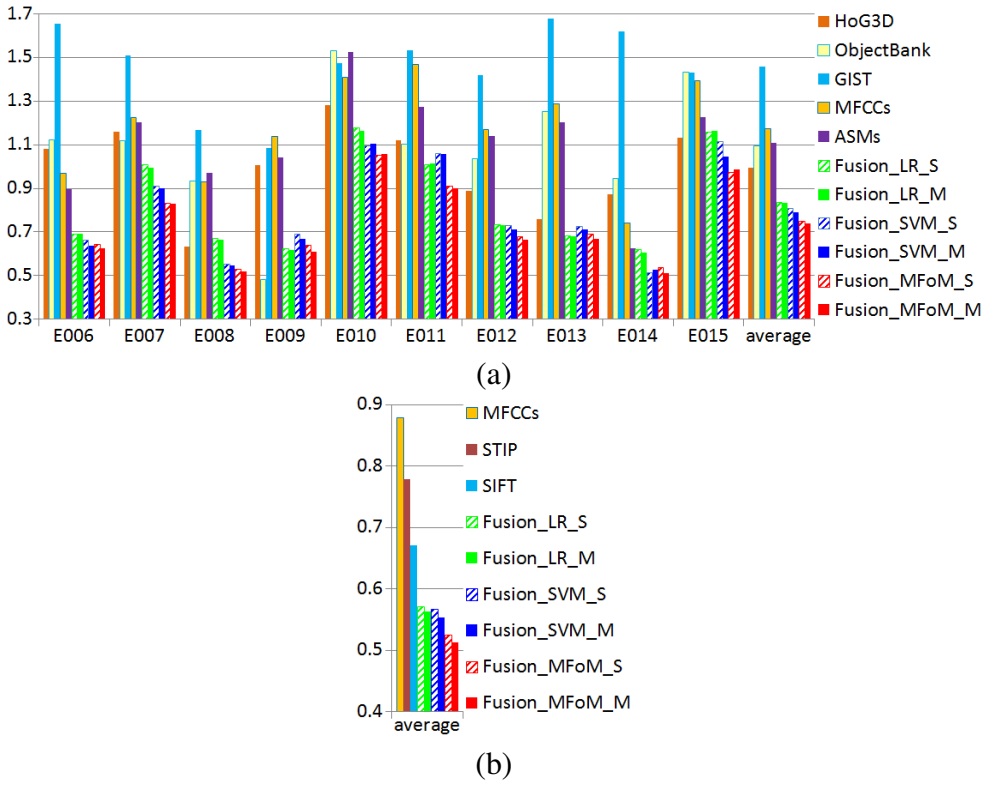


Figure 27. Comparison of performance metrics (lower is better). Results by base classifiers, LR fusion, SVM fusion, and MFoM fusion with only target class scores ('_S') and additional non-target class scores ('_M') are shown; (a) 10 classes and average from the TRECVID 2011 MED dataset and (b) Average of 20 classes from the CCV dataset

average, and approximately 31K videos were purely negative.

Five different types of features are used in our experiments. They include both video and audio features at different granularities: HoG3D [80], Object Bank (OB) [84], GIST [36], MFCCs [87], and acoustic segment models (ASMs) [87]. The features are computed on video segments and aggregated into clip-level features.

The overall performance of compared methods for 10 test classes TRECVID dataset is summarized in Figure 27(a) where lower bars indicate superior performance. For training of different fusion classifiers (MFoM, SVM, LR), identical base classifier scores were used where the results with and without non-target class scores are denoted by postfixes `_M` and `_S` respectively. Performance by individual base classifiers are shown as well. It can be observed that Fusion_MFoM_M (S_τ=0.7374) achieves the best performance consistently

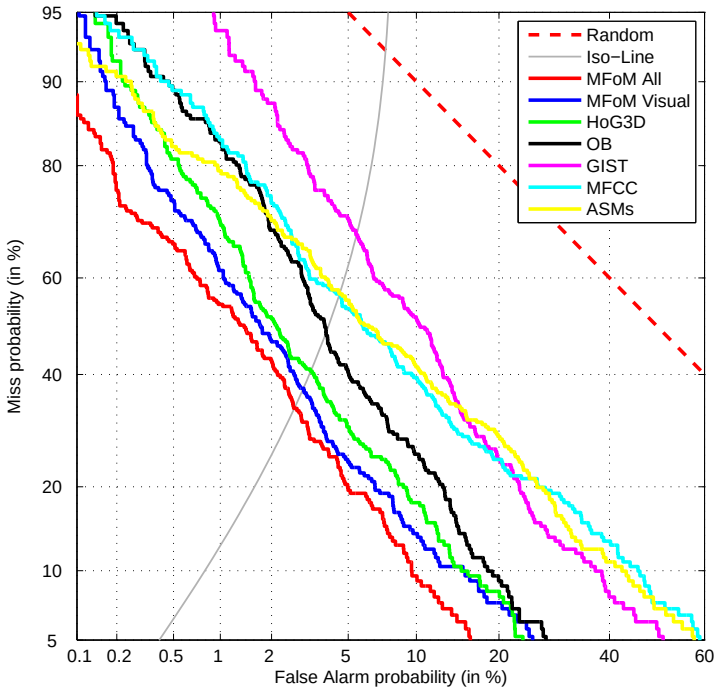


Figure 28. Comparison among the fusion results for three event classes, E007, E008, and E015. MFoM outperforms SVM and LR, especially along with the isoline of $P_{FM} : P_{MD} = 1 : 12.5$.

