THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

EFFECTIVE MULTIMEDIA EVENT ANALYSIS IN LARGE-SCALE VIDEOS

Litao Yu

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2016

School of Information Technology & Electrical Engineering

Abstract

In the last decade, video content analysis has attracted increasing research interest in the fields of multimedia and computer vision. With the explosive growth of videos on the web and other multimedia sources, it is important for many applications to build effective models that can help us automatically analyse the videos. Among different video content analysis tasks, event detection, recognition, recounting and retrieval in unconstrained cases are the most challenging, because events often consist of miscellaneous spatial-temporal semantics such as various objects, human actions and scenes. In order to better analyse the events contained in videos, scholars have tried to either design powerful visual features or build effective models. However, several technical issues have not yet been well addressed. These include, for example, how to reduce the computational complexity of the hash model training procedure for video event retrieval when given more training videos, how to integrate the spatial and temporal information well in videos for event detection, and how to make use of contextual information to enhance the model training. This thesis focuses on building effective and efficient models for video event detection, recognition and retrieval, and it contains the following four parts:

The first part aims to design a generic model *Max-margin adaptive model* (MMA) for video pattern recognition. The MMA model adopts the advantages of semi-supervised learning and transfer learning, which can utilize both labelled and unlabelled videos for model training. It considers the data distribution consistency between labelled videos and unlabelled auxiliary videos from a statistical perspective by learning an optimal mapping function. It also broadens the geometric margin between positive-labelled videos and negative-labelled videos to improve the robustness of the model.

The second part aims to build a deep spatial-temporal model for multimedia event detection (MED). In our setting, each video follows a multiple instance assumption, where its visual segments contain both spatial and temporal properties of events. Regarding these properties, we try to implement the MED system by a two-step deep training model: unsupervised recurrent video reconstruction and supervised fine-tuning, to improve the generality of the model and boost the event detection accuracy.

In the third part, we propose a context based framework for web video event recognition.

Different from content based video recognition tasks, our proposed framework considers the properties of both video content and web documents. Web videos often describe large-granular events and carry very limited textual information. In this work we first construct an event knowledge base by deeply mining the semantic information from web documents, then propose a *Two-view adaptive regression* model (TVAR) that explores the intrinsic correlation between the visual and textual cues of the web videos to learn reliable classifiers.

In the fourth part, we set out a hashing model *Visual State Binary Embedding* (VSBE) for scalable video event retrieval. The VSBE model can preserve the essential semantic information of the videos in binary codes to ensure effective retrieval performance. Compared with other video binary embedding models, one advantage of our proposed method is it only needs a limited number of key frames from the training videos for hash model training, so the computational complexity is much lower in the training phase. At the same time, we apply the pair-wise constraints generated from the visual states to sketch the local properties of the events at the semantic level, so accuracy is also guaranteed.

Declaration by Author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

- Litao Yu, Shao Jie, Xin-Shun Xu, and Heng Tao Shen *Max-margin adaptive model for complex video pattern recognition*. Multimedia Tools and Applications, Vol. 74, No. 2 (2015): 505-521.

- Litao Yu, Yang Yang, Zi Huang, Peng Wang and Heng Tao Shen *Web video event recognition by semantic analysis from ubiquitous documents*. IEEE Transactions on Image Processing. (Subbmitted on 03/03/2016 for second review)

- Litao Yu, Zi Huang, Jiewei Cao and Heng Tao Shen *Scalable video event retrieval by visual state binary embedding*. IEEE Transactions on Multimedia, Vol. 18, No. 8 (2016): 1590-1603.

- Litao Yu, Xiaoshuai Sun and Zi Huang *Robust spatial-temporal deep model for multimedia event detection*. Neurocomputing. (Accepted on 15/03/2016)

Publications included in this thesis

Litao Yu, Jie Shao, Xin-Shun Xu, and Heng Tao Shen. *Max-margin adaptive model for complex video pattern recognition*. Multimedia Tools and Applications, Vol. 74, No. 2 (2015): 505-521. -incorporated as Chapter 3.

Contributor	Statement of contribution
Litao Yu	Experiment design and conduction (100%) Paper writing (70%)
Jie Shao	Proof reading
Xin-shun Xu	Proof reading
Heng Tao Shen	Paper writing (30%)

Litao Yu, Xiaoshuai Sun and Zi Huang *Robust spatial-temporal deep model for multimedia event detection*. Neurocomputing. (Accepted on 15/03/2016) -incorporated as Chapter 4.

Contributor	Statement of contribution
Litao Yu	Experiment design and conduction (90%) Paper writing (90%)
Xiaoshuai Sun	Experiment design and conduction (10%) Paper writing (10%)
Zi Huang	Proof reading

Litao Yu, Zi Huang, Jiewei Cao and Heng Tao Shen *Scalable video event retrieval by visual state binary embedding*. IEEE Transactions on Multimedia, Vol. 18, No. 8 (2016): 1590-1603. -incorporated as Chapter 6.

Contributor	Statement of contribution
Litao Yu	Experiment design and conduction (90%) Paper writing (80%)
Zi Huang	Paper writing (20%)
Jiewei Cao	Experiment design and conduction (10%)
Heng Tao Shen	Paper writing (10%)

Contributions by others to the thesis

For all the published research work included in this thesis, Prof. Heng Tao Shen, as my principle advisor, has provided very helpful insight into the overall as well as the technical details and research problems; guidance for problem formulation as well as constructive comments and feedback. He also assisted with both the refinement of the idea and the pre-submission edition.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Acknowledgments

I would like to express my special thanks to my principal supervisor, Prof. Heng Tao Shen, for his generous support, and his valuable and in-depth guidance for my PhD study and research. In the past three years, I learned a lot from him. With his help, I learned how to discover fresh and intriguing research topics, how to peer-review publications, and how to write high-quality papers. The three-year research experience will also be of great benefit in my future career.

I would also like to thank my associative supervisor, Dr. Zi Huang. I appreciate all her contributions of time and ideas, to make my Ph.D. experience stimulating and rewarding.

I am very thankful to Dr. Yang Yang (University of Electrical Science and Technology of China), Dr. Yi Yang (University of Technology, Sydney), and Dr. Xiaoshuai Sun. They gave me practical guidance on research directions and help on technical details.

In addition, I thank the following people from the DKE group in the School of Information Technology and Electrical Engineering, for their insightful discussions with me: Prof. Xiaofang Zhou, Prof. Shazia Sadiq and Prof. Xue Li. I would like to thank my proof-reader, Julie Martyn for her attention to detail. I also have to thank the PhD students I worked with, Jiewei Cao, Peng Wang, Xuefei Li, Hongyun Cai, Chao Li and Douglas Alves Peixoto. I really appreciate having had the opportunity to work with them, and their friendships will always be remembered in my life.

Last but not least, I will express my thanks to my wife, Yunfei Shi, who always gave me the encouragement to help me get through times that were hard, sad and confused. During the last three years, I feel that we both learned a lot about life and strengthened our commitment and determination to each other and to live life to the fullest.

Keywords

machine learning, web mining, video event detection and recognition, video event retrieval, deep learning, binary embedding

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080109, Pattern Recognition and Data Mining, 45%

ANZSRC code: 080201, Analysis of Algorithms and Complexity, 45%

ANZSRC code: 080604, Database Management, 10%

Fields of Research (FoR) Classification

FoR code: 0806, Information Systems 20%

FoR code: 0801, Artificial Intelligence and Image Processing, 70%

FoR code: 0803, Computer Software, 10%

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we give a brief introduction of the research in this thesis, including the background, problem characterization, contributions, and the organization of the thesis.

1.1 Background

In the last decade, video content analysis has attracted increasing research interest in the fields of multimedia and computer vision. The significance of this research is growing due to many novel applications such as video indexing, retrieval, description, monitoring, surveillance, etc. Recently, significant research effort has been witnessed in developing new video content analysis models, which aim at automatically helping people find the videos that are related to pre-defined events.

The focus of this thesis is to address the issues of high-level or complex event analysis in large-scale videos. The definition of an "event" varies under different circumstances, and till now there has not been agreement or consensus on it. In some cases, some people define an event as a significant occurrence of a sequence of activities at specific locations and time [3], which involves a "4W" concept (What, Who, Where and When). This definition is often used for social media analysis, such as event sentiment analysis on Twitter [79] and topic sequence modelling for social events [111]. In the research fields of multimedia and computer vision, an event is often defined as a semantic pattern that involves complex objects, scenes, activities and their interactions [35]. To clarify, in this thesis we only focus on multimedia event analysis.

Usually an event has one or more of the following properties:

- It is a complex activity occurring at a specific place and time;
- It involves the interaction between people and/or other objects in some specific scenarios;
- It consists of a number of processes and activities that are loosely or tightly organized, which have either spatial or temporal semantic relationships;
- It is directly observable from the videos.

Based on the above properties, complex event analysis in large-scale videos is much more challenging than other tasks like action recognition [98, 94] or near-duplicate detection [50]. In general, the difficulties of video event analysis can be summarized as follows:

First of all, different events have different granularities, which make them difficult to understand, describe and recognize. For example, in some applications one may be interested in building models to detect some small granular events from the videos such as *repairing an appliance* and *parkour*, and sometimes we aim to find the large-granular event-related videos, such as *Wedding of Prince William and Kate Middleton*.

Second, different events have their unique characteristics. For example, some of them are sensitive to types of static evidence (e.g. human gestures or scenes), while others are sensitive to dynamic cues (e.g., motions or action sequences).

Third, the semantic information contained in the videos is hard to be directly represented by visual features, and semantically similar patterns often have vastly different visual representations.

Fourth, it is expensive to obtain sufficient video data to get robust models because manually labelling the videos for training data is quite labour-intensive.

Last but not least, when dealing with large-scale video datasets, it is extremely time and memory consuming to train reliable models.

1.2 Research topics

In this thesis we mainly focus on the following two tasks: video event detection & recognition, and video event retrieval.

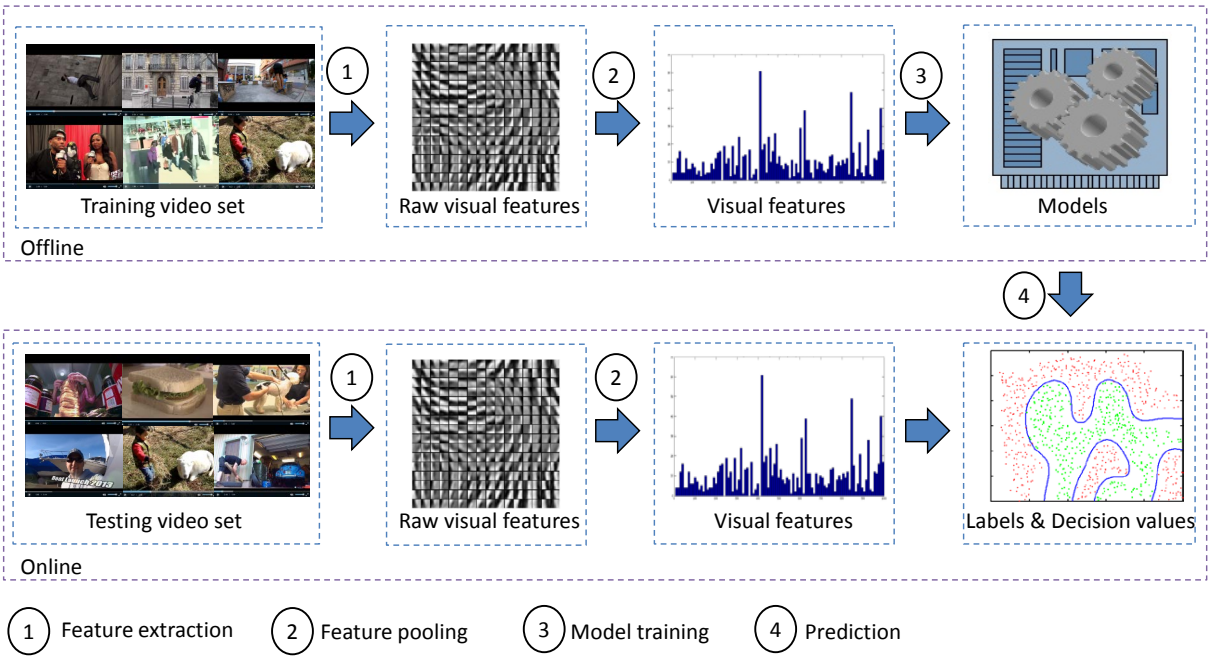


FIGURE 1.1: The general framework of video event detection & recognition

1.2.1 Video event detection & recognition

This task is mainly based on supervised or semi-supervised learning techniques, which are specially designed at a higher level of video content analysis. It can be formally defined as: given some video event kits that contain the instances of the pre-defined event, train a set of models that can automatically find the video recordings for these events based on pre-computed metadata. Similar to other video content analysis tasks such as action recognition and concept annotation, event detection & recognition aims to train a set of classifiers using a limited number of videos to predict the event labels of the test videos. The framework of this task is illustrated in Figure 1.1.

The top part of the framework is the offline processing. The first step is to extract the raw visual features from the training videos, and then the system applies a data quantization method to represent the videos as high-dimensional feature vectors for further processing. The key step of the offline processing is to apply a learning approach to train a set of models based on these feature vectors. In the online procedure, the testing videos are processed using the same approach as for the training videos, and the models trained in the offline procedure can be applied to predict the event labels or decision values in the testing video sets automatically.

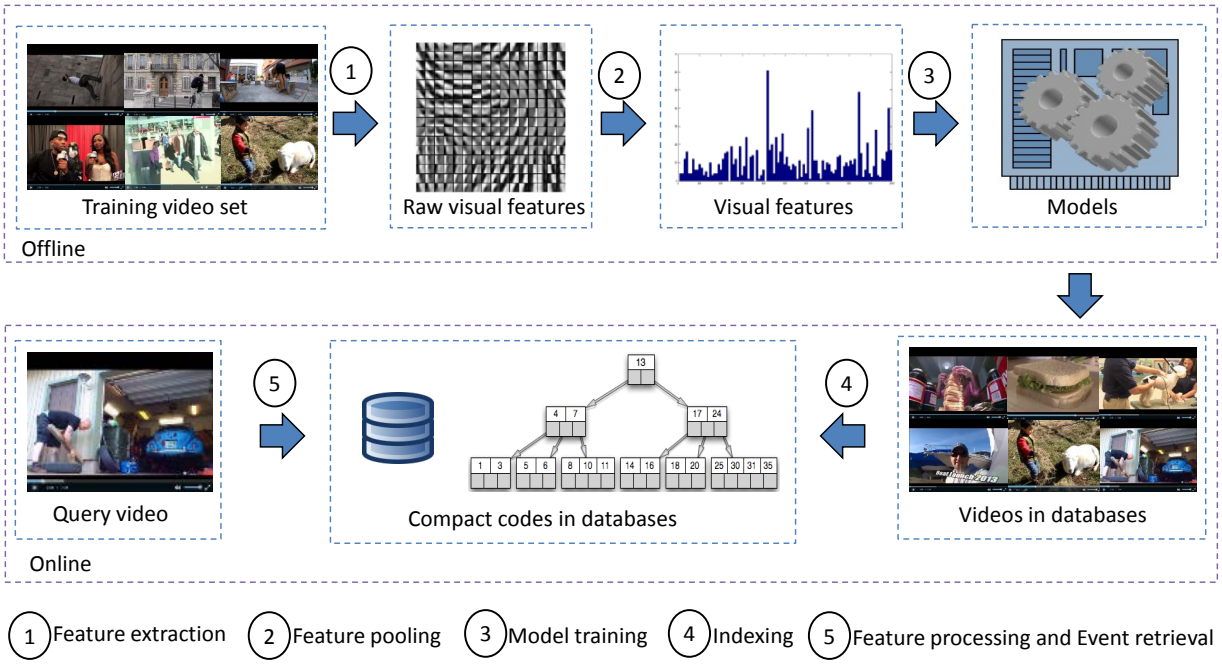


FIGURE 1.2: The general framework of video event retrieval

1.2.2 Video event retrieval

The definition of this task can be described as: given a query, search the semantically similar videos from large video databases at the event level. The query can be either a text keyword or a video clip. This task aims to design an indexing method to facilitate efficient event retrieval in large-scale video databases, and in this thesis we only focus on content-based retrieval, i.e., the query is a video clip. Compared with content-based video retrieval and near-duplicate retrieval, event retrieval in unconstrained video databases is much more challenging, because the indexing model is built at a high-semantic level rather than the visual or low-semantic level. The general framework of this task is described in Figure 1.2.

The framework of video event retrieval is similar to event detection & recognition as mentioned above, but there two major differences. First, the models trained in this task are used for indexing, rather than classification; second, given a query video, we do not know the exact event category it belongs to, and the search result should be semantically related to the given query video at the event level.

1.3 Problem formulations

In this section we describe the specific research problems in this thesis, as well as the motivations for research and the formulation of these problems.

1.3.1 Analysing the data distributions of the visual features

Automatic discovery of complex patterns has been proved to be helpful to assist video event analysis tasks [58, 60], so it is important to build effective models with a limited number of training instances. The challenges of video pattern recognition are multi-folds. First, the "semantic gap" between low-level content features and high-level semantic descriptors is hard to bridge. High-level pattern recognition involves a variety of unstructured concepts ranging from simple objects or actions to highly abstract events, and these patterns usually contain very rich contextual information with miscellaneous spatial and temporal cues. Second, the low-level features are usually noisy, which may seriously aggravate the recognition difficulty. Third, it is expensive to obtain sufficient video data to get robust recognizers because manually labelling the videos for training data is quite time-consuming and labour-intensive, especially when there are very few positive labelled instances available.

To overcome the above difficulties in complex video pattern recognition tasks, we aim to build a novel model to comprehensively analyse the data distribution by transforming the original data into a lower dimensional feature space. In the obtained lower dimensional feature space, both labelled and unlabelled videos are then used together to optimize the target prediction function.

1.3.2 Building a robust spatial-temporal deep model for event detection

Recently, the learned visual features from a pre-trained convolutional neural network (CNN) have been shown to have a far superior performance than hand-crafted features. Specifically in the task of video event detection, Xu et al. proposed an effective video representation method based on the pre-trained CNN model [100]. Applying the VLAD quantization method [33] on the pooling layer and fully-connected layer, the detection performance is outstanding on the TRECVID MED dataset. However, this video representation is unsequenced, so it loses all the temporal information.

Although the occurrences of static patterns are considered as evidence to discriminate for whether a video contains a pre-defined event (e.g., *bee keeping*), some events are still temporally sensitive (e.g., *felling a tree*).

Based on the above observation, we attempted to build a robust spatial-temporal deep model for large-scale video event detection. In our setting, each video follows a multiple instance assumption, where its visual segments contain both the spatial and temporal properties of events.

1.3.3 Incorporating the web information into event recognition models

In real world applications, we are often more interested in "large" granular events (e.g., *Beijing Olympic opening ceremony* and *Wedding of Prince William and Kate Middleton*), which may be viewed from different spatial and temporal perspectives, and contain very rich semantics. For this kind of event analysis, there exists a huge semantic gap between visual cues and event descriptors. These descriptors include audio signals and texts, and both of them can help us better understand the events contained in visual contents. In this case, one promising strategy is to explore the semantic information of videos for assistance of event analysis tasks. For this purpose one may extract some semantics from the contextual information as prior knowledge about these events. On video websites like YouTube, some auxiliary cues, such as acoustic signals and texts (titles, tags and descriptions) that are associated with the corresponding videos, and can serve as precious resources for event analysis. However, how to utilize these auxiliary texts for automatic video event recognition is a big challenge. The reasons are two-fold:

- The existing texts associated with videos often contain incomplete and imprecise information, which is probably not relevant to the videos themselves. In order to obtain comprehensive and precise descriptions of events from the semantic view, an intuitive way is to utilize a universal knowledge base (e.g., WordNet or FrameNet) to enrich the semantics. However, such a kind of knowledge base mainly describes the term correlations but falls short of explanations of the specific events, especially for emerging and evolving events happening around us.