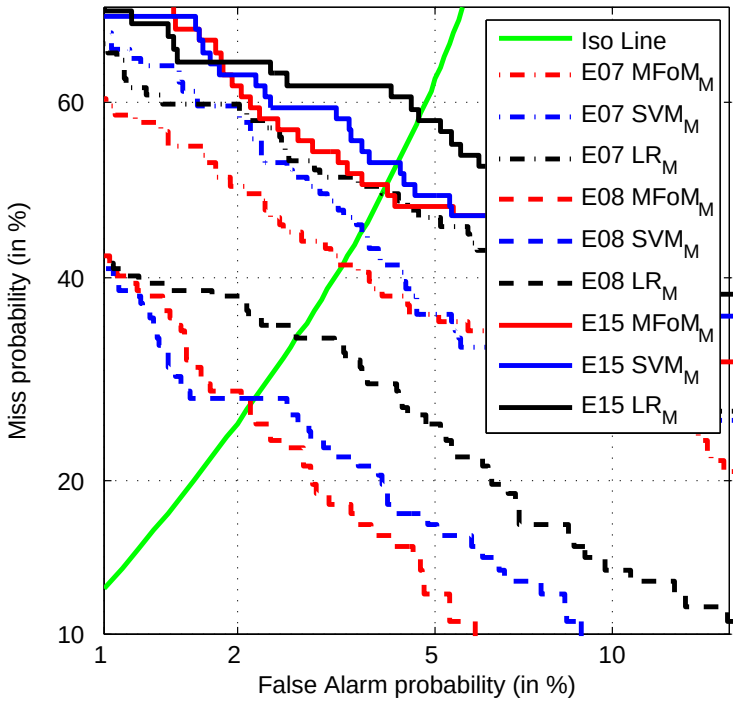


Figure 29. Comparison among the results by the base and fusion classifiers for E012. The proposed MFoM fusion with visual features (blue line) outperforms the individual per-feature base classifiers. The fusion with all the visual and audio features (red line) also shows improvement by audio features.

Figure 30. Top 30 results by the proposed fusion algorithm, top 10 results by HoG3D and OB; sorted from top-left to bottom-right. True positives are marked with green boxes.

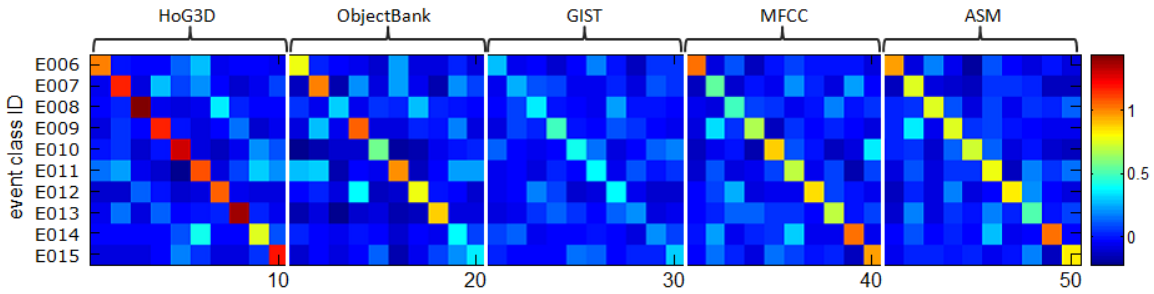


Figure 31. Learned model parameters of LDF for the event classes E006–E015 on the MED dataset. Each row is the 50-dimensional model parameter of one-versus-all fusion classifiers for every event. Each column corresponds to one of 50 base classifiers.

across all events, where it shows meaningful improvement of 12.9% on average, against Fusion_LR_M (S_τ=0.8326) and 7.3% from Fusion_SVM_M (S_τ=0.7916). A similar result holds when using only target-class scores (_S), indicating the generality of our approach.

The benefits of explicit performance metric optimization by our methods can be examined in more detail by looking at the the detection error tradeoff (DET) curves [93] for three test event classes shown in Figure 28. DET curves show error tradeoff of a classification system in the logarithmic-scaled P_{FA} and P_{MD} space. For the three event classes, the DET curves of the proposed MFoM approach (red) is superior or comparable to other events.

Other remaining seven event classes showed similar patterns. However, while MFoM performs better than other methods around the operating point, it is not always better away from the operating point (e.g. E15 (solid) and E07 (dot-dash)). This is not unexpected, since the goal of our approach is to explicitly improve performance at the operating point.

In terms of training parameters, the MFoM fusion classifiers were trained with the following parameters: $\alpha = 30$, and $\gamma = 0.2 \sim 0.4$. γ varies across classes, and was determined by cross-validation. Similar cross validation schemes were used to identify optimal parameters for SVM and LR.

Among the individual features shown in Figure 27(a), HoG3D shows the best performance on average. It is especially competitive for event classes with temporal dynamics, such as *Parkour*. Next, OB is followed, which is competitive for relatively static classes, such as *Making a sandwich*. Notably, audio features provide best performance for audio-rich events such as *Birthday party*.

All the fusion methods consistently outperform per-feature base classifiers, showing the clear benefits of fusion for consumer video retrieval tasks. For example, Figure 29 illustrates the effect of fusion by the proposed algorithm for E012 in the DET plot. It is notable that the fusion of the visual features (blue line) is better than the individual visual features (HoG3D, OB, and GIST). Furthermore, the final fusion result (red line) is more improved by additionally incorporating the audio features.

For a qualitative assessment of the fusion algorithm, Figure 30 shows the top retrieved results for *Getting a vehicle unstuck* from the fusion classifier and two of the base classifiers. The results are sorted from top-left to bottom-right, and true positives are marked with green boxes. It is interesting to see that the two visual features seem complementary. HoG3D captures textures of video scenes as well as temporal dynamics, while OB outputs responses from object detectors in video frames. Accordingly, some of the top results by HoG3D are mainly triggered by only textures of video frames such as roads or plain background, while most of the top results by OB contain a vehicle in the middle of

frames. Combining textures of scenes and responses from object detectors, the fusion results show much better performance that mostly have a vehicle object and a consistency in spatio-temporal dynamics.

The learned model parameters of our MFoM fusion scheme are shown in Figure 31. Each row represents 50-dimensional LDF parameters, which is composed of the weights for the 10-dimensional scores from each feature block. A high positive value indicates strong positive correlation of the corresponding score element to a target class, while a negative value implies a negative correlation. Diagonal structures are observed because base classifiers learned for the same target class are more discriminative, as expected. It is also interesting to see correlations between different event types. For example, the fusion classifier for *making a sandwich* (row 6) shows positive correlation with ObjectBank base classifiers (column 11) for *birthday party*, perhaps because both events frequently occur in dining rooms.

6.3.3.2 Experiments on CCV dataset

As the second dataset, we applied the proposed fusion scheme on Columbia Consumer Video (CCV) dataset [16], which is another publicly available large-scale consumer video dataset. In total, it includes 9,317 consumer videos and is labeled for 20 classes which mostly include complex events such as *ice skating* and *graduation*. In addition, it provides three types of precomputed bag-of-words features for SIFT, STIP [109], and MFCC. There are 180 training positive samples for each class on average.