- Text information and visual features are highly heterogeneous modalities. Although both are imperative for event analysis and are complementary to each other, they are from totally

FIGURE 1.3: An illustration of two completely heterogeneous features for event *Beijing Olympic opening ceremony 2008*.

different perspectives and may characterize different aspects of the events. Taking the event *Beijing Olympic opening ceremony* for example (see Figure. 1.3), the visual information of a relevant video is mainly about the fireworks or other significant visual phenomena, while the textual cues are expressed as *The world watches in wonder as this glittering event raises the curtain on the Beijing 2008 Summer Olympic Games*, which is an overall description about this event. Obviously, in most cases the visual information and textual cues are taken from different aspects of the events, i.e., neither of them could be well described by the other. Therefore, it is not suitable to directly make the existing texts of videos as the intermediate semantic representations. In this case the two-view model seems to be more appropriate for video event recognition tasks. Since these events can be viewed from both videos and texts, two independent models, one for each of them can be trained together. In the testing phase, since videos under the unconstrained circumstances do not contain any text information, knowledge adaptation from texts to videos is necessary. However, till now there have not been any effective approaches to simultaneously explore and make use of the visual and textual correlations in video event analysis.

In order to solve the problem of incomplete and imprecise semantic description, we tried to explore the ubiquitous web documents to build an event knowledge base for video semantic analysis. Using existing search engines, it is quite easy to gain access to the event-relevant documents, from which the event semantic structures generated can be used to describe events with abundant semantics under specific event contexts. With the help of this knowledge base, the partial texts accompanying the videos can be greatly enriched. Based on the enriched textual cues, we further tried to propose a model that can correlate the semantics to visual features and obtain an effective model for video event recognition.

1.3.4 Analysing the event-relevant visual states for large-scale video event retrieval

In multimedia content analysis tasks, a video can be represented either as a flat vector by feature aggregation, or as a sequence of feature vectors. However, a flat video vector may lose the internal structure of the video itself. On the other hand, representing a video as a sequence of feature vectors is deemed to increase the computational complexity significantly, especially when we need an efficient search within large-scale video datasets.

In many real-time applications, the technique of binary embedding, which is often called hashing, has been extensively adopted to encode high-dimensional feature vectors into compact binary codes, resulting in fast computation by XOR operators in the Hamming space to approximate the distance between feature vectors, thus achieving scalable information retrieval. Till now a variety of hashing models have been proposed and widely applied to the near-duplicate content search [93, 90, 53] and visual tracking [48, 57]. However, there are several issues when applying binary embedding approaches for video event analysis. On one hand, most hashing approaches are mainly designed at the visual level rather than the semantic level, with the result that a "semantic gap" may exist between the visual representation and event description. On the other hand, the transformation from the real number space into the binary space may cause severe information loss, in particular the loss of the spatial and temporal information describing complex patterns in videos. In addition, in order to train a reliable hashing model, a large number of training videos are needed, which is both time and memory consuming.

To overcome the above difficulties, we aimed to design a video binary embedding framework to facilitate the scalable video event retrieval, which can effectively reduce the computational complexity of the hash model training while ensuring the retrieval accuracy.

1.4 Main contributions

We have designed effective solutions for the research problems above, and evaluated these models by conducting comprehensive experiments on several real-world video datasets. We explain our contributions in the following subsections.

1.4.1 Max-margin adaptive model for complex video pattern recognition

As introduced in subsection 1.3.1, most existing semi-supervised learning methods have failed to analyse video data distributions or effectively bridge the "semantic gap" between low-level visual features and high-level descriptors.

In this work we propose a generic model by learning a mapping matrix to transform the original feature space into a lower dimensional subspace, which can effectively reduce the data distribution disparity between labelled and unlabelled videos and also maximize the margin between the videos from different pattern categories. The iterative optimization solution to the objective function of MMA can use very limited labelled source data and a large number of unlabelled auxiliary data for complex video pattern recognition. The experiments were conducted on two public datasets including CCV [1] for video object/event detection and HMDB [2] for action recognition. Our results demonstrated that the proposed MMA model is very effective on complex video pattern recognition tasks, and outperforms the state-of-the-art algorithms.

1.4.2 Robust spatial-temporal deep model for multimedia event detection

To address the issued described in 1.3.2 we propose a robust video event detection model, which consists of two major steps: unsupervised recurrent learning and supervised fine-tuning. The

[1] http://www.ee.columbia.edu/dvmm/CCV/

[2] http://serre-lab.clps.brown.edu/resources/HMDB/

contributions of this work are as follows:

- We propose to utilize the recurrent neural network to reconstruct the video representation in an unsupervised approach, which does not need the label information of the videos, and such a model can incorporate both the spatial and temporal information of the videos;
- In the supervised training phase, the parameters are optimized from better locations in high-dimensional space, thus boosting detection performance;
- We have conducted extensive experiments on the challenging TRECVID MED 2014 dataset, and have demonstrated that our proposed system can achieve very promising detection performance in MED tasks.

1.4.3 Web video event recognition by semantic analysis from ubiquitous documents

In order to solve the problem of incomplete and imprecise semantic description as discussed in subsection 1.3.3, we explored ubiquitous web documents to build an event knowledge base for video semantic analysis. The contributions of this work are summarized as follows:

- We have constructed an event knowledge base by deeply mining ubiquitous web documents. The knowledge base contains a set of event semantic graphs (ESGs), each of which corresponds to an event. An ESG represents an event with affluent textual words (as nodes) as well as their mutual correlations (as edges) in different hierarchical layers, which can be utilized to enhance the textual cues of videos;
- We propose a *Two-View Adaptive Regression* (TVAR) learning model, which can jointly explore the intrinsic correlation between enriched textual and visual cues for video event recognition. An efficient iterative algorithm is proposed for the optimization of the model;
- We have conducted extensive empirical studies on two real-world web video datasets, one of which is a new dataset created by ourselves for public use. The results show promising improvements over state-of-the-art approaches.

1.4.4 Scalable video event retrieval by visual state binary embedding

In order to solve the problems described in subsection 1.3.4, we propose an efficient hash model of *Visual State Binary Embedding* (VSBE). In this model, we define a novel metric to evaluate the representativeness of each key frame in a given video by considering its three importance measures at the video-level, event-level and global-level respectively. The importance of a frame at the video-level is measured by its representativeness to the video it belongs to. The event-level importance is designed to measure the relevance of a frame to its belonging event. The global importance is simply the frequency of a specific visual pattern in all videos. By considering the three importance measures, a number of highly representative key frames are then selected to sketch the semantic cues. The visual information contained in these selected key frames can reveal the most important semantics of the videos, and they are also descriptive to the corresponding events. Such kind of visual information from the training video corpus will be used to form the semantic constraints for learning the binary embedding functions. In the retrieval phase, each query video is first represented as a binary matrix, where each row stands for a key frame. Thus, the semantic information of the video is preserved in the retrieval phase. Finally, the distance between the two videos is efficiently calculated by XOR operation in the Hamming space.

The key contributions of this work are summarized as follows:

- We study key frame representativeness through a novel quantified measurement, and define a new concept of visual state to capture the semantics of videos;
- We propose a new binary coding model VSBE, which only uses a limited number of selected frames for the training of binary embedding functions, resulting in a significantly boosted training efficiency when dealing with large-scale video datasets;
- The proposed VSBE model can integrate the constraints given by the importance of the visual states, and an efficient iterative algorithm is designed for its optimization;
- Comprehensive experiments on the challenging TRECVID MED video dataset have been conducted, which confirm the performance superiority of the proposed model compared with state-of-the-art video event retrieval approaches.

1.5 Thesis organization

The rest of this thesis is organized as follows: In Chapter 2, we review the field of video event analysis and summarize the relevant techniques such as visual feature extraction and hashing. In Chapter 3 we present a content based semi-supervised model MMA for general video pattern recognition, which can be applied in action recognition and constrained video event recognition. In Chapter 4 we analyse the spatial-temporal structure of the videos, and propose a deep neural network with a two-step optimization procedure: unsupervised recurrent auto-encoding and supervised fine-tuning. In Chapter 5 we introduce a context based web event recognition framework that can deeply explore the semantic information from ubiquitous web documents to enhance the performance of video event recognition. In Chapter 6, we focus on integrating the semantic information into the hashing model for scalable video event retrieval tasks. Finally the conclusion and the possible future research directions suggested by the thesis are given in Chapter 7.

Chapter 2

Literature Review

In this chapter we give a comprehensive review of video event analysis, including the most recent related work of machine learning, computer vision, and other techniques.

2.1 Overview

We organize the research of video event analysis into three different key areas: visual feature representations, event detection & recognition models, and event retrieval models. The design of real systems mainly depends on the contexts of applications, and these three key parts are displayed in Figure 2.1.

The review of visual feature representations will be discussed first in Section 2.2, because this is the most basic technique applied in various video content analysis tasks. Then in Section 2.3 we review the recent models for event detection & recognition. Finally in Section 2.4 we give a brief summary of event retrieval models and relevant techniques.

2.2 Visual feature representations

In multimedia content analysis and computer vision, visual feature representation plays a critical role in different tasks. Specifically, a well-designed feature should be robust against variations and can well describe the visual properties of the videos. In this section we will discuss two kinds of visual features that are commonly used: hand-crafted features and learned features.

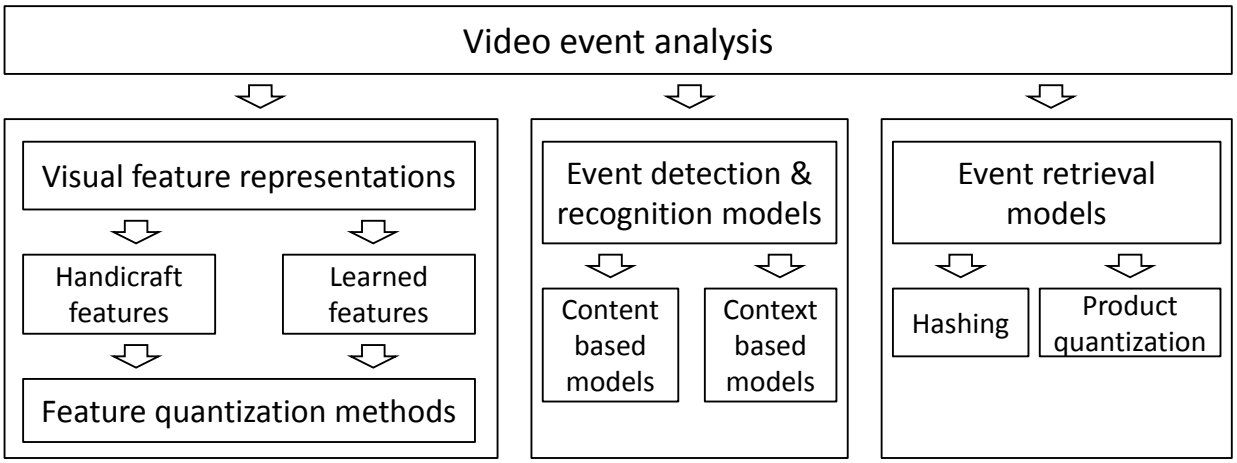


FIGURE 2.1: Three key parts of video event analysis

2.2.1 Hand-crafted features

Usually a video is represented as a sequence of frames, which contains both static and dynamic information. The static information is just reflected by the frame-based features, but the dynamic information is much more complex.

A video frame can be represented using a global feature or a set of discriminative local features. The global features are mainly based on the overall distribution of colour, texture or edge information, such as GIST and colour histogram. However, the global features fail to capture the local properties of the video frames. The idea of the local feature is to identify the local patches, then describe them in a meaningful manner. One of the most well-known local features is scale-invariant transform (SIFT) [54], which partitions a patch into equal-sized grids, each of which is described by a histogram of gradient orientations. There are several extensions of the SIFT feature, such as colour SIFT [2] and PCA-SIFT [42], which are widely used in describing the static properties of the video frames. Extracting visual features from all frames of the videos is extremely time and memory consuming, so a practical way for feature extraction is to sample the frames. In some cases the video has rapid visual changes, and the frames should be carefully sampled and selected, because a low-sampling rate may lead to the loss of important information, while a high-sampling rate usually results in redundancies.

Although single-frame based visual features are the most straightforward ways to describe

video content, they fail to depict the inter-frame changes, i.e., the dynamic properties of the videos. Unlike single-frame based features, spatial-temporal features take the time dimension of the videos into account. For example, Laptev extended the Harris corner patch detector [47] to locate spatial-temporal interest points (STIP), which are represented by the concatenation of HOG and HOF descriptors. Similarly, MoSIFT adds optical flow to the SIFT descriptor, which can be applied in video motion analysis [11]. In order to enhance the descriptive ability in videos from both spatial and temporal perspectives, densely sampling the interesting points and adding trajectory information are helpful. Wang et al. proposed the Dense Trajectories to sketch the spatial-temporal properties of the videos for action recognition. In this method, the interest points are densely selected from the frames and represented as the concatenations of trajectories, HOG/HOF, and MBH features [87]. They further boosted performance by estimating the camera motion and employing the human bounding boxes, and named the feature Improved Trajectories (IDT) [89]. Till now, IDT is still one of the best visual features for video motion analysis. However, the calculation of IDT is extremely time-consuming, needing unaffordable computational resources in real-world scenarios. In addition, Wang et al. proposed a motion feature named Expanded Relative Motion Histogram of BoW (ERMH-BoW), to encode both motion and the interactions between different objects/scenes [86].

2.2.2 Learned features

Recently, based on deep learning techniques, the learned visual features from the pre-trained convolutional neural network (CNN) show a markedly superior performance to hand-crafted features in image processing. The CNN model is an extension of traditional back-forward propagation neural network, which usually contains several convolutional layers, max-pooling layers and fully-connected layers. Popular CNN models such as AlexNet [44] and VGG net [75] were trained on a large image corpus ImageNet, with the result that the error rate of image classification has been continually decreasing over the last few years. The data from inter-mediate layers has a powerful descriptive ability thus it can be used as the static visual feature for a variety of multimedia content analysis tasks.

Variations of CNN models have also been designed for 3D video data. For example, Dutran

et al. proposed a 3D CNN framework for action recognition [80], and Karpathy et al. designed a CNN model for video classification [41]. Although the deep-learned CNN feature has achieved exceptional performance on 2D image data, its variations on 3D video data are less satisfactory. Till now there is still no deep model available for universal video analysis tasks.

2.2.3 Visual feature quantization methods

After extracting the raw visual features, the next step for further video processing is to pool these raw features into feature vectors. The simplest way is to use the bag-of-visual-words model (BoWs). By applying this method, a feature vector can be represented as a sparse vector of the occurrence of words, i.e., a sparse histogram over the pre-computed vocabulary. The vocabulary with K centres is generated by conducting the clustering algorithms on an interest point set sampled from sufficiently large videos. One of the disadvantages of the BoWs model is that it ignores the relationships among spatial-temporal patches in the high-dimensional space, which are significant in video representation. In order to solve this problem, *Fisher Vector* (FV) [65] and *Vector of Locally Aggregated Descriptors* (VLAD) [33] have been proposed, both of which have great advantages over the BoWs. The FV applies the Gaussian Mixture Model (GMM) to generate K components from the sampled visual descriptors, and the mean and covariance deviation vectors of each component are concatenated as a sparse and long vector. A variation of FV, the improved Fisher Vector (IFV), applies the non-linear additive kernel and normalization to enhance discriminative ability [66]. VLAD is a simplified version of FV. In this method, the K components are obtained in the same way as the BoWs, which omits the covariance. Recently, some extensions of VLAD such as the Riemannian version of VLAD [19], and VLAD-k [40] have been proposed, which can further enhance the traditional VLAD method.

Specifically in the task of video event detection, Xu et al. proposed an effective video representation method based on the pre-trained CNN model [100]. Applying the VLAD quantization method on the pooling layer and fully-connected layer, the detection performance is even better than IDT. However, this video representation is unsequenced, thus losing all the temporal information.

2.3 Event detection & recognition models

When given the visual feature representations, event recognition & detection can be achieved by a variety of supervised or semi-supervised methods. This is a typical machine learning technique, which contains both training and testing procedures. Here we discuss some content based models and context based models, which have appeared in the last few years.

2.3.1 Content based models

The content based models are trained on the event-related videos without any auxiliary information. One of the popular models in a range of computer vision and multimedia content analysis applications is kernel-based classification. Among different kernel-based classifiers, support vector machines (SVMs) with Gaussian or χ^2 kernels are widely used in a variety of tasks such as action recognition. Specifically in video event detection & recognition, SVM is often used as a baseline method, although some other models have been proposed in the recent years. For example, Ma et al. proposed an SAIR method that can automatically learn an intermediate representation from low-level features together with a classifier for MED [59]. Gan et al. proposed a flexible CNN specifically for video event detection and evidence recounting [21].

In recent years, multi-view learning (MVL) has shown its significance in solving many computer vision problems. One assumption of MVL is that the target data can be observed from several conditionally independent sources, with these different views being complementary to each other. Thus the integration of these views can generate robust predictive models. MVL has been extensively applied in supervised learning [110, 108, 68, 20], unsupervised learning [27, 88], semi-supervised learning [56, 103, 94, 55, 72] and transfer learning [109, 98]. Specifically in video event detection, Xu et al. proposed a weighting scheme to evaluate the importance of different visual features [101]. Natarajan et al. combined several visual features and a two-step strategy employing multi-model and late fusion methods for MED [63]. Nikolaos proposed a two-phase approach using non-linear discriminant analysis to identify the event-related subspace, then used a linear SVM to learn the final model in the derived subspace [24].

Besides the above introduced methods, there has been a plethora of literature advocating the use of temporal analysis of video event detection. For example, in Lai et al.'s work, temporal

information has been proved to achieve a better detection performance compared with the flat vector representation [46]. Similarly, Kevin et al. designed a method to learn the temporal structure based on the hidden Markov model for event detection [78]. Vignesh et al. also tried to learn the temporal embeddings [70], and Remi et al. built a temporal partitioning model for video analysis [81]. Yu et al. analysed temporal information for the surveillance video event detection task [12].

2.3.2 Context based models

In many video content analysis tasks, bridging the low-level visual features and high-level descriptors by prior knowledge is necessary because videos often carry very rich semantics. The contextual information can be obtained from multiple sources such as auxiliary videos, images and texts, and such information can be considered as the prior knowledge to enhance the performance of the models. For example, Habibian et al. proposed a multimedia embedding method that integrates the information crawled from web pages to translate the visual concepts into texts [28]. Merler et al. used 180 pre-trained textual words to formalize the videos as semantic model vectors for event recognition [62]. Ma et al. designed a method that can utilize visual attributes from multiple video sources for the task [60]. Chen et al. leveraged loosely labelled web videos and web images for video event recognition [10]. Similarly, Krishinamoorthy et al. utilized the knowledge mined from web documents for video description [43], and Wan et al. proposed a graph-based approach that can enrich the texts from web for video retrieval [85]. Ramanathan et al. proposed a method based on natural language descriptors for action recognition [69]. The influence of online videos was also studied to understand how social events were propagated in social websites [51]. Context based models are often considered as a kind of multi-task learning. For example, Chang et al. proposed a unified model for multimedia event recounting and detection [9].

2.4 Video event retrieval

Different from the video event detection & recognition task that tries to learn a set of classifiers, the video event retrieval aims to design an indexing model to store the video data in a compact way for fast search. In the retrieval phase, given a query video, we do not know the exact event category it

belongs to, and the search result should be semantically related to the query video at the event level. Since the exhaustive search in the real number space is extremely time-consuming, it is necessary to apply some indexing techniques to facilitate the computation when there are large numbers of videos in the database. Generally speaking, there are two classes of indexing techniques: hashing and product quantization.

2.4.1 Hashing

To accelerate information retrieval in large multimedia databases, one of the most promising approaches is to embed the high-dimensional data into binary codes, which is often called hashing [91, 92]. Recently, many learning based hashing algorithms have been proposed[95, 97, 105, 90, 53, 25, 73, 74], and these are mainly used in the nearest neighbour search of image data. In addition, cross-media hashing methods have been proposed for heterogeneous media retrieval [76, 31, 112]. In many multimedia retrieval applications, binary embedding approaches are mainly designed at the visual level rather than the semantic level, so it is difficult to embed the complex structures into binary codes for videos. Until now there is still only a very limited number of hashing methods specifically designed for unconstrained video event retrieval. One representative hashing model is the inductive hashing via structural learning proposed by Ye et al., which is a supervised hashing model [104]. This model attempts to learn the hash functions by discriminating event classes at the video level. As the first attempt to encode the frames into binary codes at the video level, it has three limitations: first, it requires all the key frames for hash model training, so the time complexity is extremely high; second, the proposed optimization approach needs several tens of iterations to converge, so the learning speed is very slow; third, it adopts hinge loss in the objective function, which can be considered as a “hard” margin, and the objective function value can barely satisfy this condition.

2.4.2 Product quantization

Product Quantization (PQ) [32] is another promising indexing approach for multimedia content retrieval. The idea is to decompose the space into a Cartesian product of low-dimensional subspaces and to quantize each subspace separately, so a vector can be represented by a short code

composed of its subspace quantization indices. The Euclidean distance between two vectors can be efficiently estimated from their codes based on a pre-trained code book. The optimal space decomposition is important for the PQ performance, so the *Optimized Product Quantization* (OPQ) [22] and *Locally Optimized Product Quantization* (LOPQ) [39] have been designed by rotating the original data to the optimal location in order to improve the approximation of data distribution.

In some video event analysis tasks, PQ is also used to encode the high-dimensional feature vectors to speed up computation. For example, in [71], Revaud et al. proposed a circulant temporal encoding method for video event retrieval, and PQ is applied to encode the video frames into compact codes. In [100], the VLAD pooling on the CNN features makes for very high dimensions of the feature vectors, so PQ is used to reduce the memory and accelerate the computation, while the accuracy remains almost the same.

2.5 Summary

In this chapter we illustrated some key research topics in video event analysis, including visual feature representation and learning models. Motivated by the literatures in this chapter, we are interested in applying some machine learning techniques to build effective and novel models for video event detection, recognition and retrieval. For example, the MMA model in Chapter 3 adopts the concept of transfer learning, and the TVAR model in Chapter 5 is mainly inspired by multi-view learning and intermediate visual representation. In the next four chapters, we will introduce our proposed models for effective video event analysis in detail.

Chapter 3

Max-margin adaptive model for complex video pattern analysis

3.1 Overview

Using unlabelled data to assist the model training in a semi-supervised approach is a common approach in many applications. The assumption of semi-supervised learning is that the samples with similar data distributions are more likely to share the same labels. However, such assumption does not always hold. Another challenge is there is a "semantic gap" between low-level features and high-level descriptors. To overcome the above difficulties in complex video pattern recognition tasks, we specifically consider the data distribution of the visual features in our proposed MMA model. This adaptive model aims to obtain a mapping matrix which can be used to transform the original data into a lower dimensional feature space, with the objective of not only minimizing the data distribution disparity between labelled and unlabelled data but also maximizing the margins between positive labelled videos and negative labelled ones for a given pattern. In the obtained lower dimensional feature space, both labelled and unlabelled videos are then used together to optimize the target predict function.

3.2 Max-margin adaptive model

3.2.1 Problem formulation

Generally speaking, video pattern recognition can be considered as a supervised or a semi-supervised learning task. Suppose the labelled source video collection is represented as a matrix $X^s = \{\mathbf{x}_1^s, \mathbf{x}_2^s, \ldots, \mathbf{x}_{n_s}^s\} \in \mathbb{R}^{m \times n_s}$, where each column is a video sample, and n_s and m are the numbers of training samples and the dimension of the feature vectors respectively. Given a label, its existence on all the training videos is represented as a vector $\mathbf{y}^s = [y_1^s, y_2^s, \ldots, y_{n_s}^s] \in \{-1, 1\}^{n_s \times 1}$. Similarly the auxiliary video collection is $X^a = \{\mathbf{x}_1^a, \mathbf{x}_2^a, \ldots, \mathbf{x}_{n_a}^a\} \in \mathbb{R}^{m \times n_a}$, and n_a is the number of auxiliary videos. The target is to utilize both labelled training videos and unlabelled auxiliary ones to get a robust recognizer for a specified pattern recognition task in unseen video sets.

Different from binary or multi-class classification problems, video pattern recognition aims to find the *has-a* correlations between the specified patterns and videos instead of *is-a* ones. Since there are many hidden patterns in the videos, most of them may be redundant and even have negative effects. Here we want to build a robust recognizer that can both effectively utilize the unlabelled auxiliary videos and select the most discriminative feature components to identify the predefined patterns.

Given that the discriminative information in labelled videos is quite limited, the MMA model can utilize a large number of unlabelled videos to assist the pattern recognition tasks. In real-world applications, the unlabelled auxiliary videos can be acquired by searching the keywords of users' interests from public video sources such as YouTube. However, the following reasons make it difficult to fully take advantage of auxiliary videos and get a robust recognizer in semi-supervised learning approaches: 1) the quality of auxiliary videos varies, i.e., the videos from public sources are often quite noisy; and 2) the number of positive labelled videos is much smaller than the number of negative labelled videos, leading to highly imbalanced numbers, and 3) the data distribution of the videos from different sources may be varied.

The MMA model jointly considers two optimization objectives: 1) minimizing the loss function and corresponding regularization; and 2) finding a discriminative subspace by learning a linear

mapping matrix that can simultaneously minimize the data distribution difference between training data and auxiliary data and maximize the margin between positive labelled videos and negative labelled ones for a pattern.

Suppose C and $\xi = [\xi_1, \xi_2, \ldots, \xi_{n_s}]$ are the penalty cost and hinge loss for the labelled training data. We can construct a standard linear SVM model as follows:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^{n_s} \xi_i \\ \text{s.t.} \quad & y_i^s(\mathbf{w}^\top \mathbf{x}_i^s + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, \forall\, i = 1, 2, \ldots, n_s. \end{aligned}$$

By solving this problem, the decision function is:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b \tag{3.1}$$

where $\mathbf{w}$ is a weight vector and b is a bias.

However, this supervised learning algorithm can neither utilize unlabelled auxiliary data, nor consider the most discriminative feature components from the perspective of data distributions. As we know, usually only a subset of the features has positive effect for the recognition of a particular pattern while other features are redundant or even negative. So in our framework, instead of obtaining a global classifier on the original feature vector $\mathbf{x}$, we use a mapping matrix $\Theta \in \mathbb{R}^{r \times m}$ ($r < m$) to transform the original vector space to a new subspace $\mathbf{x}'$, i.e.,

$$\mathbf{x}' = \Theta \mathbf{x} \tag{3.2}$$

Thus, the linear decision function becomes:

$$f(\mathbf{x}) = \mathbf{u}^\top \Theta \mathbf{x} + b \tag{3.3}$$

where $\mathbf{u}$ is the corresponding weight vector in the subspace.

In order to take advantage of unlabelled data, we want to minimize the data distribution disparity to improve the performance. The MMD (maximum mean discrepancy) [7] is defined as:

$$dist(X^s, X^a) = ||\frac{1}{n_s}\sum_{i=1}^{n_s}\phi(\mathbf{x}_i^s) - \frac{1}{n_a}\sum_{i=1}^{n_a}\phi(\mathbf{x}_i^a)||_{\mathcal{H}} \tag{3.4}$$

where ϕ is a function that can transform the original feature space to a higher dimensional Reproduced Kernel Hilbert space.

Considering the mapping matrix Θ and the finite data samples, let $X = [X^s, X^a] \in \mathbb{R}^{m\times(n_s+n_a)}$, $\mathbf{d}$ be a column vector with the first n_s entries having the value of $1/n_s$ and last n_a entries having the value of $-1/n_a$, and let $D = \mathbf{d}\mathbf{d}^\top$. The square empirical MMD with a linear mapping can be written as:

$$J(\Theta) = dist^2(X^s, X^a) = tr(\Theta X D X^\top \Theta^\top) \tag{3.5}$$

The minimization of $J(\Theta)$ can facilitate the model training when there exists data distribution disparity between labelled data samples and unlabelled auxiliary ones, and the rigorous theorem is proved in [6] for domain adaptation. Suppose $\mathcal{H}$ is a hypothesis space of VC-dimension r and $n = n_s = n_a$, for each $h \in \mathcal{H}$ at least with the probability $1-\delta$, the empirical risks of a hypothesis on labelled training data ϵ^s and unlabelled auxiliary data ϵ^a have the following relationship:

$$\epsilon^a \leq \epsilon^s + dist(X^s, X^a) + 4\sqrt{\frac{2r\log(2n) + \log(\frac{4}{\delta})}{n}} + \lambda \tag{3.6}$$

where λ is a small number. The above relationship indicates that a smaller MMD leads to a smaller distribution difference between labelled training data and auxiliary data.

Since minimizing the square MMD between training data and auxiliary data does not contain any label information, it is still unable to get the most discriminative features and to reduce the redundant ones due to the fact that only a small subset of features contributes to the specified patterns. Under this circumstance, these useful features are usually disturbed by irrelevant ones, and we call this phenomenon as "feature submergence". To solve this problem, we also add another item $Q(\Theta)$ to maximize the margin between positive data and negative data identified with the label information. Assume the labelled positive video collection is $X^{s+} = [\mathbf{x}_1^{s+}, \mathbf{x}_2^{s+}, \ldots, \mathbf{x}_{n_s^+}^{s+}] \in \mathbb{R}^{m\times n_s^+}$ and the labelled negative one is $X^{s-} = [\mathbf{x}_1^{s-}, \mathbf{x}_2^{s-}, \ldots, \mathbf{x}_{n_s^-}^{s-}] \in \mathbb{R}^{m\times n_s^-}$, where n_s^+ and n_s^- are the numbers of positive samples and negative samples respectively, and $X^s = [X^{s+}, X^{s-}]$. In order to enhance the effect of useful features and shrink the weight values of the redundant

features to zero, we aim to maximize the MMD between X^{s+} and X^{s-}. Similarly, let $\mathbf{h}$ be a column vector with the first n_s^+ entries are $1/n_s^+$ and last n_s^- entries are $-1/n_s^-$, $H = \mathbf{h}\mathbf{h}^\top$. This linear transformation is represented as:

$$Q(\Theta) = dist^2(X^{s+}, X^{s-}) = tr(\Theta X^s H X^{s\top} \Theta^\top) \tag{3.7}$$

It can be seen that $Q(\Theta)$ can also be considered as a feature selection function. Intuitively, if two groups of videos are quite similar with each other but belong to different pattern classes, maximizing $Q(\Theta)$ means that the most discriminative features can be identified in the new feature space. By performing $Q(\Theta)$, positive samples can be better separated from negative samples in the new feature space.

At the same time, by considering the unlabelled auxiliary data, i.e., $J(\Theta)$, the final MMA model can be formulated as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u},b,\Theta} \quad & \tfrac{1}{2}\mathbf{u}^\top\mathbf{u} + C\sum_{i=1}^{n_s}\xi_i + \tfrac{1}{2}\lambda_1 J(\Theta) - \tfrac{1}{2}\lambda_2 Q(\Theta) + \tfrac{1}{2}\eta||\Theta - I||_2^2 \\ \text{s.t.} \quad & y_i^s(\mathbf{u}^\top\Theta\mathbf{x}_i^s + b) \le 1 - \xi_i^s \\ & \xi_i^s \ge 0, \forall\, i = 1, 2, \ldots, n_s \end{aligned}$$

This objective function jointly considers the following goals: (i) minimization of the empirical risk and hinge loss; (ii) minimization of the distribution disparity between the training data and the auxiliary data; (iii) maximization of the margin between positive labelled data and negative labelled data; and (iv) keeping Θ orthoganal. It is conducted for the solution of the weight parameter $\mathbf{u}$, the bias parameter b, and the mapping matrix Θ which achieves both goals (ii) and (iii).

3.2.2 Solution of the adaptive model

Since the optimization problem is neither convex nor concave, we use a heuristic approach described as follows.

By adding the Lagrange multiplier $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_{n_s}]$ and $\beta = [\beta_1, \beta_2, \ldots, \beta_{n_s}]$ on the constraints of the objective function, we can get:

$$L = \frac{1}{2}\mathbf{u}^\top\mathbf{u} + C\sum_{i=1}^{n_s}\xi_i^s + \frac{1}{2}\lambda_1 J(\Theta) - \frac{1}{2}\lambda_2 Q(\Theta) + \frac{1}{2}\eta||\Theta - I||_2^2 - \sum_{i=1}^{n_s}\alpha_i[y_i^s(\mathbf{u}^\top\Theta\mathbf{x}_i^s + b) - 1 + \xi_i^s] - \sum_{i=1}^{n_s}\beta_i\xi_i. \tag{3.8}$$

We can then calculate the partial derivative of Θ,

$$\frac{\partial L}{\partial \Theta} = \lambda_1\Theta X H X^\top - \lambda_2\Theta X^s H X^{s\top} + \eta\Theta - \sum_{i=1}^{n_s}\alpha_i^s y_i^s \mathbf{u}\mathbf{x}_i^{s\top}. \tag{3.9}$$

Let $\frac{\partial L}{\partial \Theta} = 0$, we can arrive at:

$$\Theta = (\sum_{i=1}^{n_s}\alpha_i^s y_i^s \mathbf{u}\mathbf{x}_i^{s\top} + I)(\lambda_1 X H X^\top - \lambda_2 X^s H X^{s\top} + I)^{-1}. \tag{3.10}$$

The detailed solution can be achieved by alternatively executing steps 2 and 3, as described below until convergence:

step 1: Initialize and fix the parameter $\Theta = I$;

step 2: Solve the supervised learning problem, and get the optimal $\mathbf{u}$, which can be solved by SVM as follows:

$$\begin{aligned} \min_{\mathbf{u},b} \quad & \frac{1}{2}\mathbf{u}\mathbf{u}^\top + C\sum_{i=1}^{n_s}\xi_i \\ \text{s.t.} \quad & y_i^s(\mathbf{u}^\top\Theta\mathbf{x}_i^s + b) \le 1 - \xi_i \\ & \xi_i \ge 0, \forall\, i = 1, 2, \ldots, n_s \end{aligned}$$

with the optimal solution:

$$\mathbf{u} = \sum_{i=1}^{n_s}\alpha_i^* y_i \mathbf{x}_i^s \tag{3.11}$$

where α_i^* is the optimal solution of the dual problem.

step 3: Update Θ by equation (3.10).

When we get the optimal mapping matrix Θ, the data distribution becomes quite suitable for semi-supervised learning that can enhance the generalization ability of the classifier. Thus, the final prediction function can be achieved by solving the following semi-supervised problem on $\mathbf{u}$ and b in the mapped feature space:

$$\begin{aligned}
\min_{\mathbf{u},b,\{y_i^a\}} \quad & \tfrac{1}{2}\mathbf{u}^\top\mathbf{u} + C(\sum_{i=1}^{n_s}\xi_i^s + \sum_{i=1}^{n_a}\xi_i^a) \\
\text{s.t.} \quad & y_i^s(\mathbf{u}^\top\Theta^*\mathbf{x}_i^s + b) \le 1 - \xi_i^s \\
& y_i^a(\mathbf{u}^\top\Theta^*\mathbf{x}_i^a + b) \le 1 - \xi_i^a \\
& \xi_i^s \ge 0, \forall\, i = 1, 2, \ldots, n_s \\
& \xi_i^a \ge 0, \forall\, i = 1, 2, \ldots, n_a
\end{aligned}$$

where ξ_i^a and y_i^a are the i-th hinge loss and the pseudo label of auxiliary data respectively, and the optimization problem can be implemented by TSVM algorithms like [37].

Given a testing video, its original features are first mapped into the new feature space by Θ, followed by Equation (3) using $\mathbf{u}$ and b to predict its pattern.

At last, we give a brief analysis for time complexity of the algorithm: Before applying the two iterative steps, the right item in equation (3.10) can be pre-computed. The time complexity of SVM by sequential minimal optimization (SMO) is $o(n_s^{2.3})$, and that for matrix operation is $o(n_s mr)$. After achieving the optimal Θ, applying TSVM needs $o((n_s + 2n_a)^3)$. Suppose the iteration time is T, the overall time complexity for MMA is $o(T(n_s^{2.3} + n_s mr) + (n_s + 2n_a)^3)$.

3.3 Experiments and analysis

In this section we conduct the experiments to evaluate the performance of our proposed MMA model and provide the corresponding analysis.

3.3.1 Dataset and experiment setup

We conduct the experiments on two public datasets.

- Columbia consumer video (CCV) [36]. It is a collection of 20 classes of events or objects with 4,659 training videos and 4,658 testing ones, and all these videos were captured by ordinary consumers without post-editing. The 20 events/objects are *Basketball, Baseball, Soccer, IceSkating, Skiing, Swimming, Biking, Cat, Dog, Bird, Graduation, Birthday,*

WeddingReception, WeddingCeremony, WeddingDance, MusicPerformance, NonMusicPerformance, Parade, Beach, and *Playground*. According to the given label set, most of the videos contain only one specified pattern, and a small number of them contain more than one patterns. About 1,000 videos in both training and testing sets do not contain any specified patterns. So in our experiments, CCV dataset is divided into 20 independent recognition tasks. For each pattern, only the first 60 positive labelled videos are selected for training, and the remaining labelled ones are treated as unlabelled auxiliary videos (i.e., their labels are removed to get the unlabelled auxiliary videos).

- HMDB [45]. This is a human motion dataset for action recognition. There are 51 actions in this dataset with three different divisions for training and testing. Each action has 70 labelled videos. Similar to CCV dataset, for each action only the first 30 instances are selected as the labelled videos for training, and the rest of them are regarded as unlabelled auxiliary videos.

For video features, the combination of several low-level features is preferable for recognition tasks because of the different perspectives from their observations. The CCV videos are represented by SIFT (5,000d), STIP (5,000d) and MFCC (4,000d) features. In our experiments, the three given features were concatenated as 14,000-dimensional vectors, and KPCA was applied with χ^2 kernel to reduce the dimensionality to 6,000. So the final video representations used for CCV dataset are 6,000d combined features. For HMDB dataset, only STIP feature (4,096d) which can carry motion information was adopted for the action recognition task.

To evaluate the performance, we use average precision (AP) and normalized detection cost (NDC) for CCV dataset, and accuracy for HMDB dataset as the evaluation metrics. AP is a single-valued measurement that can reflect the ranking of the testing data. Let R as the true positive video in the testing set and R_j is the numbers of relevant videos in the top j list. AP is calculated as:

$$AP = \frac{1}{R} \sum_j \frac{R_j}{j} \times E_j \tag{3.12}$$

where $E_j = 1$ if the j-th video is correctly detected and 0 otherwise. AP prefers the high-rank correct videos, and the model with higher AP is in favour.