The identical experiments are conducted on CCV dataset, and the proposed fusion method is compared against SVM and LR. A performance metric of S_τ with $\tau = 10$ was assumed for this evaluation. For all three types of features, base classifiers are learned using HIK SVMs. Then, fusion classifiers were learned on top of the identical base classifiers.

Experimental results on CCV dataset are summarized on Figure 27(b). Patterns identical to the results on TRECVID dataset has been observed for all 20 event classes. For brevity, only the average performance across all classes is shown here. Overall, there is an

average gain of 10.1% and 6.3% achieved by MFoM-based fusion (MFoM_M, S_τ=0.5208), over the LR fusion method (LR_M, S_τ=0.5637) and the SVM fusion method (SVM_M, S_τ=0.5536), respectively.

6.4 Summary

In this chapter, we have presented the novel frameworks that explicitly optimize given performance metrics. First, a novel learning scheme that optimizes a ranking performance measure in an MFoM framework was proposed, with a focus on one of the most widely used ranking performance metrics, AP. We discussed the behavior of AP as a staircase function with respect to each individual sample score. Our approximation scheme for AP gradients showed remarkably reduced computational complexity when compared to the pair-wise ranking approximation. The experimental results on the two challenging datasets showcased the usefulness of the proposed algorithm, while showing a meaningful improvement over a learning scheme maximizing F_1 and significantly reduced learning time over a pair-wise method. The framework is more general and easy to be applied to other ranking performance metrics.

In addition, we showcased an effective approximation scheme for the important class of weighted metrics which can include sub-metrics such as P_{MD} and P_{FA}. The experimental results on two large consumer video archives are promising, and suggest that our approach will add value for real-world computer vision applications with sophisticated user needs.

CHAPTER 7

AN INTEGRATED SYSTEM FOR MULTIMEDIA EVENT DETECTION AND RECOUNTING

Throughout Chapters 4-6, a few techniques for multimedia event detection and recounting have been studied, by separately addressing the challenging issues at different stages of a multimedia retrieval system, especially in large-scale and unconstrained consumer video data. In particular, in Chapter 4, the multi-way local feature pooling method is proposed, by using scene concept analysis. In Chapter 5, it is presented that the per-exemplar learning efficiently address within-class diversity in a complex multimedia event class. Finally, in Chapter 6, efficient fusion methods that explicitly optimize sophisticated performance metrics, which are widely used in measuring the quality of multimedia retrieval system.

In this chapter, an integrated system is proposed, which attempts to take full advantages of the proposed techniques, by using each scheme as a module of an entire system. It is noted that the suggested approach is an example of utilizing the benefits of previously studied techniques, and they can be flexibly deployed for general uses. Extensive experiments have been conducted on the integrated system, by evaluating effects of the implemented sub-routines in contrast to conventional methods.

7.1 Overview of the Integrated System

The suggested integration is summarized in Figure 32, which is based on a hierarchical approach. It is noted that the system is applied to each multimedia event class type. First, once features are extracted from a video clip, the multiple local feature pooling method (MLP), discussed in Chapter 4, is applied, in contrast to a conventional video-level feature pooling method. Then, an HIK-SVM is learned as a base classifier on the constructed feature, while generating a confidence score by a corresponding feature type to a target event class. In addition, assignment vectors regarding multiple scene concepts, generated by Eq. (10), are

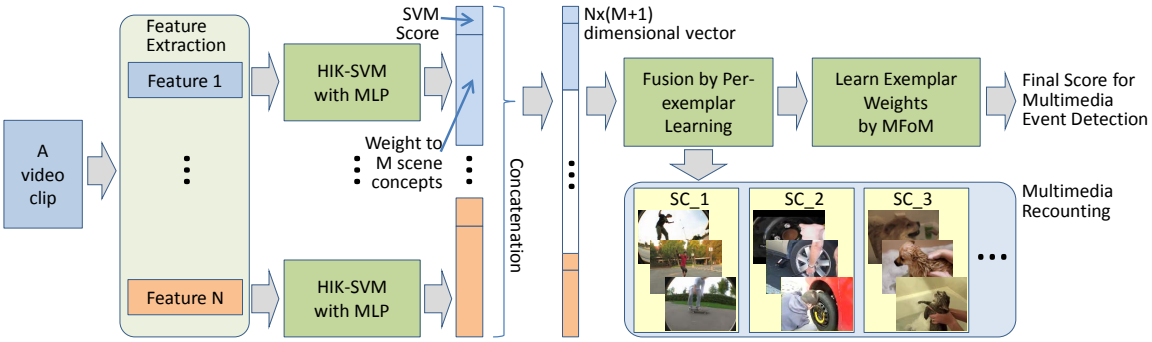


Figure 32. Diagram of the proposed integrated system: it takes the frameworks developed in the previous chapters as sub-routines, and improves the quality of multimedia event detection and recounting.

also collected and concatenated in a vector form with the additional confidence score. For example, if an algorithm involves M scene concepts, an $(M+1)$-dimensional feature vector is constructed for each feature type. The reason of incorporating the soft-assignment values into the feature representation is to further improve multimedia recounting capabilities of the integrated system. In this way, we can take advantages of MLP in representing a video clip and the useful information from correlation between a video clip and constructed scene concepts.

The second module in Fig. 32 is to fuse multiple features by using the per-exemplar learning method, presented in Chapter 5. In particular, the base classifier results from various features are concatenated into a vector form as input for fusion. It is assumed that there are N types of features; accordingly, the dimension of the concatenated vector is $N \times (M+1)$. As studied in Chapter 5, a fusion method based on per-exemplar learning provides a competitive fusion results and additional multimedia recounting capabilities. Since the soft-assignment weights regarding constructed scene concepts are used as elementary distance measures to learn a local distance function of per-exemplar learning, multimedia recounting can be conducted on various scene types (in total, $N \times M$ scene concepts from different feature types) in a video as well as feature types by confidence scores from base classifiers. For multimedia retrieval, learned local distance functions for all training exemplars are delivered to the next module in Fig. 32.

After fusion is made by per-exemplar learning, the proposed MFoM learning frameworks, discussed in Chapter 6, are used to improve the search quality of the integrated system. In particular, effects from different training exemplars toward final retrieval scores are learned in a way to explicitly optimize a given performance metric, in this dissertation, AP or a weighted sum of P_{MD} and P_{FA} at a target error ratio. Let $E = \left\{e_i | e_i \in R^{N\times(M+1)},\ 1 \le i \le N_{trn}\right\}$ and $X = \left\{x_j | x_j \in R^{N\times(M+1)},\ 1 \le j \le N_{tst}\right\}$ be a set of training exemplars and test samples, respectively. Then, a final retrieval score for the j-th test sample x_j is defined as a linear combination of local distance functions learned for training exemplars, provided by the previous fusion sub-routine in per-exemplar learning as following:

$$s_j = \sum_{i=1}^{N_{trn}} \lambda_i D_{e_i}\left(x_j\right), \tag{54}$$

where $D_{e_i}(\cdot)$ is a local distance function learned for the i-th training exemplar e_i, as in Eq. (11). Then, model parameters $\Lambda = \{\lambda_i | 1 \le i \le N_{trn}\}$ are learned by the MFoM learning frameworks, discussed in Chapter 6. In this way, a final retrieval score is recomputed from local distance measures to simulate user needs, characterized by a preferred performance metric.

In all, while the proposed integration scheme involves multiple and complex sub-routines, its objectives are clear: improving the quality of multimedia event detection and recounting, by addressing sophisticated user needs as well as unstructured contents and within-class variability in consumer video data.

7.2 Experiments and Analysis

The proposed integration has been evaluated by using TRECVID 2011 MED data. For feature types, to fully verify the system in various feature modalities, a large array of features including both visual and audio features is considered as following: HoG3D [80], GIST [36], Color SIFT [81], ISA [82], TCH [81], SUN09 [83], ObjectBank [84], MFCCs, and ASMs [86]. The detail of the features can be found in Chapter 3. It is noted that MLP is only applied to HoG3D, GIST, ObjectBank, MFCCs, and ASMs, since the other feature

Table 10. Comparison of per-exemplar learning(PEL) with and without scene concept assignments in mAP (%).

eventID	Chance	PEL /wo SC assignments	PEL /w SC assignments
E006	0.54	34.66	**34.76**
E007	0.35	37.54	**38.34**
E008	0.42	**60.66**	60.40
E009	0.26	35.20	**36.26**
E010	0.25	12.74	**12.82**
E011	0.43	15.82	**16.01**
E012	0.58	34.43	**34.88**
E013	0.32	41.70	**42.28**
E014	0.27	51.37	**51.62**
E015	0.26	20.73	**21.81**
mAP	0.37	34.49	**34.92**

types are extremely sparse, or frame-level features are not available. For such features, base classifiers are trained on a clip-level feature with HIK-SVM, and only confidence scores are delivered to the fusion sub-routine.

7.2.1 Multimedia Event Detection

To evaluate the effects of the sub-routines implemented in the proposed integration, the following experiments were conducted: (1) per-exemplar fusion learning with vs. without scene concept assignments, and (2) association-based fusion scores by per-exemplar learning only vs. scores recomputed by MFoM that optimizes a domain specific performance metric, defined in Eq (54). It is noted that the usefulness of MLP per feature type has already been studied in Section 4.2.

First, in Table 10, the per exemplar learning methods with and without scene concept assignments are compared in mAP (%). We can observe consistent improvement by per-exemplar learning with scene concept assignments over that without scene concept assignments. While the improvement is not very significant, the use of scene concept assignments additionally improves multimedia recounting capabilities, discussed in the following section. Furthermore, since the soft assignment values to scene concepts for a video clip are already computed during learning base classifiers, no additional computation is required,

Table 11. Comparison of average performance of fusion scores learned by association-based per-exemplar learning, and recomputed by MFoM-AP and MFoM-S_τ (S_τ can be found in Eq. (50)). For brevity, mean performance across the 10 event classes, mean P_{FA} @ a target error ratio (TER) (%) and mAP (%), is presented.

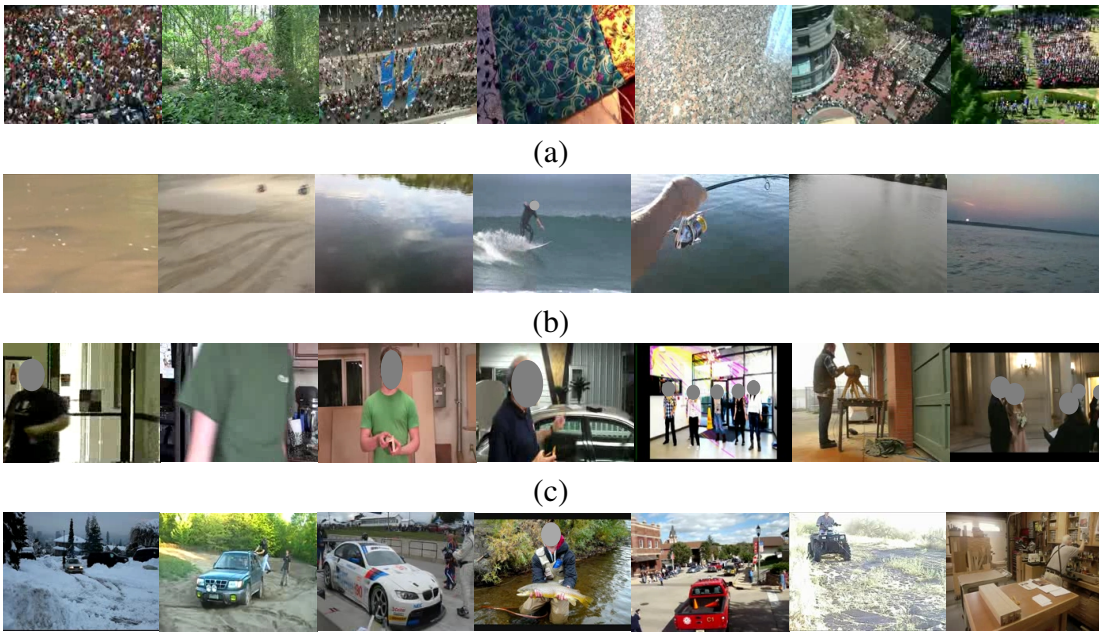
	Chance	Assoc.-based PEL	MFoM-AP	MFoM-S_τ
mean P_{FA} @ a TER	7.41	2.46	2.45	**2.17**
mAP	0.37	34.92	**36.34**	34.15

which makes it appealing to include the scene concept assignments in per-exemplar learning.