Different from AP, NDC uses a flexible way to evaluate the performance. For each event, we set a recognition threshold to the minimum value of all relevant videos in testing set. If the videos

TABLE 3.1: MAP comparison of MMA and other algorithms on CCV dataset

	SVM	**SAIR**	**TSVM**	**A-MKL**	**MFCU**	**MLHR**	**MMA**
MAP	0.1547	0.1751	0.3001	0.2612	0.3094	0.3109	**0.3149**
NDC	0.6831	0.6578	0.6371	0.6516	0.6282	0.6335	**0.6232**

TABLE 3.2: Accuracy comparison of MMA and other algorithms on HMDB dataset

	SVM	**SAIR**	**TSVM**	**A-MKL**	**MFCU**	**MLHR**	**MMA**
Accuracy	0.1096	0.1247	0.1454	0.1178	0.1443	0.1456	**0.1461**

with confidence scores higher than the threshold, they are considered to be relevant. Suppose P_{MD} and P_{FA} be the missing detection rate and false alarming rate respectively, and P_T be the proportion of true relevant videos, NDC is defined as:

$$NDC = \frac{\gamma_1 \times P_{MD} \times P_T + \gamma_2 \times P_{FA} \times (1 - P_T)}{\min(\gamma_1 \times P_T, \gamma_2 \times (1 - P_T))} \tag{3.13}$$

where γ_1 and γ_2 are the weight parameters on the importance of P_{MD} and P_{FA} respectively. In our evaluation we set $\gamma_1 = 80$ and $\gamma_2 = 1$ respectively. It can be seen that the lower the NDC is, the better performance the model has.

The recognition accuracy is used as a statistical measure of how well the experimental test correctly identifies or exclude a condition. i.e., the proportion of the true results among the total number of cases examined.

3.3.2 Experiment results

Comparison with baseline methods

We first compare the proposed method MMA with three baseline methods: χ^2-SVM, semantic analysis via intermediate representations (SAIR) [59] , transductive SVM (TSVM) [37], domain transfer multiple kernel learning (DTMKL) [18], multiple feature correlation uncovering (MFCU) [94] and multi-feature mearning via hierarchical regression (MLHR) [103]. SAIR is a supervised learning algorithm that jointly extracts the intermediate representation implicitly and trains the

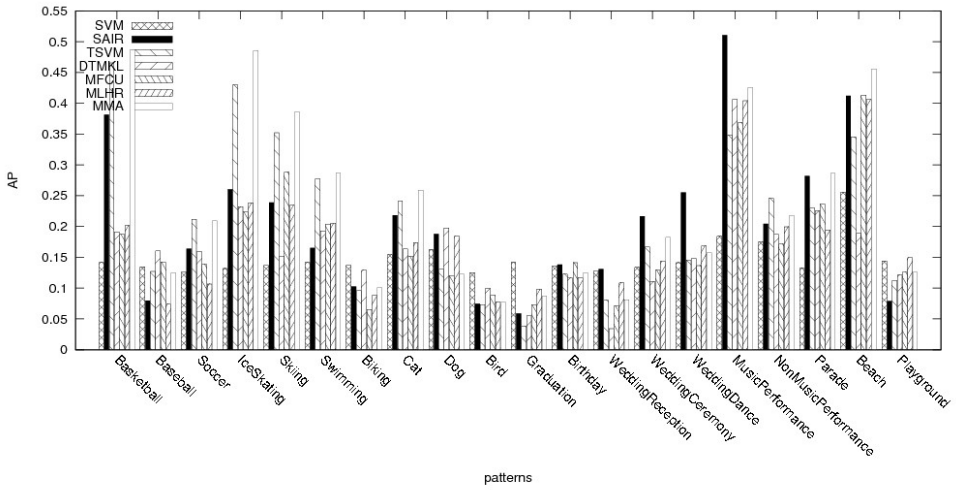


FIGURE 3.1: AP comparison of different algorithms on CCV

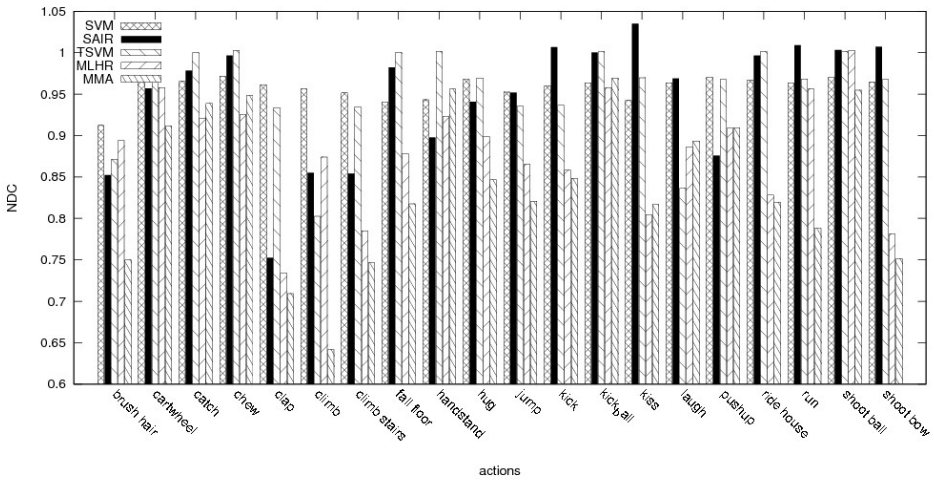


FIGURE 3.2: NDC comparison of different algorithms on CCV

objective classifiers. TSVM is an extension of SVM that can utilize unlabelled data to improve the generalization ability. DTMKL is a tranductive transfer model that can combine several kernels to achieve the data distribution similarity. Both MFCU and MLHR adopt the multi-view strategy to locally discover the structure of patterns, and MFCU uses similar subspace mapping strategy like our MMA model. The comparison of mean AP (MAP) and NDC for CCV dataset is shown in

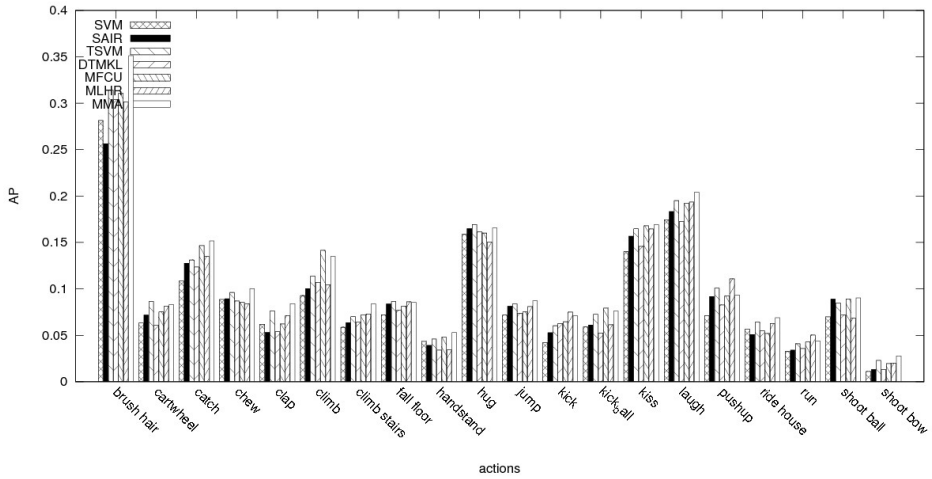


FIGURE 3.3: Accuracy comparison of different algorithms on HMDB (partial results)

Table 3.1. We also give AP, NDC and accuracy comparisons of some specified patterns in Figure 3.1, 3.2, 3.3.

From the table and the figures we can see that our proposed MMA model outperforms the other algorithms on both AP and NDC in average. SAIR algorithm tries to extract some intermediate feature representations, which is quite similar to the mapping function Θ in our proposed MMA model, and it outperforms the basic χ^2-SVM. However, it cannot discover the latent structure from unlabelled videos. TSVM utilizes the unlabelled instances and adopts the heuristic approach to find the maximum margins of different patterns, but it does not use the data distribution information. SAIR and TSVM can be considered as complementary methods to each other for pattern recognition tasks. Although DTMKL uses multiple kernels and their combinations to achieve the minimization of data distribution, its effect is limited, which can be further improved by semi-supervised algorithms. MLHR adopts both local and global feature structures and semi-supervised classification methods to improve the generalization ability, and similarly MFCU uses mapping strategies for several visual features correspondingly. Both MLHR and MFCU are very competitive algorithms.

Our proposed MMA algorithm can take the advantages of the methods above, and at the same time utilizes the label information in learning the mapping function, which can maximize the

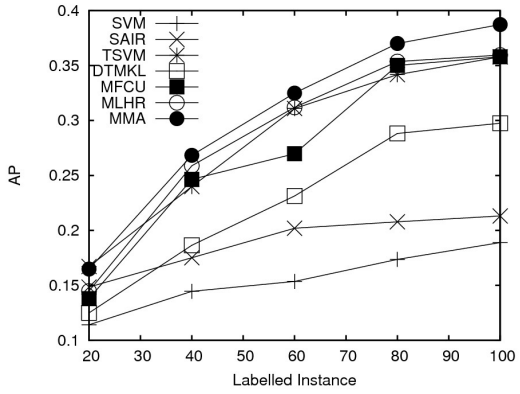


FIGURE 3.4: AP effect of different numbers of labelled videos on CCV

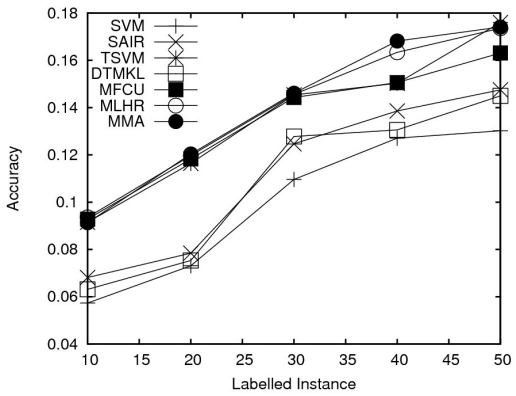


FIGURE 3.5: Accuracy effect of different numbers of labelled videos on HMDB

geometric margin between positive labelled video and negative labelled videos for a pattern.

Generally the performance on CCV dataset is better than that on HMDB dataset in our experiment, although the evaluation metrics are different. The reasons could be multi-fold. The most important reason we believe is due to the visual feature used on HMDB dataset (with STIP feature only), while three different features are combined in CCV dataset. Another reason is the number of training instances. Meanwhile, action recognition on HMDB dataset could be more challenging than object detection on CCV dataset.

The effect of labelled instance numbers

The quantity and quality of labelled data instances play an important role in supervised or semi-supervised learning. Actually in many machine intelligence systems, manual labeling and expert annotation still cannot be replaced by unsupervised approaches currently. Similar to other learning models, the performance of our MMA is also influenced by the number of labelled data instance. In this experiment, we study the effect of different numbers of labelled videos on the performance for each pattern, ranging from 20 to 100 for CCV dataset and from 10 to 50 for HMDB dataset respectively. Curves of AP and NDC are plotted in Figure 3.4 and 3.5.

From the figures we can see that as the numbers of labelled videos increase, both AP and NDC become better in all algorithms. This is reasonable since more labelled videos contain richer data distribution information which can benefit pattern recognition tasks.

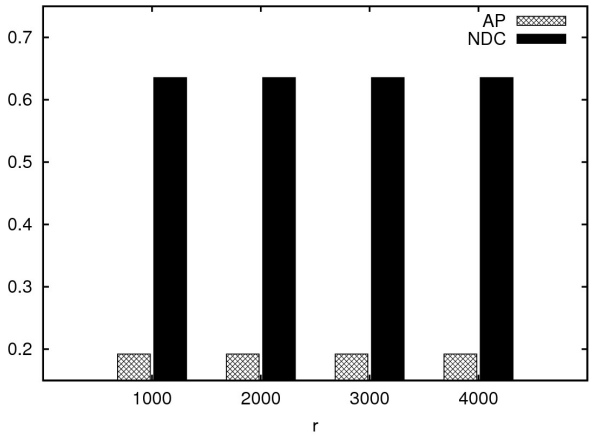


FIGURE 3.6: The effect of the mapped dimensionality r on CCV dataset

For HMDB dataset, the accuracy χ^2-SVM using all the labelled actions (70 instances for each action) is around 20% according to [45], and with partial labelled instance and MMA model, we can also get the satisfactory results, which tells using MMA model with less labelled data instances, we can also get very satisfactory recognition results.

Sensitivity of r

Since in the MMA model the matrix Θ is used to map the original feature space to a lower dimensional space, the selection of r which is the dimensionality of the new subspace may affect the recognition performance. We also conducted the experiments with different values of r, ranging from 1,000 to 4,000 in CCV dataset, and observed the changes of the performance as shown in Figure 3.6 . It is indicated that there are very tiny fluctuations when the subspace dimensionality changes, which signifies our MMA model is quite stable and insensitive to the parameter r. A relatively small r can be chosen for its more compact representations. The similar conclusion can be proved on HMDB dataset.

3.3.3 Discussions

Does unlabelled data help

Although semi-supervised learning with unlabelled auxiliary data has been shown more effective than supervised methods in many datasets, including the two datasets used in this work, it is still unclear whether it is always an effective approach in all real-world applications. In video content analysis, when there are large amount of unlabelled data that are near-duplicate to the labelled samples, semi-supervised learning can boost the performance in most cases. However, when video data has large intra-class variations and very rich semantics, semi-supervised learning actually degrades the performance of pattern recognition systems. One explanation is that the consistence of the data distributions between labelled source data and unlabelled auxiliary data directly affects the performance. Unfortunately, there is no theoretical proof to provide an accurate error bound on the performance changes, except experimental results on different datasets.

Early or late feature fusion strategy

Currently there are two feature fusion strategies, including early fusion and late fusion. The former one combines the features before the conduction of recognition, while the latter one combines the outputs of models recognized from different features [23, 16, 82, 8]. In our experiments on CCV dataset, MMA adopts the early fusion strategy while MLHR uses the late fusion strategy. Both strategies have their merits and demerits, and it is really unclear which one is always better than the other.

Scalability

For recognition tasks, the improvement of accuracy is often accompanied with the increase of time complexity. Due to the fact that the proposed MMA model takes several of the advantages of TSVM and SAIR, the training process needs more time because it involves the large matrix computation, especially the reverse of the matrix in Equation. 3.10. However, the time complexity of MMA is much lower than MFCU and MLHR, because the computation of adjacency matrix for graph decomposition is time-consuming. For large scale pattern recognition tasks, the scalability can be improved by hashing methods, but it may sacrifice the accuracy.

Relations with event detection

Compared with event detection, video pattern recognition can be considered as a basic work. One difficulty of event detection is the large intra-class visual difference but similar semantic descriptor. Besides, the mutual correlations among different patterns vary under a variety of circumstances. Thus semi-supervised algorithms and transfer learning models can hardly achieve the satisfactory results. However, it is worth trying to get appropriate intermediate feature representations, so unlabelled instances may help improve the detection performance.

3.4 Summary

In this work we proposed a video pattern recognition model called MMA by learning a mapping matrix Θ to transform the original feature space into a lower dimensional subspace, which can effectively reduce the data distribution disparity between labelled and unlabelled videos and also maximize the margin between positive labelled and negative labelled videos. The iterative optimization solution to the objective function of MMA may make use of both labelled source videos and unlabelled auxiliary videos for complex pattern recognition. The experimental results on two public datasets demonstrated that the proposed MMA model outperforms the state-of-the-art methods.

In the next Chapter, we propose a deep model that can combine both static and dynamic properties. Our model can learn the temporal patterns from static features, to achieve satisfactory performance in video event detection.

Chapter 4

Robust spatial-temporal deep model for multimedia event detection

4.1 Overview

In the last chapter, we proposed a learning model based on intermediate representation and domain adaptation. Multimedia event detection (MED) is a specific case for video pattern recognition, which is a much more challenging task. As is distinct from the traditional machine learning techniques as introduced in the last chapter, in this chapter we propose a novel learning framework based on 3D auto-encoder to sketch the spatial-temporal properties of videos.

Recently, the learned visual features from pre-trained Convolutional Neural Network (CNN) have shown a much more superior performance than hand-crafted features, and they have been used in different visual tasks such as pedestrian detection [67] and video classification [41, 80]. Specifically in the task of video event detection, Xu et al. proposed an effective video representation method based on a pre-trained CNN model [100]. Applying the VLAD quantization method [33] on the last pooling layer and two fully-connected layers, the detection performance is outstanding on the TRECVID MED dataset. However, this video representation is unsequenced, so it loses all the temporal information. Another issue is that it cannot make use of a large number of unlabelled videos to acquire prior knowledge, which could help improve the detection performance of the MED models. Based on these reasons, we set two goals: 1) we aim to learn the joint

models that can sketch both static and dynamic properties; 2) we aim to find the hidden spatial-temporal patterns in both labelled and unlabelled videos, as prior knowledge, before we obtain the final models.

In this work, we propose a novel deep learning framework for MED tasks. First we extracted the frame-level visual features and applied the temporal segmentation techniques to represent each video as a sequence of feature vectors. After that, we used the recurrent neural network to learn the intrinsic temporal properties of the videos in an unsupervised way. Finally, we aggregated the feature vectors and used an activation layer that connected to the labels. In this way we conducted a fine-tuning procedure to obtain the final detection model. We undertook extensive experiments on the TRECVID MED 2014 dataset, and demonstrated that our proposed system can achieve very promising performance in video event detection tasks.

4.2 A two-step training model for multimedia event detection

In this section, we introduce our proposed spatial-temporal deep learning model specifically designed for MED tasks in detail. Our main idea is to utilize both unsupervised recurrent video reconstruction and supervised fine-tuning, to construct a model that simultaneously captures the spatial and the temporal properties of the videos.

4.2.1 Video temporal segmentation

Our framework starts with the video representation. In order to sketch the temporal properties, a common way is to use frame sequences to represent videos. However, it is extremely time and memory consuming if we extract the visual features from all frames, so in most applications, frames are often sampled for further processing. In some cases the video has rapid visual changes, and it is difficult to sample and select the most appropriate frames, because a low-sampling rate may lead to the loss of important information, while a high-sampling rate usually results in redundancies. Through our observation, the temporal granularity for MED is comparably large, so a better option to sketch the spatial-temporal properties of videos is to split each video into several short segments. By doing so the number of temporal units can be greatly reduced, thus in the

learning procedure, the computational complexity can be also reduced.

Based on such analysis, we applied the *Kernel Temporal Segmentation* (KTS) [67] to segment the videos. Using this approach, the visual features of a video at the frame level were extracted and the kernel matrix computed to represent the affinities of the frames. Then dynamic programming was applied to discover the changing points, which represent the jumps of the signal. As a result, each video is represented as a sequence of non-overlapping segments. This kernel temporal segmentation has originally been used for video summarization, but in our proposed framework, we have only used it as a pre-processing procedure for the videos, to reduce computational complexities.

Representing a video as a sequence of frame feature vectors or segments for the MED task can be considered as a kind of multiple-instance learning. However, the detection accuracy reported in [46] shows that both mi-SVM and MI-SVM have far inferior performances compared to the SVM trained on the flat vector representation of the videos. This implies we need to build the model carefully, in order to avoid the negative effect brought by the overly-loose assumption of multiple-instance learning. In addition, the temporal information should also be incorporated into the model to improve detection accuracy.

4.2.2 Video re-construction using the GRU auto-encoder

A key step of our proposed framework is to reconstruct the video representation in an unsupervised way using a recurrent auto-encoder.

The traditional auto-encoder [29] aims to reproduce the input data as a compressed and similarly distributed representation. Since in the hidden layer the dimensionality is lower than that of the input layer, the auto-encoder can be considered to make a non-linear dimensionality reduction. Normally the auto-encoder is used in the deep multi-layer perceptron (MLP) by pre-training the weights of the neural networks in a greedy layer-wise approach, which is an unsupervised procedure.

Unlike the conventional auto-encoder which is mainly used in image processing, we used a recurrent neural network (RNN) to reconstruct the video representations. An MLP can only map from input to output vectors, while an RNN can in principle map from the history of previous inputs

to each output, so it is able to deal with sequential prediction problems, especially for sentence understanding in natural language processing [26]. In multimedia content analysis, RNNs have also been used in acoustic novelty detection [61] and video highlight extraction [102]. Among different RNN models, the Long Short-Term Memory (LSTM) [30] and the Gated Recurrent Unit (GRU) [13, 14, 38] are the most popular ones, and both of them can overcome the difficulty of *vanished gradient* problem [4]. Although there are a number of structural differences between LSTM and GRU, it is generally unclear which types of gating units have a better performance.