The second experiment was conducted to verify effects of fusion score re-computation by MFoM learning. As two performance metrics were discussed in Chapter 6, namely AP and a weighted sum of P_{FA} and P_{MD} at a target error ratio (TER), two MFoM schemes, which are denoted as MFoM-AP and MFoM-S_τ, respectively, were compared with the association-based per-exemplar learning. The results are summarized in Table 11, in the corresponding performance metrics. It is noted that P_{FA} at a TER ($P_{FA} : P_{MD} = 1 : 12.5$, as suggested by [20]) delivers the same quality of a weighted sum of P_{FA} and P_{MD} at a target error ratio (TER); however, it is more compactly presented. The results clearly demonstrate the benefits of MFoM learning in recomputing fusion scores at the back-end of the proposed system: MFoM-S_τ shows superior performance in mean P_{FA} @ a TER, while MFoM-AP can improve mAP.

7.2.2 Multimedia Recounting

The proposed system provides further improved multimedia recounting capabilities beyond the per-exemplar learning on discriminative element distance measures, discussed in Chapter 5. Such improvement is particularly enabled by using a rich set of scene concepts. In Figure 33, examples of video segments for corresponding scene concepts by HoG3D are presented. Across the presented scene concepts, common spatio-temporal visual aspects have been found, which are imposed by HoG3D. For example, in Figure 33(a), visual contents in the video segments look similar. These segments are included in one scene concept,



(a)

(b)

(c)

(d)

Figure 33. Examples of video segments for corresponding scene concepts by HoG3D: (a) complex textures (often involving crowd), (b) plain region, (c) human standing, and (d) a big square-shaped object.

which involves complex textures, e.g., crowd scenes. On the other hand, we can observe that video segments in Figure 33(b) convey plain region such as sand or see. In Figure 33(c), most video segments include a human standing in the middle of video scenes, while video segments in Figure 33(d) contain a big square-shaped object, mostly a vehicle.

As studied in Section 5.3, the per-exemplar learning method generates a unique weight vector on an elementary distance, which is measured by a similarity with respect to scene concepts in the integrated system, per training exemplar. It is expected that a retrieved video sample can be reasoned with the associated training exemplars in terms of its relevance to the exemplars by referring to a rich set of scene concepts.

CHAPTER 8

CONCLUSION AND FUTURE WORK

This dissertation discussed the schemes that effectively improve the quality of multimedia event detection systems by addressing challenging issues that have not been intensively studied in previous research. Such challenging issues include the sparseness of strong evidence in unstructured video data, the within-class content variability in complex multimedia event categories, and the necessity to design a detection model that explicitly optimizes a domain-specific performance metric to simulate user experiences. Through extensive experiments by comparison with many state-of-the-art schemes, the usefulness of the proposed schemes is verified.

8.1 Summary of the Research in this Dissertation

The research presented in this dissertation is summarized as following. First, in Chapter 3, various features for representing multimedia data are discussed. The detailed methods to construct feature vectors from audio/visual and low/high-level features are presented, along with their capabilities to describe various types of information in multimedia data. In addition, non-linear kernels, which have been widely used in the communities of computer vision and multimedia processing, are evaluated for the tasks. The extensive comparison among the features provides useful information to efficiently utilize them for video representations.

In Chapter 4, a segmental multi-way local feature pooling method by using scene concept analysis is proposed. This scheme demonstrated benefits over conventional methods by constructing clip-level representations via average-based global pooling. The key idea of the framework is to utilize similarities between two videos in terms of various scene concepts and to improve a discriminative power by using kernelization techniques. In particular, the proposed method utilizes scene concepts that are pre-constructed by clustering

video segments into categories in an unsupervised manner. Then, a video is represented with multiple feature descriptors with respect to scene concepts. Finally, multiple kernels are constructed from the feature descriptors, and then, are combined into a final kernel that improves the discriminative power for multimedia event detection. This method is evaluated on TRECVID 2011 MED data with an extensive comparison to other state-of-the-art methods. The experimental results showcased the usefulness of the proposed multi-way local feature pooling method on widely used visual and audio features.

In Chapter 5, a per-exemplar learning scheme is proposed with a focus on fusing multiple types of heterogeneous features for video retrieval. While the conventional approach for multimedia retrieval involves learning a single classifier per category, the proposed scheme learns multiple detection models, one for each training exemplar. In particular, a local distance function is defined as a linear combination of element distance measured by each features. Then, a weight vector of the local distance function is learned in a discriminative learning method by taking only neighboring samples around an exemplar as training samples. In this way, a retrieval problem is redefined as an association problem, i.e., test samples are retrieved by association-based rules. In addition, it is shown that the proposed per-exemplar learning scheme can enable a rich set of recounting capabilities, where the rationale for each retrieval result can be automatically described to users in order to aid their interaction with the system. The algorithm is verified on challenging consumer video corpora, the TRECVID 2011 MED and CCV data, while showing competitive fusion performance compared to other state-of-the-art fusion methods. Moreover, multimedia event recounting capabilities are demonstrated on real video examples.

In Chapter 6, in MFoM learning, novel algorithms were proposed to explicitly optimize two challenging metrics, AP and a weighted sum of P_{MD} and P_{FA} at a target error ratio. Most conventional learning schemes attempt to optimize their own learning criteria, as opposed to domain-specific performance measures. By addressing this discrepancy, the proposed learning scheme approximates the given performance measure, which is discrete

and makes it difficult to apply conventional optimization schemes, with a continuous and differentiable loss function which can be directly optimized. Then, a GPD algorithm is applied to optimizing this loss function. In particular, a key contribution in Chapter 6 is extending the MFoM learning to the two challenging metrics, which are complex compared to simple error metrics, e.g., precision or F-scores. Optimizing AP was evaluated in an AIA problem, and minimizing a weighted sum of P_{MD} and P_{FA} at a target error ratio was verified on a fusion problem for multimedia retrieval. For both studies, experiments were conducted on large-scale image/video data, along with extensive comparison with state-of-the-art methods. The experimental results are appealing, while suggesting the usefulness of the proposed algorithm for multimedia retrieval.