In this work we chose the GRU to reconstruct video representations, the reason being that it has fewer parameters for optimization, thus the computational complexity is comparably lower than that for the LSTM.

After the kernel video segmentation introduced in the previous subsection, assume a video $\mathbf{x}$ is represented as a sequence of feature segments $\mathbf{x} = (x_1, \ldots, x_T)$, where $x_t \in \mathbb{R}^d$ is the visual feature vector of the t-th segment of the video $\mathbf{x}$, $t = 1, \ldots, T$, and T is the number of the video segments. An RNN based auto-encoder first computes the hidden vector sequence $\mathbf{h} = (h_1, \ldots, h_T)$, $h_t \in \mathbb{R}^{d'}, d' < d$, then it computes the output sequence $\tilde{\mathbf{x}} = (\tilde{x}_1, \ldots, \tilde{x}_T)$, which is an approximation of $\mathbf{x}$, i. e, $\tilde{\mathbf{x}} \approx \mathbf{x}$.

The GRU has gating units that model the flow of information inside the unit without separate memory cells. Let h_t^j be the j-th activation at the time t, which is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\tilde{h}_t^j$:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j \tag{4.1}$$

where z_t^j is the update gate that decides how much the unit updates its activation. z_t^j is computed by:

$$z_t^j = \sigma(W_z x_t + U_z h_{t-1})^j \tag{4.2}$$

where σ is the sigmoid activation function. W_z is the weight matrix for the update gate, and U_z is the self-connected weight matrix for the hidden layer. The candidate activation $\tilde{h}_t^j$ is computed by:

$$\tilde{h}_t^j = \tanh(W x_t + U(r_t \odot h_{t-1}))^j \tag{4.3}$$

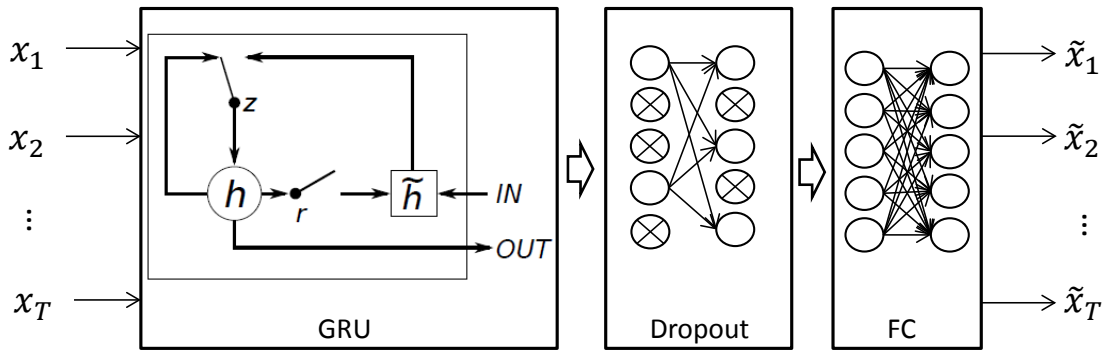


FIGURE 4.1: The GRU based recurrent auto-encoder.

where W and U are weight matrices for the candidate activation. r_t is a set of reset gates and $\odot$ is an element-wise multiplication. When r_t^j is close to 0, the reset gate can make the unit act as if it is reading the first symbol of an input sequence, allowing it to forget the previous state. r_t^j is computed as:

$$r_t^j = \sigma(W_r x_t + U_r h_{t-1})^j \tag{4.4}$$

where W_r and U_r are the matrices for the reset gate. Similar to other feed-forward neural network models, the optimization of the GRU can be achieved by gradient descent methods. For further details of the GRU, please refer to [13].

Our GRU based video reconstruction model is illustrated in Figure 4.1. The input of the neural network is a 3D tensor, where the first dimension is the number of training videos, the second dimension is the temporal video segments, and the last one is the dimensionality of the visual feature vector. The GRU recurrent layer is connected to the input layer, followed by a dropout layer. The dropout layer is a simple yet effective way to reduce over-fitting and thus is a major improvement over other regularization methods [77]. After the dropout layer, we apply the fully-connected layer (FC) with the linear activation that is connected to the output. The output is exact the same as the input video segments. As a result, we can just apply the L_2 loss to optimize the parameters in the model:

$$L_u = \|\tilde{\mathbf{x}} - \mathbf{x}\|_2 \tag{4.5}$$

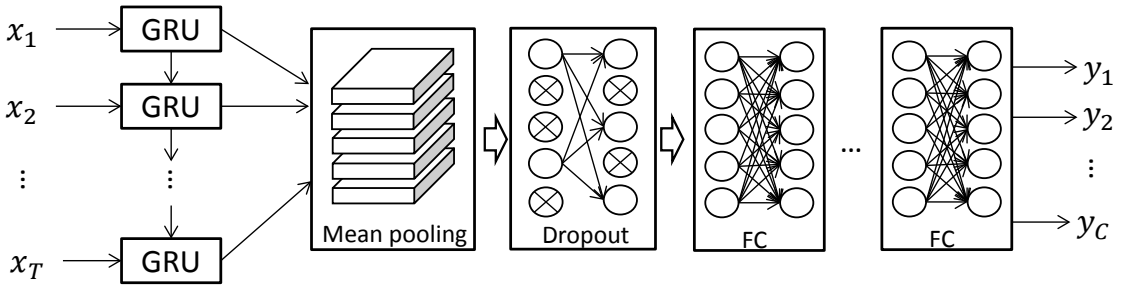


FIGURE 4.2: The prediction model for event detection.

The advantages of utilizing unsupervised recurrent learning are summarized as follows: first of all, it can make a much better initialization of the neural network; second, it can well reflect the temporal information between video segments; last but not least, a large number of videos without any label information can be used to enhance the robustness of the model and avoid the over-fitting problem.

4.2.3 Supervised fine-tuning for event detection models

After the unsupervised recurrent learning, the output from the GRU layer is the reconstruction of the original video segments. In the supervised learning phase, we just kept the GRU layer, removed the dropout layer and FC layer as introduced above, and added several new layers: a mean pooling layer, a new dropout layer and one or more new FC layers. The last FC layer was connected to the labels of the event categories $\mathbf{y} = (y_1, \ldots, y_C)$ with a softmax activation, and C is the number of event categories. In our experiment, we used the cross-entropy as the loss function:

$$L_s = -\mathbf{y} \log f(\mathbf{x}) - (1 - \mathbf{y}) \log(1 - f(\mathbf{x})) \tag{4.6}$$

where f is the final prediction function of the model. The supervised learning model is illustrated in Figure 4.2.

During the supervised training phase, the weight matrix that connects the dropout layer and label layer was randomly initialized and optimized, while the GRU layer only needed to be fine-tuned without dramatic fluctuations.

4.3 Experiments and discussions

4.3.1 Dataset and data pre-processing

We have conducted the event detection experiment on the TRECVID MED Test 2014 dataset [1]. This dataset contains 8,030 training videos, and 23,953 testing videos over 20 complex events under unconstrained cases, and we have followed the NIST official data split *100Ex* to train and evaluate our model. In the training phase there are approximately 100 positive videos for each event, while the rest are irrelevant to any pre-defined events. As our proposed framework is able to deal with a large number of unlabelled videos in a "semi-supervised" way, besides all the labelled training videos, we randomly select 4,000 videos from the testing set in the procedure of recurrent video reconstruction described in Section 4.2.2.

Recently the Convolutional Neural Network (CNN) has given a very high accuracy in image classification. With a GPU device, the deep feature can be efficiently extracted from the raw images based on a pre-trained CNN model. In our experiment we adopted the VGG ILSVRC model [75] to obtain the visual features from the videos. In the VGG-net, the first 13 layers are 2D convolutional and max-pooling layers, followed by 3 fully-connected layers. In Xu et al.'s work, the descriptors from the *pool5* layer can be considered as the latent concepts to describe the local properties of the videos [100]. The output dimension of this layer is a $7 \times 7 \times 512$ data cubic, thus they can be decomposed into 49 vectors with 512 dimensions. Applying the VLAD pooling method, the video can be finally represented as a high-dimensional vector, and such video representation has been proved to outperform the same feature aggregation of 6th and 7th fully-connected layers. The reason for this is that the fully-connected layers are highly influenced by the label information of ImageNet, so they have a slight inability to be generalizable. Following Xu et al.'s work, we extracted the feature from the 5th pooling layer, and applied the VLAD method to aggregate the latent concepts. As a result, each video segment is represented as a 65,536 dimensional vector.

4.3.2 Experimental settings

We set different output dimensions of the GRU layer: 128, 256, 512, and 1,024 respectively. In the dropout layers in both unsupervised and supervised phases, we set the dropout rate to 0.5 without any change. In the optimization procedure, we applied the AdaDelta algorithm [107] to optimize the parameters in the neural network. The Adadelta algorithm restricts the windows of accumulated past gradients to the fixed size, and it does not need to set a default learning rate, as it has been eliminated from the update rule. Theoretically, AdaDelta can achieve faster convergence than SGD (Stochastic gradient descent) [1] and avoid the fluctuations.

We used Mean Average Precision (MAP) to evaluate the detection performance, which is a single-valued measurement that reflects the ranking of the model on testing data. The model with the higher MAP value is in favour.

Our experiment was conducted on a workstation equipped with the Ubuntu operating system and a GeForce GTS Titan GPU device. We implemented the system based on the Theano package [2], which supports GPU acceleration and automatic gradient calculation. The error threshold of the model was set to 10^{-5}, and the program ran for about 3 days until convergence.

4.3.3 Baselines

We then compared the event detection performance of our proposed two-step neural network model with three baseline models: 1)Multiple instance SVM (MI-SVM) [17]; 2) Inferring Temporal Instance Labels (ITIL) [46]; and 3) VLAD aggregation of the pool5 feature, combined with SVM as a classifier (LCD+VLAD) [100]. In addition, we also reported the experiment results using the GRU+FC model in a purely supervised way (GRU+FC+S), i.e., without the unsupervised video reconstruction step proposed in Section 4.2.2. Our proposed method of applying the unsupervised GRU auto-encoder and supervised fine-tuning is marked as GRU+FC+US.

[1] https://en.wikipedia.org/wiki/Stochastic_gradient_descent

[2] http://deeplearning.net/software/theano/

TABLE 4.1: The effect of the dimensions in the GRU layer

D_{gru}	**128**	**256**	**512**	**1024**
MAP	0.2856	0.3112	0.3283	0.3282

TABLE 4.2: The effect of the FC layer numbers

N_{fc}	**1**	**2**	**3**
MAP	0.3084	0.3283	0.3283

4.3.4 Results

Analysis of the impact of parameters

There are two parameters to be pre-set for our proposed framework: The mapping dimension of the GRU layer in the unsupervised video reconstruction, and the number of fully-connected layers in the supervised fine-tuning phase.

The dimension in the GRU layer is much smaller than the input data, so the recurrent auto-encoder is considered as a non-linear dimensionality reduction with the consideration of temporal information. We set the reduced dimensions D_{gru} to 128, 256, 512 and 1,024 respectively, and used a single fully-connected layer to the label layer. The MAP performance is displayed in Table 4.1. The table shows when the dimension of the GRU layer is set to 256 or 512, the model can ensure similar detection performance.

In the popular deep neural network models, there are generally two or more FC layers that are connected to the label layer. For example, the VGG net has 3 FC layers. The explanations are illustrated as follows: first, more FC layers can approximate more complex non-linear structures in the high-dimensional data space; second, the weights in the "shallower" layers that may reflect the input data structure are less affected, so this can alleviate the over-fitting problem. We set the number of FC layers (N_{fc}) to 1, 2 and 3, and the resulting detection performance of our model is displayed in Table 4.2.

From the table we can see that when there are more than two FC layers in our network model during the fine-tuning phase, the detection achieves better performance.

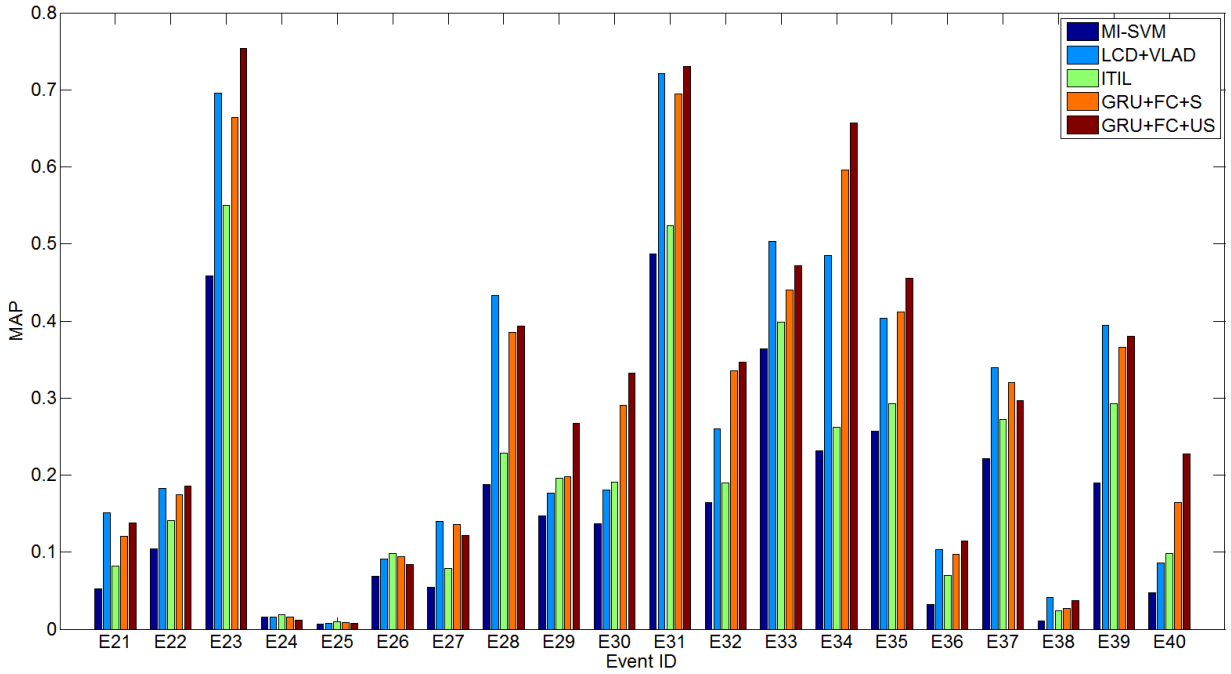


FIGURE 4.3: The AP comparisons of different models.

TABLE 4.3: The overall MAP comparisons of different models

Model	**MI-SVM**	**LCD+VLAD**	**ITIL**	**GRU+FC+S**	**GRU+FC+US**
MAP	0.1624	0.2805	0.2241	0.3012	0.3283

Comparisons with other MED models

We conducted the MED experiment using all the baseline models: MI-SVM [17], LCD+VLAD [100], ITIL [46], GRU+FC+S and GRU+FC+US, and the experiment results are displayed in Figure 4.3 and Table 4.3. We can see our proposed two-step learning model outperforms others, so we can conclude that by simultaneously considering the spatial and temporal information of the video segments, the data representation of the recurrent auto-encoder can better reflect the intrinsic structure of the discriminative patterns. Thus our model leads to superior results, and the accuracy is very close to the MED system implemented by CMU [106].

Figure 4.4 shows the key frames in videos of classes E31 - E35 that are top-ranked by our proposed model. The red boxes are misclassified videos, i.e., they are irrelevant to the pre-defined events. We can see that our framework can successfully detect important visual cues for the events.

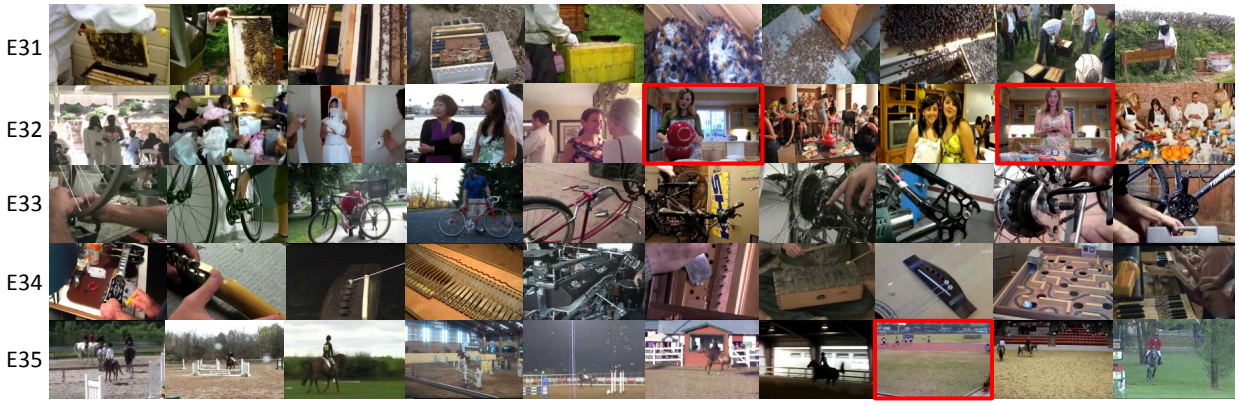


FIGURE 4.4: The top 10 ranked videos for events E31 - E35.

4.3.5 Discussions

The success of our proposed two-step learning model for the MED task can be attributed to a number of factors. First of all, the promise of the video recurrent auto-encoding is that by exploiting the massive amount of unlabelled video data, we can learn a superior initial value for the weights in the GRU layer. The self-taught learning step has been proved to result in much better classifiers in the later supervised learning procedure. Second, after the unsupervised learning to reconstruct the videos, the weights in the GRU layer can start at a better location in the parameter space than if they are initialized at random. As a result, the gradient from such a location is more likely to yield a better minimum in the neural network, because the unlabelled videos have already provided a significant amount of prior information about the hidden patterns of the input videos. Third, applying recurrent video reconstruction may help discover some temporal properties of the videos, which can boost the descriptive power of the learned video representations.

4.4 Summary

In this work we have proposed a novel framework for the MED task by conducting a two-step learning model. First we used the GRU model as an unsupervised procedure for the recurrent video reconstruction, and then we used the better initialized GRU layer and mean pooling, to generate visual features for the final detection model. Our proposed framework simultaneously

considers the spatial and temporal properties of videos, so it can effectively boost the performance of state-of-the-art MED models.

Chapter 5

Web video event recognition by semantic analysis from ubiquitous documents

5.1 Overview

In this work, we study the problem of web video event recognition, where web videos often describe large-granular events and carry limited textual information. Key challenges include how to accurately represent event semantics from incomplete textual information and how to effectively explore the correlation between visual and textual cues for video event understanding. We propose a novel framework to perform complex event recognition from web videos. In order to compensate for the insufficient expressive power of visual cues, we construct an event knowledge base by deeply mining semantic information from ubiquitous web documents. This event knowledge base is capable of describing each event with comprehensive semantics. By utilizing this knowledge base, the textual cues for a video can be significantly enriched. Furthermore, we introduce a two-view adaptive regression model which explores the intrinsic correlation between visual and textual cues of the videos to learn reliable classifiers.

The framework proposed in this work is illustrated in Figure 5.1. Given a set of training web videos, both visual and textual features are firstly extracted. A large number of event-relevant documents are then collected from web sources by searching event keywords. After that, for each event an event semantic graph (ESG) is constructed to have a comprehensive representation of the

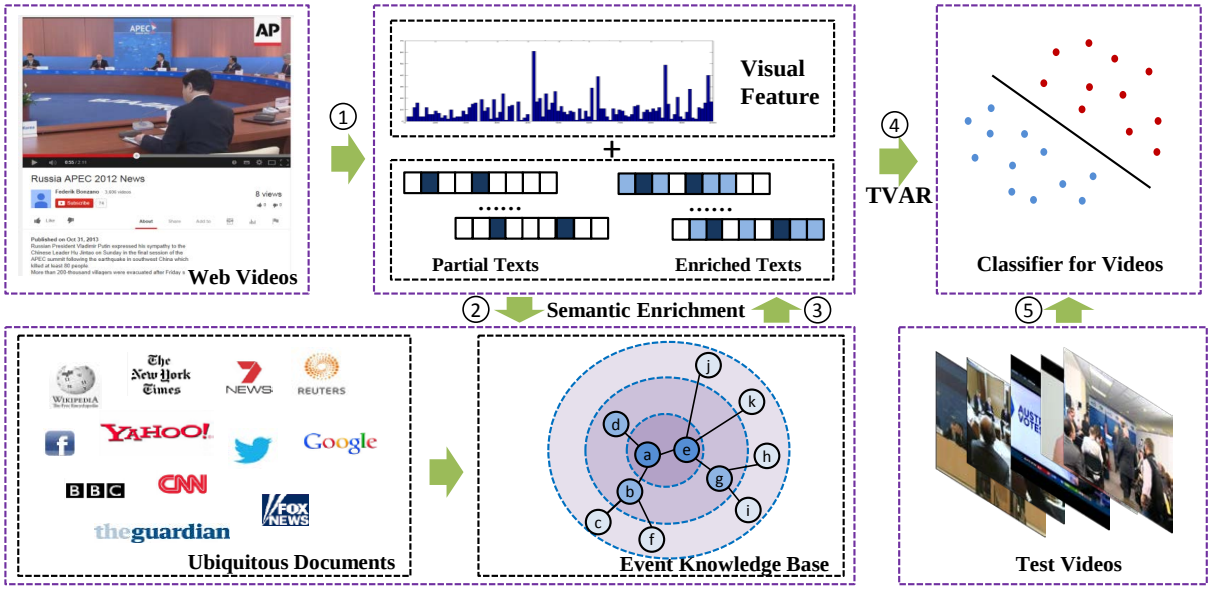


FIGURE 5.1: The flowchart of the proposed framework.

event semantics. All ESGs form the event knowledge base. By utilizing the ESGs, videos' textual cues can be enriched. Finally, we use both enriched textual and visual cues to train a two-view adaptive regression (TVAR) model and realize the knowledge adaptation from enriched auxiliary texts to videos for event recognition.

To the best of our knowledge, the proposed framework is the first attempt to utilize event semantic structures obtained from ubiquitous web documents as the knowledge base for video event analysis. There are several advantages of this framework. Firstly, compared with knowledge adaptation from different visual feature sources [10], the auxiliary text source is more accurate in semantic description and much easier to access. Secondly, web videos and the event knowledge base generated from web documents are complementary to each other. The resulting coherence is useful for event understanding. Last but not least, with the help of the generated event knowledge base, our framework can deal with video events of large granularities with rich semantics.

5.2 Event Knowledge Base

5.2.1 Knowledge base construction

Videos from web services like YouTube are often attached with some text information such as titles, tags and descriptions. From the semantic perspective, such textual information is potentially useful to describe the events. In order to complement the expressive power of the visual cues, a straightforward way is to make use of this auxiliary semantic information to assist learning a more effective model, thus enhancing the event recognition performance. The relevant semantic information could be achieved by searching the event keywords from Google or Bing, and such descriptions contained in relevant web pages can help enrich the partial texts associated with the web videos. However, in the learning procedure it is not trivial to directly involve the raw texts as the intermediate representation, due to the following problems. First, although the videos and auxiliary texts may be relevant to the same event, they often characterize it from totally different perspectives, which makes it difficult to achieve consistency between visual features and textual information. Second, the user-generated texts are often arbitrarily provided to the videos, which may not give complete and precise descriptions to the events. Third, as a semantic cue, mutual correlations among textual words are often ignored for event description.

To solve the above problems, we constructed an event knowledge base which consists of a set of event semantic graphs (ESGs). Each ESG represents an individual event from the semantic perspective. It is constructed by deeply exploring knowledge from related web documents and is capable of enriching the event semantics. An ESG is defined as an undirected graph with three layers. The central layer contains event keywords which are quite general for event description. The middle layer is constructed by those words extracted from other partial texts[1], while the outermost layer contains the additional words extracted from the event-relevant web documents[2] which can potentially describe the corresponding events in a more detailed way. In an ESG, the weight of the edge between two words indicates their mutual correlation in describing the event.

An illustrative example of the ESG for the event *APEC Russia* is shown in Figure. 5.3. From

[1]The partial texts refer to the titles, descriptions and tags (but not the event keywords) on the YouTube web pages.

[2]The web pages were redirected from search results on Google, and we applied the approaches of text preprocessing to extract the contents from these web pages.

	a	b	c	d	e
a	-	0.96	0.85	0.32	0.20
b	0.96	-	0.64	0.61	0.43
c	0.85	0.64	-	0.37	0.12
d	0.32	0.61	0.37	-	0.16
e	0.20	0.43	0.12	0.16	-

	a	b	c	d	e
a	-	0.96	0.85	0.32	0.20
b	0.96	-	0.64	0.61	0.43
c	0.85	0.64	-	0.37	0.12
d	0.32	0.61	0.37	-	0.16
e	0.20	0.43	0.12	0.16	-

	a	b	c	d	e
a	-	0.96	0.85	0.32	0.20
b	0.96	-	~~0.64~~	0.61	0.43
c	0.85	~~0.64~~	-	0.37	0.12
d	0.32	0.61	0.37	-	0.16
e	0.20	0.43	0.12	0.16	-

	a	b	c	d	e
a	-	0.96	0.85	0.32	0.20
b	0.96	-	~~0.64~~	0.61	0.43
c	0.85	~~0.64~~	-	0.37	0.12
d	0.32	0.61	0.37	-	0.16
e	0.20	0.43	0.12	0.16	-

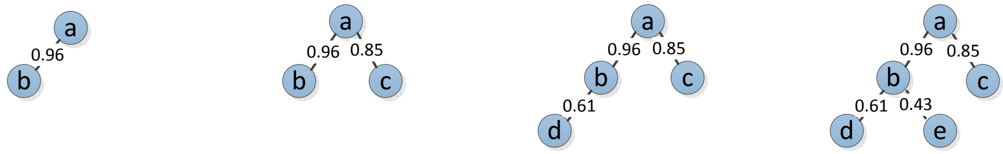


FIGURE 5.2: An illustration of an ESG formulation.

the figure, the event keywords *APEC* and *Russia* are in the central layer of the graph. The words from other partial texts attached to the videos are in the middle layer, and those words from the event-relevant web documents are in the outer-most layer of the graph. The value on the edge between two words indicates their mutual dependency for the specific event.

In order to discover the words' mutual correlations, a simple yet effective way is to compute their second-order product approximation of a joint probability distribution. Given two words t_m and t_n, their mutual information (MI) is defined as:

$$MI(t_m; t_n) = p(t_m, t_n) \log \frac{p(t_m, t_n)}{p(t_m)p(t_n)}. \tag{5.1}$$

Instead of building a universal ESG that can globally reveal the word correlations for all events, we constructed an ESG for each individual event. This is because the semantics of the same textual words as well as their dependencies may vary dramatically in different event contexts.

Suppose there are c events, and let e_q represent the q-th event ($q = 1, \ldots, c$), the conditional dependency between textual words t_m and t_n under the event e_q is measured by their conditional mutual information (CMI):

$$CMI(t_m; t_n|e_q) = p(t_m, t_n|e_q) \log \frac{p(t_m, t_n|e_q)}{p(t_m|e_q)p(t_n|e_q)}. \tag{5.2}$$

The use of CMI can be intuitively explained as follows: if two words co-occur more frequently in the texts that describe the event, their correlation should be greater.

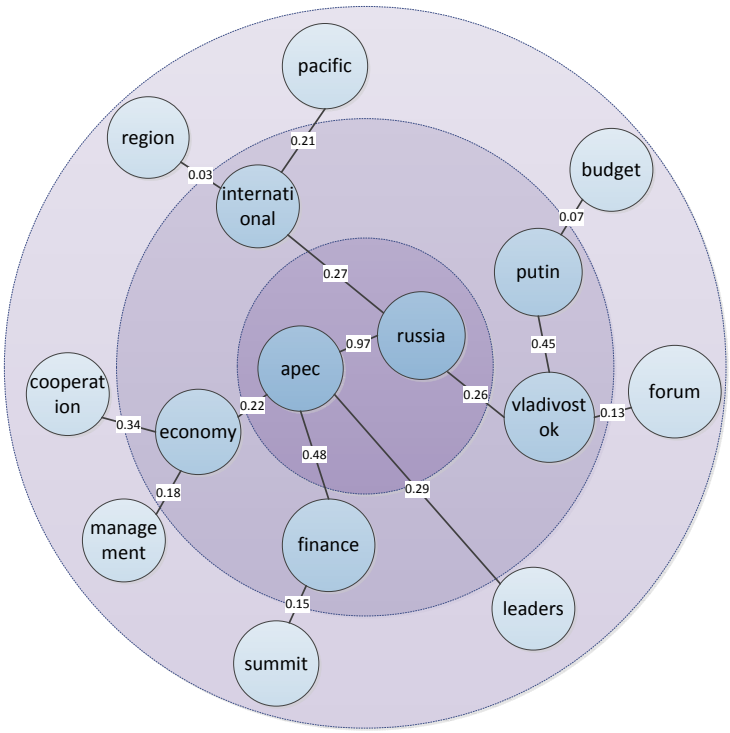


FIGURE 5.3: An illustration of ESG constructed for event *APEC Russia 2012*.

To construct an ESG for an event, we first calculated the pairwise conditional mutual information among words, and then found the maximum weight spanning tree according to the values of conditional mutual information. In Figure 5.2, we give an illustration of how to formulate a maximum weight spanning tree with 5 words. The identified maximum weight spanning tree is taken as the ESG for the event.

Since the graph is a tree-like undirected structure, it is acyclic. If there are N_q nodes in total for event e_q, only $(N_q - 1)$ edges are required to connect all of them. Therefore this property ensures the simplicity of the textual words propagation without cyclic structures.

All the ESGs form the event knowledge base, which is constructed based on the partial texts accompanying the videos and the event-relevant web documents. Since the partial texts contain limited semantics, the event knowledge base may greatly expand the contextual information by computing the probability of potential relevance from web documents that can also describe the event. It is expected that such an event knowledge base is able to help users understand the events comprehensively. How to collect event-relevant web documents will be explained in experiments.

5.2.2 Textual enrichment with knowledge base

With the aid of external web documents, for each event we built an ESG to describe the corresponding event with rich semantics. Next, we elaborate how to utilize the event knowledge base to enrich the textual cues for a training video.

We represent all the partial texts accompanying a video as a bag-of-words (BoW) vector $x^{(t)} \in \mathbb{R}^{d_t \times 1}$, where d_t is the size of vocabulary. Let $X^{(t)} = [x_1^{(t)}, \ldots, x_{n_s}^{(t)}]^\top \in \mathbb{R}^{n_s \times d_t}$ denote the matrix of all the vectors, where n_s is the number of the training videos, and each row of $X^{(t)}$ is a vector. We use $x_h^{(t)}[i]$ to represent the i-th entry of h-th vector ($1 \leq h \leq n_s$ and $1 \leq i \leq d_t$), and the value of it is the term frequency.

The goal of building the ESGs is to enrich the textual cues for each training video. Suppose a training sample $x_h^{(t)}$ from the q-th event contains the word t_m, i.e., $x_h^{(t)}[m] > 0$. The probability of t_m's neighbour node t_n in the q-th graph that can also be used to describe event e_q is calculated as:

$$p(t_n|t_m, e_q) = \frac{p(t_m, t_n, e_q)}{p(t_m, e_q)} = \frac{p(t_m, t_n|e_q)}{p(t_m|e_q)}. \tag{5.3}$$

Then, the n-th entry in the vector can be updated as:

$$\tilde{x}_h^{(t)}[n] = \max\{x_h^{(t)}[n], x_h^{(t)}[m] \times p(t_n|t_m, e_q)\} \tag{5.4}$$

Using the above update scheme, a set of complementary textual words can be derived using the constructed ESGs to enhance the original textual cues. Therefore, the updated BoW vector $\tilde{x}_h^{(t)}$ for a video can be understood as a set of word distribution variables, where more entries have non-zero values. The updated matrix $\tilde{X}^{(t)}$ where each row $\tilde{x}_h^{(t)}$ is the new representation of $\tilde{x}_h^{(t)}$ ($1 \leq h \leq n_s$) represents the new textual cues for all training videos.

Note that $\tilde{x}_h^{(t)}[i]$ is a real value instead of the discrete one, since the way we enrich the textual cues is probabilistic. Take the example in Figure. 5.3. Each node in this graph represents a textual word, and the weight of an edge indicates the conditional mutual information value between the two connected words. If one video's partial texts, i.e. the middle layer, contains the words *finance* and *putin*, then potentially *summit, vladivstok* and *budget* are complementary to the event description. Thus the textual cues of the video can be enriched according to the words' mutual correlations in the corresponding ESG under this event context.

The enrichment of the textual cues makes the matrix $\tilde{X}^{(t)}$ less sparse. However, the updated representations carry more textual information, which leads to higher discriminative ability for better event understanding.

5.3 A two-view regression model for video event recognition

Using the event knowledge base, we can obtain more comprehensive and precise text information for videos. In order to learn more reliable classifiers for event recognition, it is imperative to discover the intrinsic correlation between visual and enriched textual cues. However, this is a non-trivial task because 1) visual and textual cues are highly heterogeneous, and 2) only the visual cue is available in the testing phase for video event recognition. Based on the above reasons, we present a *two-view adaptive regression* (TVAR) model in this section, which can simultaneously capture their correlations and learn reliable classifiers.

5.3.1 Problem formulation

We represent the low-level visual feature set of the training videos as a matrix $X^{(v)} \in \mathbb{R}^{n_s \times d_v}$, where d_v is the dimension of the visual feature, n_s is the number of video samples, and each row $x_h^{(v)}$ ($1 \leq h \leq n_s$) is a video sample. Here we assume there are multiple target events for recognition, and $Y \in \mathbb{R}^{n_s \times c}$ is the label matrix where c is the number of events. If $x_h^{(v)}$ contains the k-th event ($1 \leq k \leq c$) then $Y_{hk} = 1$ else $Y_{hk} = -1$. To train a recognition model with $X^{(v)}$, $\tilde{X}^{(t)}$ and Y, the objective optimization function of our proposed model can be formalized as follows:

$$\min_{f^{(v)}, f^{(t)}} \mathcal{F}_v(f^{(v)}(X^{(v)}), Y) + \lambda \mathcal{F}_t(f^{(t)}(\tilde{X}^{(t)}), Y) + \gamma \Lambda(f^{(v)}(X^{(v)}), f^{(t)}(\tilde{X}^{(t)})) \tag{5.5}$$

where the first two terms $\mathcal{F}_v(\cdot, \cdot)$ and $\mathcal{F}_t(\cdot, \cdot)$ are the regression functions for visual and textual cues respectively, and they are balanced by a tunable parameter λ, $0 < \lambda < 1$. The third term $\gamma \Lambda(\cdot, \cdot)$ controls the consistency of the two views.

The reason to adopt the parameter λ is that the two models $\mathcal{F}_v(\cdot,\cdot)$ and $\mathcal{F}_t(\cdot,\cdot)$ can be optimized as two 1-dimensional projections, but the two views are completely heterogeneous, thereby they might show quite different value ranges. The visual view $\mathcal{F}_v(\cdot,\cdot)$ is dominant and its optimal solution is used for event recognition in the testing phase, whereas the textual view $\mathcal{F}_t(\cdot,\cdot)$ is auxiliary, so its importance should never be higher than the visual view.

For the low-level visual view of video data, a general model is to minimize the following objective function:

$$\mathcal{F}_v = loss(f^{(v)}(X^{(v)}), Y) + \alpha\Omega(f^{(v)}) \tag{5.6}$$

where $loss(\cdot,\cdot)$ is the loss function, $\Omega(\cdot)$ is the regularization term to avoid over fitting, and α is the balance parameter.

Similarly, with the consideration of the text matrix $\tilde{X}^{(t)}$ mentioned in the previous section, the objective function for the textual view can be described as:

$$\mathcal{F}_t = loss(f^{(t)}(\tilde{X}^{(t)}), Y) + \beta\Omega(f^{(t)}) \tag{5.7}$$

where the loss function and regularization term are balanced by β.

For the low-level visual view, we followed the SAIR model proposed by Ma et al. [59], which can simultaneously train an intermediate representation in an unsupervised way and obtain a target classifier:

$$\begin{aligned} \min_{W_v,\Theta} \mathcal{F}_v &= \left\|X^{(v)}\Theta W_v - Y\right\|_{2,1} + \alpha\|W_v\|_F^2 \\ s.t. &\Theta^\top\Theta = I \end{aligned} \tag{5.8}$$

where $\Theta \in \mathbb{R}^{d_v \times r}$ is the transformation matrix that can map the original feature space to a lower and condensed r-dimensional subspace ($r < d_v$). $W_v = [w_v^{1\top}, \ldots, w_v^{c\top}] \in \mathbb{R}^{r\times c}$ is the weight matrix on $X^{(v)}$ where each column is a weight vector for the corresponding event class. $\|\cdot\|_{2,1}$ and $\|\cdot\|_F$ are $\ell_{2,1}$-norm and Frobenius norm respectively.

The consideration of orthogonal matrix Θ has a two-fold reason: to preserve the most information of visual features, and to avoid over-fitting. Since the low-level visual features have much deeper semantics than natural languages, it is necessary to find an intermediate representation to bridge the large "semantic gap", and this joint optimization also ensures the intermediate representation can be tightly coupled with the high-level event descriptors. Another explanation for the adoption of Θ to improve the performance is it can be considered as the hidden layer of the models to realize the non-linear mapping, which is quite similar to the reason why multi-layer perceptron has the better performance than logistic regression.

For the textual view of video data, a simple regression model can be applied:

$$\min_{W_t} \mathcal{F}_t = \left\| \tilde{X}^{(t)} W_t - Y \right\|_{2,1} + \alpha \|W_t\|_F^2 \tag{5.9}$$

Dissimilar to visual features, here we did not use the transformation matrix because text features have comparative more "shallow" semantics and they could be easier to understand by both humans and machines. In our proposed model it is unnecessary to map the text feature to the lower dimensional subspace. In the next section and we demonstrate that the adoption of intermediate subspace transformation on the text feature has a very limited influence on the event recognition in the experiment.

The above two regression models can be optimized as two 1-dimensional projections. However, the two views are completely heterogeneous. In order to improve the recognition performance of $f^{(v)}$ by $f^{(t)}$, we combined the two models by introducing the similarity consistency of the two views between the two projections:

$$\Lambda = \left\| X^{(v)} \Theta W_v - \tilde{X}^{(t)} W_t \right\|_{2,1} \tag{5.10}$$

The above object function plays a key role to adapt the knowledge from textual cues to visual cues, because it may make some of the feature coefficients shrink to zero to facilitate the improvement of the two models (5.8) and (5.9). In the loss functions and consistency term, we adopted $\ell_{2,1}$-norm based regression, because it has been proved that its minimization is robust to outliers and efficient in computing [64].