Finally, in Chapter 7, an integrated framework, via taking advantage of the aforementioned schemes, is discussed. The extensive experimental results demonstrate the advantages of each scheme, while the full advantage has been achieved by using a combination of all proposed schemes. It is noted that the integrated framework involves an exemplar usage of the proposed schemes, which are more flexible for general uses.

8.2 Avenues for Future Work

Although this dissertation presents a novel framework for multimedia event detection and recounting, there still exists room for further improvements. One immediate future research venue concerns investigating more feature types, which can provide further information embedded in consumer videos. It has been observed that most features from different granularities have complementary properties, i.e., a system usually improves, when additional features are combined by a fusion scheme. Such features may include high-level semantic features beyond low-level features, which are primarily used in this thesis. However, extracting meaningful and robust high-level semantic features still seems challenging due to unconstrained contents in consumer videos.

Another future work worthy of consideration is to extend the proposed per-exemplar

learning to incorporating low-level features beyond base classifier scores. While base classifier scores provide useful discriminative distance measures, multimedia recounting capabilities can be further improved by directly using low-level features. For example, within an image feature, local distances in the low-level feature spaces can provide more detailed recounting information, by suggesting specific scene-types. For this purpose, a compact description of the low-level features needs to be studied, e.g., features selection or dimension reduction, since most low-level features are represented in a BoW descriptor with thousands codewords and may introduce bias issues in high-dimensional spaces.

In addition, we can also investigate extended uses of the proposed fusion learning scheme beyond a linear combination of base classifier scores. For example, fusion by the geometric mean has been found to show competitive performance when compared to a linear discriminant function. This investigation will naturally involve the study of effective score normalization methods. The distribution of base classifier scores can significantly differ by feature/classifier types. In such cases, learning only linear weights on these scores might not address the large variance of score distributions, and accordingly, will generate a less optimal solution. A normalization scheme can be defined in a parameterized model, and then, learned while training a fusion classifier.

In all, this dissertation involves a valuable discussion on multimedia event detection and recounting. It is hoped that the research presented in this dissertation contributes to further developments in this area.

BIBLIOGRAPHY

[1] "YouTube statistics," http://www.youtube.com/yt/press/statistics.html, Mar. 2013, [Online; accessed Mar. 1, 2013].

[2] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[3] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith, "Optimal multimodal fusion for multimedia data analysis," in *Proceedings of ACM Multimedia*, 2004.

[4] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "A benchmark database and an evaluation of human and machine performance," in *Proceedings of ACM International Conference on Multimedia Retrieval*, 2011.

[5] S. Gao, W. Wu, and C.-H. Lee, "A MFoM learning approach to robust multiclass multi-label text categorization," in *Proceedings of International Conference on Machine Learning*, 2004.

[6] N. V. Patel and I. K. Sethi, "Video shot detection and characterization for video databases," *Pattern Recognition*, vol. 30, pp. 583–592, Apr. 1997.

[7] W. Xiong and J. Lee, "Efficient scene change detectionan dcamera motion annotation for video classification," *Computer Vision and Image Understanding*, vol. 71, pp. 166–181, Aug. 1998.

[8] Y.-G. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S.-F. Chang, "Columbia-UCF TRECVID2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching," in *Proceedings of TRECVID Workshop*, 2010.

[9] A. G. A. Perera, S. Oh, M. Leotta, I. Kim, B. Byun, C.-H. Lee, S. McCloskey, J. Liu, B. Miller, Z. F. Huang, A. Vahdat, W. Yang, G. Mori, K. Tang, D. Koller, F.-F. Li, K. Li, G. Chen, J. Corso, Y. Fu, and R. Srihari, "GENIE TRECVID2011 multimedia event detection: Late-fusion approaches to combine multiple audio-visual features," in *Proceedings of TRECVID Workshop*, 2011.

[10] L. Kennedy and A. Hauptmann, "LSCOM lexicon definition and annotation version 1.0," in *Proceedings of DTO Challenge workshop on large scale concept ontology for multimedia*, 2006.

[11] A. G. Hauptmann, M. G. Christel, and R. Yan, "Video retrieval based on semantic concepts," pp. 602–622, 2008.

[12] C. G. M. Snoek, M. Worring, J. C. van Gemert, J. M. Geusebroek, and A. W. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proceedings of ACM Multimedia*, 2006.

[13] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proceedings of International Conference on Computer Vision*, 2005.

[14] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proceedings of International Conference on Pattern Recognition*, 2004.

[15] S. Zanetti, L. Zelnik-Manor, and P. Perona, "A walk through the web's video clips," in *Proceedings of Computer Vision for Pattern Recognition Workshop on Internet Vision*, 2008.

[16] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: a benchmark database and an evaluation of human and machine performance," in *Proceedings of International Conference on Multimedia Retrieval*, 2011.

[17] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *Proceedings of Computer Vision and Pattern Recognition*, 2009.

[18] J. Liu, Y. Yang, and M. Shah, "Learning semantic visual vocabularies using diffusion distance," in *Proceedings of Computer Vision and Pattern Recognition*, 2009.

[19] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proceedings of ACM Multimedia Information Retrieval*, 2006.

[20] "2012 TRECVID multimedia event detection track," http://www.nist.gov/itl/iad/mig/med12.cfm, Aug. 2012, [Online; accessed Aug. 21, 2012].

[21] A. G. A. Perera, M. P. S. Oh, T. Ma, A. Hoogs, A. Vahdat, K. Cannons, G. Mori, S. McCloskey, B. Miller, S. Venkatesh, P. Davalos, P. Das, C. Xu, J. Corso, R. Srihari, I. Kim, Y. Cheng, Z. Huang, C. Lee, K. Tang, F.-F. Li, and D. Koller.

[22] H. Cheng, J. Liu, S. Ali, O. Javed, Q. Yu, A. Tamarakar, A. Divakaran, H. S. Sawhney, R. Manmatha, J. Allan, A. Hauptmann, M. shah, S. Bhattacharya, A. Dehghan, G. Friedland, B. Martinez Elizalde, T. Darrel, M. Witbrock, and J. Curtis, "[sri-sarnoff aurora."