Now we integrate the three optimization objectives (5.8), (5.9) and (5.10), and formulate them into a unified model:

$$\begin{aligned}\min_{\Theta,W_v,W_t} &\left\|X^{(v)}\Theta W_v - Y\right\|_{2,1} + \alpha\|W_v\|_F^2 + \\ &\lambda(\left\|\tilde{X}^{(t)}W_t - Y\right\|_{2,1} + \beta\|W_t\|_F^2) + \\ &\gamma\left\|X^{(v)}\Theta W_v - \tilde{X}^{(t)}W_t\right\|_{2,1} \\ s.t. &\Theta^\top\Theta = I\end{aligned} \tag{5.11}$$

When the optimal solutions W_v^* and Θ^* were achieved, we could apply them on testing videos for event recognition since there were no available textual cues in the testing phase. The decision function on testing videos $X_{test}^{(v)}$ is:

$$f^{(v)}(X_{test}^{(v)}) = X_{test}^{(v)}\Theta^* W_v^* \tag{5.12}$$

5.3.2 Solution of the problem

Since the problem (5.11) is neither convex nor concave, and the problem involves the $\ell_{2,1}$-norm which is non-smooth, it is difficult to achieve its global optima. However, we can iteratively optimize the parameter W_v, Θ and W_t by dividing the formulation (5.11) into three sub quadratic problems. The solution is described as follows:

Denote $X^{(v)}\Theta W_v - Y = Z_v$, $\tilde{X}^{(t)}W_t - Y = Z_t$, and $X^{(v)}\Theta W_v - \tilde{X}^{(t)}W_t = Z_{vt}$, where $Z_v = [z_v^1, \ldots, z_v^n]^\top$, $Z_t = [z_t^1, \ldots, z_t^n]^\top$, and $Z_{vt} = [z_{vt}^1, \ldots, z_{vt}^n]$. Define diagonal matrices D_v, D_t, and D_{vt} with their diagonal elements $D_v^{ii} = \frac{1}{2\|z_v^i\|_2}$, $D_t^{ii} = \frac{1}{2\|z_t^i\|_2}$ and $D_{vt}^{ii} = \frac{1}{2\|z_{vt}^i\|_2}$ respectively. The optimization problem (5.11) is equivalent to:

$$\begin{aligned}\min_{\Theta,W_v,W_t} & tr[(X^{(v)}\Theta W_v - Y)^\top D_v(X^{(v)}\Theta W_v - Y)]+ \\ & \lambda tr[(\tilde{X}_t W_t - Y)^\top D_t(\tilde{X}_t W_t - Y)]+ \\ & \alpha tr(W_v^\top W_v) + \lambda\beta tr(W_t^\top W_t)+ \\ & \gamma tr[(X^{(v)}\Theta W_v - \tilde{X}_t W_t)^\top D_{vt}(X^{(v)}\Theta W_v - \tilde{X}_t W_t)] \\ s.t. & \Theta^\top\Theta = I \end{aligned} \tag{5.13}$$

In this formulation, $tr(\cdot)$ is the trace operator. Calculating the partial derivative of W_v and W_t and setting them to 0, we can get:

$$W_v = A^{-1}(\Theta^\top X^{(v)\top} D_v Y + \gamma\Theta^\top X^{(v)\top} D_{vt}\tilde{X}^{(t)} W_t) \tag{5.14}$$

$$W_t = B^{-1}(\lambda\tilde{X}^{(t)\top} D_t Y + \gamma\tilde{X}^{(t)\top} D_{vt} X^{(v)}\Theta W_v) \tag{5.15}$$

In the above two equations, $A = \Theta^\top\tilde{X}^{(v)\top} D_v\tilde{X}^{(v)}\Theta + \alpha I + \gamma\Theta^\top X^{(v)\top} D_{vt} X^{(v)}\Theta$, and $B = \lambda\tilde{X}^{(t)\top} D_t\tilde{X}^{(t)} + \lambda\beta I + \lambda\gamma\tilde{X}^{(t)\top} D_{vt}\tilde{X}^{(t)}$, where I is the identity matrix.

The traditional method to solve Θ is to substitute Eq. (5.14) and (5.15) to objective function (5.13), and calculate the eigenvalues and eigenvectors. However, due to the complex data distribution and noise, sometimes it is difficult to keep orthogonal constraints feasible, and the time complexity is very high. To deal with these difficulties, Wen et al. proposed a scalable algorithm which applies a constraint-preserving update with curvilinear search [96].

It can be proved that alternatively updating W_v, W_t, and Θ can ensure the monotonic decrease of objective function (5.11) and make it convergent to a local optima. The TVAR algorithm can be described by Algorithm 1.

We now give a short analysis of the computational complexity of the TVAR algorithm. When achieving the optimal W_t and W_v by calculating the reverse of the matrices for texts and videos, the time complexity is $O(d^3)$. The solution of Θ needs $O(d^3)$. Suppose the number of iteration is T, since r is smaller than d, the overall time complexity is $O(T * d^3)$.

Algorithm 1: Two-view adaptive regression

Input: Training visual feature matrix $X^{(v)}$, auxiliary text feature matrix $\tilde{X}^{(t)}$, label matrix Y, mapping parameter r for $X^{(v)}$, regularization parameters α and β, and balancing parameters λ and γ.

Output: Local optimal solution of W_v, W_t and mapping matrix Θ.

Randomly initialize W_v and W_t;

Randomly initialize Θ and orthogonalize it;

repeat

Compute $Z_v = [z_v^1, \ldots, z_v^n]^\top$, , $Z_t = [z_t^1, \ldots, z_t^n]^\top$, and $Z_{vt} = [z_{vt}^1, \ldots, z_{vt}^n]$;

Set diagonal matrices D_v, D_t, and D_{vt} with their diagonal elements $D_v^{ii} = \frac{1}{2\|z_v^i\|_2}$, $D_t^{ii} = \frac{1}{2\|z_t^i\|_2}$ and $D_{vt}^{ii} = \frac{1}{2\|z_{vt}^i\|_2}$ respectively;

Update W_v using Eq. (5.14);

Update W_t using Eq. (5.15);

Compute Θ using the orthogonality constraint algorithm described in [96];

until Convergence;

Return W_v, W_t and Θ.

5.3.3 Non-linear extensions of TVAR

In our proposed TVAR model, the linear mapping is from both visual and textural cues to label space. It is commonly known that non-linear models can further improve the recognition performance, and so the TVAR model can also be extended to non-linear ones by applying kernel tricks. When there are basis functions $\phi : \mathbb{R}^{d_v} \to \mathcal{H}_v$ and $\varphi : \mathbb{R}^{d_t} \to \mathcal{H}_d$ that can map the visual and textural feature to high-dimensional spaces, the objective function (5.11) can be written as:

$$\begin{aligned} \min_{\phi(\Theta), W_v, W_t} & \|\phi(X^{(v)})\phi(\Theta)W_v - Y\|_{2,1} + \alpha\|W_v\|_F^2 + \\ & \lambda(\|\varphi(\tilde{X}^{(t)})W_t - Y\|_{2,1} + \beta\|W_t\|_F^2) + \\ & \gamma\|\phi(X^{(v)})\phi(\Theta)W_v - \varphi(\tilde{X}^{(t)})W_t\|_{2,1} \\ s.t. & \phi(\Theta)^\top\phi(\Theta) = I \end{aligned} \tag{5.16}$$

The solution of the non-linear TVAR (KTVAR) is similar to the linear one, which only needs to replace the inner-products to kernel functions. Although in many applications the performance of kernelized models outperforms the linear ones, its computational complexity is higher.

5.4 Experiments and analysis

In this section, we explain how we conducted the experiments on two datasets to evaluate the performance of our proposed framework, especially on the enrichment of the partial texts by the event knowledge base and the two-view adaptive regression model.

5.4.1 Datasets and web document collection

UQE50 dataset

Till now there are very few datasets available for context-based video analysis, especially when we need auxiliary texts to describe "large" granular events. Based on this fact we introduce a new video dataset UQE50 (**UQ E**vent database with **50** pre-defined video events) for web video analysis. All videos were downloaded from YouTube. As distinct from other video datasets, UQE50 is dedicated for video analysis from both content and context views. There are two major characteristics for UQE50 dataset as follows:

- The granularity of the events is comparably larger than existing video event datasets. The videos from UQE50 are all from hot events which happened in the last few years, which may contain very complex patterns that are not only activities or action sequences, but also event elements like scenes and other minor visual cues. For instance, some events happened at the same places but lasted a longer time like *APEC Russia 2012*, and some videos for a specific event were shot at the same time but from different camera angles like *William and Kate Wedding 2011*.

- On some occasions it is quite difficult to understand the events from the videos without any prior knowledge, so in the UQE50 dataset the partial textual information for each training

video was collected from the same YouTube web page as the video so as to be able to partly describe the video content and the corresponding event.

- The events defined in the UQE50 dataset are very ad hoc, i.e., some of the them have very few positive instances, so the event recognition task turned out to be very difficult.

All the videos in UQE50 were split into two separate subsets for training (2,122 labelled videos with auxiliary texts, and 1,243 distractors without texts) and testing (1,340 labelled videos and 12,538 distractors). The distractors are completely irrelevant to any events, and in the training video set, the 1,243 distractors were used for cross-validation and model selection.

In video content analysis tasks, motion features such as Dense Trajectories [87, 89] are usually used to sketch dynamic properties. For some tasks such as action recognition, the spatial-temporal information plays a key role in describing the video content. However, these hand-crafted features may not be well suited for the "large" granular video events which contain much more complex visual cues such as scenes and objects. Through our observation, for "large" granular events, the static visual cues of the frames are even more discriminative. Recently, the deep-learned visual features have shown very promising performance in recognition tasks. For example, Xu et al. directly utilized the CNN features extracted from the frames for event detection tasks. Without any motion information, their proposed feature representation even outperforms the state-of-the-art video features [100].

In the video feature extraction phase, we followed Xu et al.'s approach to extract the 5th pooling layer of the pre-trained VGG net from the videos [100], and applied the VLAD aggregation method to represent each video as a 32,768d vector. We also used the vlfeat package[3] to extract the SIFT features from the videos, and processed them in the similar way to the deep feature. Finally, we conducted the PCA on both visual features and concatenated them to represent each video as a 512d vector.

For text features, besides the provided partial texts with the videos, we collected the top 200 to 500 documents for each event by searching the event keywords from the Google search engine and we crawled the web pages. Because the texts from the web pages are quite noisy, we applied

[3] http://www.vlfeat.org/

the following approaches using the Lucene package[4] to improve the data quality: 1) we filtered out all the numbers and the words that start with numbers; 2) we removed the words with more than 15 characters (e.g., urls); and 3) we removed the words that only appear once in the constructed knowledge base systems. By removing the stop words and stemming, each document is represented as a 56,183 dimensional BoW vector. After that we applied Latent Dirichlet Allocation (LDA) [5] model (using the toolbox provided by Mark Steyvers et al. [5]), which is able to smooth the text representation and suppress the influence of noise ([49, 99]), to find 100 topics and then we calculated the posterior possibilities of each word. Finally the dimensionality of text vectors was reduced to 100. For text features, the obtained LDA model can approximately reflect the correct semantic information of events. Even though web documents are influenced by noise, the newly generated topics are basically close to the events, so these posterior probabilities of the topics can replace the original text samples as the new training samples.

EVVE dataset

This dataset [71] was created for video event retrieval, in which 13 events are defined[6]. We used the 620 query videos for training, and the rest for testing. Since all the videos were downloaded from YouTube website, the video IDs are provided in this dataset. We downloaded the videos as well as the partial texts, and extracted the visual and textual features in the same way as described for the UQE50 dataset.

5.4.2 Experiment setup

Using the partial texts and collected web documents, we constructed two types of knowledge base: 1)the knowledge base with a universal ESG that globally discovers the words' correlation by calculating the MI according to Eq.(5.1), and 2)the knowledge base containing a set of ESGs for each single event, 50 for UQE50 and 13 for EVVE respectively. Then the partial texts for the videos were enriched by the knowledge base, as described in section 5.2.

We used the mean average precision (MAP) to measure the performance of the framework.

[4]https://lucene.apache.org/

[5]http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

[6] http://pascal.inrialpes.fr/data/evve/

The regularization parameters α, β and γ in our algorithm were tuned in a grid-search scheme, ranging from 10^{-3} to 10^{3}, and we reported the best results. The mapped dimension r in Θ was set to 200 without any change in the experiments.

5.4.3 Baseline algorithms

We compared the proposed model TVAR with different algorithms, which are briefly introduced as follows:

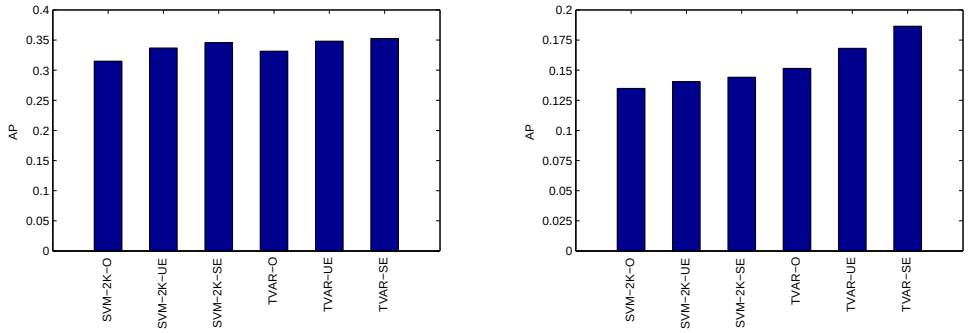
- SVM: the most classical model used in a variety of classification tasks. We adopted the χ^2 kernel on low-level visual features of the datasets without any help from the auxiliary text information.

- SAIR [59]: Semantic Analysis via Intermediate Representation, an effective model for event detection, which may simultaneously learn a intermediate mapping matrix in an unsupervised way, as well as a classifier.

- CCA and KCCA: Canonical Correlation Analysis and its kernel version, the common approaches to discover the correlations between two groups of variables. In our experiment we first conducted CCA and KCCA (with radial basis kernel) on visual and text feature to find the canonical mapping, and then applied the linear SVM on the testing data in the mapped canonical space.

- SVM-2K [20]: an extension of standard SVM which aims to identify the relevant subspaces from two views of the same data, and two classifiers are generated from each view. In this approach we adopted χ^2 kernel and radial basis kernel for visual data and textual data respectively.

- SVM+ [83]: an extension of SVM that could use privileged information in the correcting space to augment the decision space. The privileged features are only available in the training phase. Similar to the setting of SVM-2K, we used χ^2 kernel and radial basis kernel in our experiment.



(a) UQE50 (b) EVVE

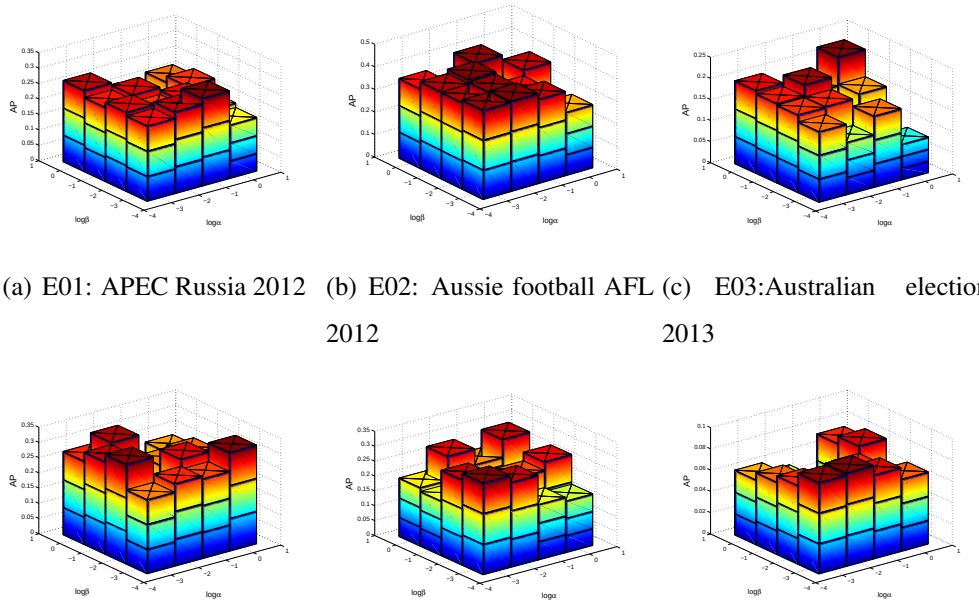
FIGURE 5.4: Effect of semantic enrichment

- TVAR and KTVAR: Two-view adaptive regression and its non-linear version (χ^2 kernel), the algorithms proposed in this work.

In the training phase, both low-level visual features and auxiliary text features were treated as the input and two classifiers (i.e., $f^{(v)}$ for videos and $f^{(t)}$ for texts) which were generated in SVM-2K and TVAR, while no texts were available in testing phase. Thereby only the model $f^{(v)}$ from the visual feature's view was adopted for testing videos in our experiments.

5.4.4 Does semantic enrichment help?

In order to prove the effectiveness of auxiliary text enrichment for our task, we first ran the SVM-2K and TVAR algorithm (balancing parameters are not optimized) on both original auxiliary texts and enriched ones, i.e., we used $X^{(t)}$ and $\tilde{X}^{(t)}$ separately as text input. In addition, we implement the semantic enrichment by both knowledge bases using MI and CMI, introduced in Section 5.2 respectively. Here we name SVM-2K-O and TVAR-O for the models assisted by original auxiliary texts, SVM-2K-UE and TVAR-UE for those enriched by texts with the knowledge base with only a universal ESG, and SVM-2K-SE and TVAR-SE for models enriched by texts with the knowledge base that contains a set of ESGs for each event. The bar charts are plotted in Figure 5.4 for the two datasets.



(a) E01: APEC Russia 2012 (b) E02: Aussie football AFL 2012 (c) E03:Australian election 2013

(d) #1: Presidential victory speech of Barak Obama 2008 (e) #2: Wedding of Prince William and Kate Middleton (f) #3: Arrest of Dominique Strauss-Kahn

FIGURE 5.5: Sensitivity of parameter α and β

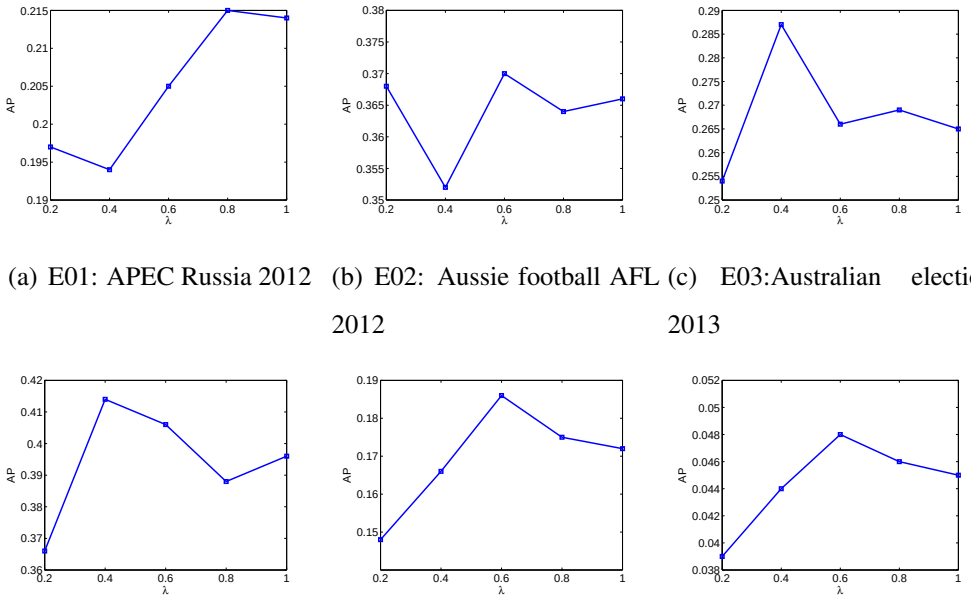
We can see that with the help of the knowledge base constructed by web documents, the enrichment of the text information carries more semantic information, and using these models, it boosts the MAP performance on both the UQE50 and EVVE datasets. Especially, the ESGs constructed for each event can better sketch the semantic contexts to improve event recognition performance.

5.4.5 Sensitivity of parameter α, β, λ and γ

In the TVAR model, the minimization of the two views' difference in Eq. (5.10) plays an important role in knowledge adaptation from text to visual features. For the optimizations of (5.11), we conducted a 5-fold cross-validation on the training set by first fixing the values of λ and γ with 1 and changing the two parameters α and β ranging from 10^{-3} to 10^{0}, then plotted the 3D bar chart for AP as in Figure 5.5.