[23] P. Natarajan, S. Wu, X. Zhuang, A. Vazquez-Reina, S. N. Vitaladevuni, K. Tsourides, C. Andersen, R. Prasad, G. Ye, D. Liu, S. Chang, I. Saleemi, M. Shah, Y. Ng, B. White, A. Gupta, and I. Haritaoglu, "BBNVISER: BBNVISER TRECVID 2012 multimedia event detection and multimedia event recounting systems," in *Proceedings of TRECVID workshop*, 2012.

[24] N. C. L. Cao, L. Gong, M. Hill, G. Hua, M. Merler, J. R. Smith, S. Chang, C. Cotton, D. Ellis, Y. Mu, F. X. Yu, and J. Kender.

[25] M. Akbacak, R. C. Bolles, J. B. Burns, M. Eliot, A. Heller, J. A. Herson, G. K. Myers, R. Nallapati, S. Pancoast, J. V. Hout, E. Yeh, A. Habibian, D. C. Koelma, Z. Li, M. Mazloom, S. Pintea, K. E. van de Sande, A. W. Smeulders, C. G. Snoek, S. C. Lee, R. Revatia, P. Sharma, C. Sun, and R. Trichet, "The 2012 SESAME multimedia event detection system," in *Proceedings of TRECVID workshop*, 2012.

[26] K. Grauman and T. Darrel, "Pyramid match kernels: Discriminative classification with sets of image features," in *Proceedings of International Conference on Computer Vision*, 2005.

[27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," 2006.

[28] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *Transaction on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 591–605, Apr. 2009.

[29] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of Computer Vision and Pattern Recognition*, 2008.

[30] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *Transaction on Pattern Analysis and Machine Intelligence*, vol. 32, Sept. 2010.

[31] Y. Wang and G. Mori, "A discriminative latent model of image region and object tag correspondence," in *Proceedings of Advances in Neural Information Processing Systems*, 2010.

[32] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Proceedings of Advances in Neural Information Processing Systems*, 2003.

[33] J. C. Niebles, C.-W. Chen, and F.-F. Li, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proceedings of European Conference on Computer Vision*, 2010.

[34] K. Tang, F.-F. Li, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proceedings of Computer Vision for Pattern Recognition*, 2012.

[35] L. Cao, Y. Mu, A. Natsev, S.-F. Chang, G. Hua, and J. R. Smith, "Scene aligned pooling for complex video recognition," in *Proceedings of European Conference on Computer Vision*, 2012.

[36] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, pp. 145–175, May 2001.

[37] A. Frome and Y. Singer, "Image retrieval and classification using local distance functions," in *Proceedings of Advances in Neural Information Processing Systems*, 2006.

[38] T. Malisiewicz and A. A. Effros, "Recognition by association via learning per-exemplar distance," in *Proceedings of Computer Vision and Pattern Recognition*, 2008.

[39] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proceedings of International Conference on Computer Vision*, 2011.

[40] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *Transaction on Pattern Analysis and Machine Intelligence*, June 1996.

[41] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proceedings of Computer Vision and Pattern Recognition*, 2006.

[42] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li, "YouTubeCat: Learning to categorize wild web videos," in *Proceedings of Computer Vision and Pattern Recognition*, 2010.

[43] G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik, "Finding meaning on YouTube: Tag recommendation and category discovery," in *Proceedings of Computer Vision and Pattern Recognition*, 2010.

[44] W. Yang and G. Toderici, "Discriminative tag learning on YouTube videos with latent sub-tags," in *Proceedings of Computer Vision and Pattern Recognition*, 2011.

[45] P. Natarajan, P. Natarajan, V. Manohar, S. Wu, S. Tsakalidis, S. N. Vitaladevuni, X. Zhuang, R. Prasad, G. Ye, D. Liu, I.-H. Jhuo, S.-F. Chang, H. Izadinia, I. Saleemi, M. Shah, B. White, T. Yeh, and L. Davis, "BBN VISER TRECVID 2011 multimedia event detection system," in *Proceedings of TRECVID workshop*, 2011.

[46] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of International Conference on Computer Vision*, 2007.

[47] P. V. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proceedings of International Conference on Computer Vision*, 2009.

[48] D.-H. Wang, S. Gao, Q. Tian, and W.-K. Sung, "Discriminative fusion approach for automatic image annotation," in *Proceedings of International Workshop on Multimedia Signal Processing*, 2005.

[49] S. Gao, D.-H. Wang, and C.-H. Lee, "Automatic image annotation through multi-topic text categorization," in *Proceedings of Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2006.

[50] W. Scheirer, A. Rocha, R. Micheals, and T. Boult, "Robust fusion: extreme value theory for recognition score normalization," in *Proceedings of European Conference on Computer Vision*, 2010, pp. 481–495.

[51] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recogn.*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005.

[52] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of International Conference on Machine Learning*, 2005.

[53] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[54] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *Proceedings of Computer Vision for Pattern Recognition*, 2012.

[55] O. R. Terrades, E. Valveny, and S. Tabbone, "Optimal classifier fusion in a non-bayesian probabilistic framework," *Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, Sep. 2009.

[56] J. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *Proceedings of International Converence on Multimedia and Expo*, 2003.

[57] J. Liu, S. McCloskey, and Y. Liu, "Local expert forest of score fusion for video event classification," in *Proceedings of European Conference on Computer Vision*, 2012.

[58] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactinos on Speech and Audio Processing*, vol. 5, May 1997.

[59] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua, "A maximal figure-of-merit learning approach to text categorization," in *Proceedings of Special Interest Group on Information Retrieval*, 2003.

[60] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a family of design algorithm based upon the generalized probabilistic descent method." *Proceedings of the IEEE*, pp. 2345–2373, Nov. 1998.

[61] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgments," in *Proceedings of ACM International Conference on Information and Knowledge Management*, 2006.

[62] "2011 TRECVID Multimedia Event Detection Evaluation Plan Version 3.0." http://www.nist.gov/itl/iad/mig/upload/MED11-EvalPlan-V03-20110801a.pdf.

[63] I. Kim and C.-H. Lee, "Optimization of average precision with maximal figure-of-merit learning," in *Proceedings of International Workshop on Machine Learning for Signal Processing*, 2011.

[64] I. Kim, S. Oh, B. Byun, A. G. A. Perera, and C.-H. Lee, "Explicit performance metric optimization for fusion-based video retrieval," in *Proceedings of European Conference on Computer Vision Workshop on Information Fusion in Computer Vision for Concept Recognition*, 2012.

[65] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of SIG KDD*, 2002.

[66] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," in *Journal of Machine Learning Research*, vol. 4, 2003, pp. 933–969.

[67] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of International Conference on Machine Learning*, 2005.

[68] S. Gao, C.-H. Lee, and J. H. Lim, "An ensemble classifier learning approach to ROC optimization," in *Proceedings of International Conference on Pattern Recognition*, 2006.

[69] S. Gao and Q. Sun, "Improving semantic concept detection through optimizing ranking function," vol. 9, pp. 1430–1442, 2007.

[70] T. Fawcett, "Introduction to roc analysis," *Pattern Recognition Letters*, pp. 861–874, June 2006.

[71] B. McFee, "Metric learning to rank," in *Proceedings of International Conference on Machine Learning*, 2010.

[72] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: from pairwise approach to listwise approach," in *Proceedings of International Conference on Machine Learning*, 2007.

[73] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li, "Listwise approach to learning to rank - theory and algorithm," in *Proceedings of International Conference on Machine Learning*, 2008.

[74] C. Ma and C.-H. Lee, "An efficient gradient computation approach to discriminative fusion optimization in semantic concept detection," in *Proceedings of International Conference on Pattern Recognition*, 2008.

[75] L. Wang, J. Lin, and D. Metzler, "Learning to efficiently rank," in *Proceedings of ACM SIGIR*, 2010, pp. 138–145.

[76] B. B. Cambazoglu, H. Zaragoza, O. Chapelle, J. Chen, C. Liao, Z. Zheng, and J. Degenhardt, "Early exit optimizations for additive machine learned ranking systems," in *Proceedings of ACM WSDM*, 2010.

[77] S. Yaman, M. Siniscalchi, and C.-H. Lee, "A multi-objective programming-based approach to language model adaptation," in *Proceedings of Advances in Neural Information Processing Systems*, 2009.

[78] I. B. Vapnyarskii, *Encyclopedia of Mathematics*. Springer, 2001, ch. Lagrange Multiplier.

[79] S. Oh, S. McCloskey, I. Kim, A. Vahdat, K. Cannons, H. Hajimirsadeghi, G. Mori, A. G. A. Perera, M. Pandey, and J. J. Corso, "Multimedia event detection and recounting with multimodal feature fusion and temporal concept localization," *Journal of Machine Vision and Applications*, 2013.

[80] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *Proceedings of British Machine Vision Conference*, 2008.

[81] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *Transactions on Pattern Analysis and Machien Intelligence*, vol. 32, Sept. 2010.

[82] Q. Le, W. Zou, S. Yeung, and A. Ng, "Learning hierarchical spatio-temporal features for action recognition with independent subspace analysis," in *Proceedings of Computer Vision and Pattern Recognition*, 2011.

[83] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "SUN Database: Large-scale Scene Recognition from Abbey to Zoo," 2010.

[84] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li, "Object bank: A high-level image representatino for scene classification and semantic feature sparsification," in *Advances in Neural Information Processing Systems*, 2010.

[85] K. Lee and D. P. W. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE Transactions on Audio, Speech, and Language Processing*, August 2010.

[86] B. Byun, I. Kim, S. M. Siniscalchi, and C.-H. Lee, "Consumer-level multimedia event detection through unsupervised audio signal modeling," in *Proceedings of Conference of the International Speech Communication Association*, 2012.

[87] C.-H. Lee, F. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 1988.

[88] J. Reed and C.-H. Lee, "On the importance of modeling temporal information in music tag annotation," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2009.

[89] Y. Tsao, H. Sun, H. Li, and C.-H. Lee, "An acoustic segment model approach to incorporating temporal information into speaker modeling for text-independent speaker recognition," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[90] T. Jebara and R. Kondor, "Bhattacharyya and expected likelihood kernels," in *Proceedings of Conference on Learning Theory*. press, 2003.

[91] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machies is efficient," in *Proceedings of Computer Vision for Pattern Recognition*, 2008.

[92] D. Zhang, X. Chen, and W. S. Lee, "Text classification with kernels on the multinomial manifold," in *Proceedings of Special Interest Group on Information Retrieval*, 2005.

[93] A. F. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proceedings of Eurospeech*, 1997.

[94] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan, "Multimodal feature fusion for robust event detection in web videos," in *Proceedings of Computer Vision for Pattern Recognition*, 2012.

[95] Y.-L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition," in *Proceedings of International Conference on Computer Vision*, 2011.

[96] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Proceedings of International Conference on Machine Learning*, 2009.

[97] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *Journal of Machine Learning Research*, March 2011.

[98] I. Kim, S. Oh, A. G. A. Perera, and C.-H. Lee, "Per-exemplar fusion learning for video retrieval and recounting," in *Proceedings of International Conference on Multimedia and Expo*, 2012.

[99] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199

[100] A. Kumar, A. Niculescu-Mizil, K. Kavukcoglu, and H. Daume, "A binary classification framework for two-stage multiple kernel learning," in *Proceedings of International Conference on Machine Learning*, 2012.

[101] P. Over, G. Awad, J. Fiscus, M. Michel, A. F. Smeaton, and W. Kraaij, "TRECVID 2009-goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID Workshop*, 2009.

[102] B. Byun and C.-H. Lee, "A kernelized maximal figure-of-merit learning approach based on subspace distance minimization," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2011.

[103] B. Byun, C.-H. Lee, S. Webb, and C. Pu, "A discriminative classifier learning approach to image modeling and spam image identification," in *Proceedings of Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, 2007.

[104] L. Yan, R. Dodier, M. C. Mozer, and R. Wolniewicz, "Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic," in *Proceedings of International Conference on Machine Learning*, 2003.

[105] H. Watanabe, J. Tokuno, T. Ohashi, S. Katagiri, and M. Ohsaki, "Minimum classification error training with automatic setting of loss smoothness," in *Proceedings of International Workshop on Machine Learning for Signal Processing*, 2011.

[106] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Springer, 2006, pp. 102–117.

[107] "Guidelines for the TRECVID 2006 evaluation," http://www.nist.gov/itl/iad/mig/upload/MED11-EvalPlan-V03-20110801a.pdf, 2006, [Online; accesed Aug. 21, 2012].

[108] J. R. Bellegarda, "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, pp. 1279–1296.

[109] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proceedings of Computer Vision for Pattern Recognition*, 2008.