We could see that the value of regularization terms α and β are set to lower values, i.e., 10^{-3} can ensure the best performance.

Then we set the parameters α, β and γ to 10^{-3}, changed the parameter λ ranging from 0.2 to



(a) E01: APEC Russia 2012 (b) E02: Aussie football AFL 2012 (c) E03:Australian election 2013

(d) #1: Presidential victory speech of Barak Obama 2008 (e) #2: Wedding of Prince William and Kate Middleton (f) #3: Arrest of Dominique Strauss-Kahn

FIGURE 5.6: Sensitivity of parameter λ

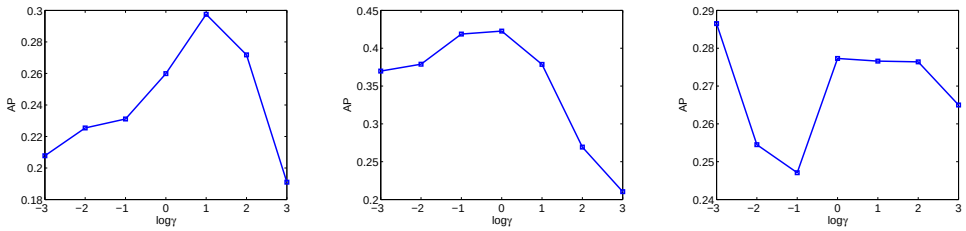
1, and observe the recognition accuracy. The curves of AP are drawn in Figure 5.6 for the two datasets.

At last we fixed the two regularization terms α and β as 10^{-3}, changed the parameter γ in Eq. (5.11) ranging from 10^{-3} to 10^{3}, and observed how this affects the recognition performance. The curves of AP are drawn in Figure 5.7 for the two datasets.

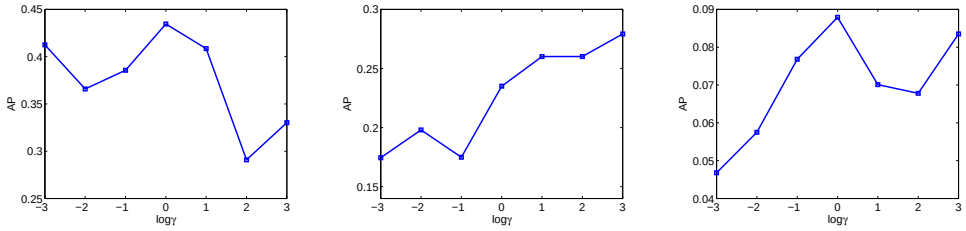
5.4.6 Overall comparison

The MAP of event recognition results is displayed in Table 5.1 for the two datasets (with enriched text features used in CCA, KCCA, SVM-2K, TVAR and KTVAR). Generally speaking, the MAP in UQE50 is higher than EVVE dataset. Based on both of the datasets, we can conclude that the auxiliary texts attached with videos have the ability to enhance the description of the events, and they are eventually beneficial for recognition tasks.

From the table we can see that our proposed model, which obtains the best performance over most of the events, outperforms the others. Generally the TVAR and KTVAR algorithms improve the MAP of the second best competitor on the two datasets. SAIR utilizes a mapping matrix to train



(a) E01: APEC Russia 2012 (b) E02: Aussie football AFL 2012 (c) E03:Australian election 2013



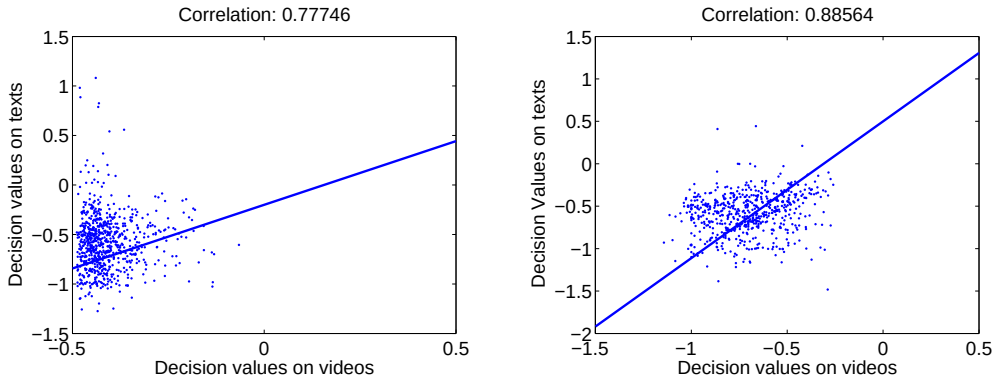
(d) #1: Presidential victory speech of Barak Obama 2008 (e) #2: Wedding of Prince William and Kate Middleton (f) #3: Arrest of Dominique Strauss-Kahn

FIGURE 5.7: Sensitivity of parameter γ

TABLE 5.1: MAP comparison of TVAR to other algorithms

	SVM	**SAIR**	**CCA**	**KCCA**	**SVM-2K**	**SVM+**	**TVAR**	**KTVAR**
UQE50	0.2741	0.2835	0.2926	0.3272	0.3457	0.3390	0.3526	0.3612
EVVE	0.1254	0.1332	0.1563	0.1721	0.1848	0.1855	0.2171	0.2294

a set of intermediate feature representation in an unsupervised way, and it can effectively bridge the information gap between low-level visual features and high-level descriptors, and both SVM and SAIR are direct classification algorithms when there is no auxiliary information. CCA and KCCA try to discover the correlations between low-level visual features and auxiliary text features, but the recognition performance is not so satisfactory in a mapped canonical feature space. The two-view algorithm SVM-2K can not integrate the two heterogeneous features well. From the comparisons we can also observe that the performance improvement of SVM+ is similar to that of SVM-2K. Our proposed TVAR model and its kernelized version are able to integrate the advantages of both SAIR and SVM-2K by simultaneously bridging the information gap and maximizing the consistency of the two views, thus achieving the best recognition performance.



(a) UQE50 (b) EVVE

FIGURE 5.8: The correlation of the two-view data

5.4.7 The correlation of the two views

Similarly to SVM-2K, the TVAR runs on both visual data and textual data in the training phase. Eq. (5.10) shows that the minimization of the two-view difference can achieve knowledge adaptation from the text to visual cues. We calculated the ℓ_2-norm difference between the two models $f^{(v)}$ and $f^{(t)}$ on the training video set. We then made 2D plots as shown in Figure. 5.8, in which each dot represents a pair of decision values, and the line is the linear regression result.

From the graphs we can see that the two-view correlation is high, which reveals the two-view learning strategy is effective in knowledge adaptation.

5.4.8 Do TVAR+ and TVAR- perform well?

Based on the TVAR model, if we simultaneously apply the subspace transformation on both auxiliary text features and visual features, the model becomes "TVAR+", which needs to calculate another mapping matrix in the optimization procedure. Here we use Θ_v and Θ_t to represent the mapping matrices for visual features and text features respectively. The objective function of TVAR+ is:

TABLE 5.2: TVAR,TVAR- and TVAR+

	TVAR	**TVAR-**	**TVAR+**
UQE20	0.3526	0.3251	0.3502
EVVE	0.2171	0.1893	0.2134

$$\begin{aligned}\min_{\Theta_v,\Theta_t,,W_v,W_t} & \left\|X^{(v)}\Theta_v W_v - Y\right\|_{2,1} + \alpha\|W_v\|_F^2 + \\ & \lambda(\left\|\tilde{X}^{(t)}\Theta_t W_t - Y\right\|_{2,1} + \beta\|W_t\|_F^2) + \\ & \gamma\left\|X^{(v)}\Theta_v W_v - \tilde{X}^{(t)}\Theta_t W_t\right\|_{2,1} \\ s.t. & \Theta_v^\top\Theta_v = I \\ & \Theta_t^\top\Theta_t = I \end{aligned} \tag{5.17}$$

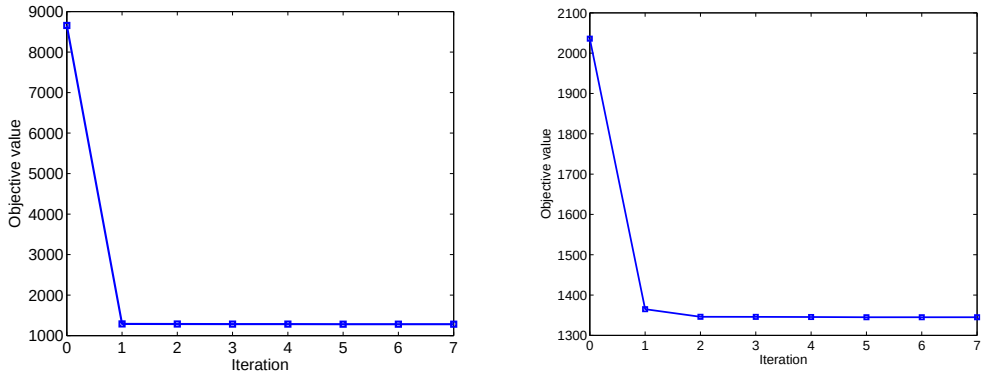
By contrast, if we remove the subspace learning of Θ, the model is similar to SVM-2K, only with different loss functions and the weighting parameter λ. We call the model without the Θ as "TVAR-".

$$\begin{aligned}\min_{W_v,W_t} & \left\|X^{(v)}W_v - Y\right\|_{2,1} + \alpha\|W_v\|_F^2 + \\ & \lambda(\left\|\tilde{X}^{(t)}W_t - Y\right\|_{2,1} + \beta\|W_t\|_F^2) + \\ & \gamma\left\|X^{(v)}W_v - \tilde{X}^{(t)}W_t\right\|_{2,1}\end{aligned} \tag{5.18}$$

We conducted TVAR+ and TVAR- on the two datasets. The AP values are reported in Table 5.2. We can see that neither TVAR+ nor TVAR- have better recognition performance than TVAR, which proves that it is necessary to balance the visual and text features in the same semantic level.

5.4.9 Convergence study

As mentioned in Section 5.3, the objective function (5.11) is optimized iteratively by solving its sub-quadratic problems. In our experiment we also recorded the objective function value for each iteration, and we display the resulting convergence curves in Figure 5.9.



(a) UQE20 (b) EVVE

FIGURE 5.9: Convergence curves of TVAR algorithm

The figure indicates the alternative optimization for the TVAR model converges within 5 iterations, which proves the computational efficiency is very high.

5.4.10 Relations with MED task

Similarly to the multimedia event detection defined by NIST TRECVID, the event recognition task also aims to discover event relevant videos from a large number of unconstrained videos, both of which can be achieved by similar learning approaches. However, there are three major differences between them:

- From the vision's view, human motion features play a significant role in MED tasks , but the importance of this is comparably weaker in our event recognition. Based on this we did not adopt the motion visual features for model construction.

- The intra-class variance in MED video data is very high, which causes semantically similar concepts to have vastly different visual feature representations. In event recognition, although the videos are recorded from different spatial and temporal perspectives, there are several near duplicate frames.

- The events defined in MED are with small granularity, and the semantics these videos contain are simpler than our "large" granular event videos. Thus for the recognition task, the auxiliary texts are of great help to the semantic understanding. However, the assistance of

texts may have very limited effects on MED performance, because only a small number of visual cues have correlations with text information.

5.5 Summary

In this work we proposed a novel framework for web video event recognition, which can make use of video texts and their enrichments to help improve recognition performance. First we built an event knowledge base consisting of several event semantic graphs, which can enrich the textual cues given by partial texts appearing with the videos, then with these enriched textual cues we proposed a two-view adaptive regression algorithm TVAR and its iterative solution approach. The experimental results on two datasets demonstrated that the proposed framework can greatly improve video event recognition performance.

Chapter 6

Scalable video event retrieval by visual state binary embedding

6.1 Overview

In this work we propose a binary embedding model VSBE specifically designed for scalable video event retrieval. To facilitate fast and accurate video event retrieval, we propose a class-specific method to evaluate the event-representativeness of the frames in the videos, so only a limited number of key frames are selected and pair-wise constraints are applied to integrate the semantic information into hash model training. The advantage of the key frame selection is that it can effectively reduce the memory and time consumption, thus it can deal with large-scale video datasets. At the same time, the event retrieval accuracy can be ensured. We conducted the experiment on the challenge TRECVID MED dataset, and have proved the superiority of the VSBE model.

6.2 Visual state binary embedding

6.2.1 Framework

Generally, after pre-processing, a video can be represented as a sequence of key frames. However, some of these are more informative than others, and can thus better reflect the relevant cues of the events at a particular time. In other words, those important key frames stand for the key scenes

of events at different stages. As illustrated in Figure 6.1, four frames from two different events represent four important scenes: *dancing*, *hands with a ring*, *cutting a tomato* and *loaves in a dish*. Intuitively, the first two frames are closely related to the event *wedding ceremony*, while the latter two can partly describe the event *making a sandwich*. These frames exemplify visual states as defined below.

In our work, a *visual state* is a concept which represents the essence of events. Visual states can carry more semantics, and in the entire video collection, the most discriminative information can be well summarized by the visual states to facilitate the training of the binary embedding model.

In this section, we give a detailed explanation of the selection of key frames that contain visual states and then describe the construction of a binary embedding model with novel semantic constraints and optimal training methods. To determine which key frames can be considered as visual states is non-trivial. In addition, to what extent a visual state can describe an event remains an issue, and manually evaluating the importance of the visual states for specified events is unrealistic because: 1) we do not know how many visual states are necessary to describe the events well, nor do we know their importance to the corresponding events; and 2) it is extremely labour-consuming to manually label the frames with the visual states. Thus, we design a group of importance measures to automatically evaluate the event representativeness of the key frames. A sequence of visual states can indicate whether a video is relevant to a specific event. Compared with approaches which try to discover patterns from video motion features, the adoption of visual state sequences has its own advantages: first, the mid-level visual patterns such as object and motion could be well expressed by visual states; second, the distance between the two visual states can better measure semantic similarity instead of visual similarity; last but not least, the sequences of visual states in videos can both locally describe the temporal information and effectively capture the event-relevant moments in videos. Once the visual states are determined, they will be represented by mid-level codes based on the transformation of low-level features. Finally, the query videos are encoded and match the unlabelled videos for scalable event retrieval.

The framework of the proposed model is described in Figure. 6.2. We first decompose the videos into key frames; then we conduct the key frame selection procedure based on visual state evaluation. After that we build the semantic constraints and train the hash models. Finally, each video is represented as a binary matrix for event retrieval.

FIGURE 6.1: An illustration of four visual states representing some aspects for two events *wedding ceremony and* making a sandwich

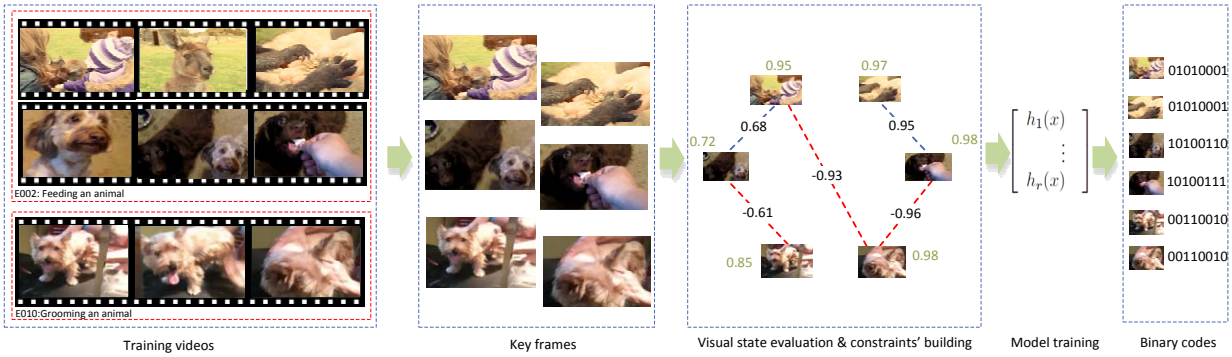


FIGURE 6.2: The framework of the VSBE model.

In the step *visual state evaluation and constraints' building*, we illustrate six key frames that contain several visual states to reflect the essence of two different events: *feeding an animal* and *grooming an animal*. The green font numbers are the representative abilities of the key frames, and the blue and red lines represent the intra-event and inter-event constraints, which are measured by positive and negative numbers respectively.

6.2.2 Visual state evaluation

Suppose the positive training video set for the e-th event is denoted as $\mathbb{X}^e = \{X_1^e, X_2^e, \ldots, X_{C_e}^e\}$, where C_e is the total number of training videos for the e-th event. The q-th training video $X_q^e \subset \mathbb{X}^e$

($q = 1, 2, \ldots, C_e$) is represented as a sequence of N_q^e key frames $X_q^e = [x_q^e(1), x_q^e(2), \ldots, x_q^e(N_q^e)]^\top \in \mathbb{R}^{N_q^e \times d}$, where each row is a d-dimensional visual feature vector of the corresponding frame. The event training videos have the following properties: 1) the visual content in each video is fairly focused and smooth, i.e., the visual variance of key frames in the same video is relatively low; 2) different subsets of consecutive key frames may present different stages of an event, and these stages can reveal the events from different perspectives, for example, a video clip that describes the event *making a sandwich* is reflected by the frames at the beginning, middle, and tail of the video separately; and 3) different key frames may have different representative abilities, where some may contain more event-descriptive information and some contain less.

Based on the above observations, we aim to select the most representative key frames from the training videos to form a condensed training set to build up a binary embedding model, where the efficiency of the training process may be highly improved. We design a group of measures to evaluate the importance of single frames from different levels, including the video-level, the event-level and the global level. For each event training set $\mathbb{X}^e$ and each key frame from any video in $\mathbb{X}^e$, we will calculate the frame importance to $\mathbb{X}^e$. Considering the frame x from $X_q^e \in \mathbb{X}^e$, we explain the three importance measures as explained below.

The importance of a frame at the video-level is measured by its representativeness to the video it belongs to, which is reflected by its visual distance to the dominant visual information carried by the video. Specifically, we use the average of the feature vectors of all key frames contained by the video to represent its dominant visual information. Assume a key frame x from video X_q^e is represented as a feature vector (e.g. SIFT and HOG quantized by VLAD [33]). Let the video-level importance of x be denoted as $VI(x, X_q^e)$, which can be calculated by the following equation:

$$VI(x, X_q^e) = \exp\left(-\frac{\|x - \sum_{i=1}^{N_q^e} x_q^e(i)/N_q^e\|^2}{2}\right) \tag{6.1}$$

The larger the value of VI is, the more representativeness the frame x has for the video X_q^e, so this frame is more likely to sketch the corresponding event.

The event-level importance is designed to measure the relevance of a frame to its belonging event, which is formally defined as:

$$EI(x, \mathbb{X}^e) = \exp\left(-\frac{\|x - \sum_{x' \in \mathbb{X}^e x' \notin X_q^e x' \in \mathcal{N}^e(x)} x'/k\|^2}{2}\right) \tag{6.2}$$

where $\mathcal{N}^e(x)$ is a set of k-nearest neighbours of x from other videos that belong to the same events $\mathbb{X}^e$. If the value of EI is small, it implies that the distance from x to its neighbours is large and x might be an outlier or outstanding frame of the e-th event. A frame with lower EI is less relevant to the e-th event while a greater value of EI stands for a more powerful descriptive ability of the frame for the event.

The global-level importance of a frame is defined as:

$$GI(x, \mathbb{X}^e) = \exp\left(-\frac{\|x - \sum_{x' \notin \mathbb{X}^e, x' \in \mathcal{N}^g(x)} x'/k\|^2}{2}\right) \tag{6.3}$$

where $\mathcal{N}^g(x)$ is a set of k-nearest neighbours of x selected from the training videos belonging to other events. Different from the other two importance measures, the greater value of GI means the key frame x is unspecific to the event, which in turn implies it has weaker descriptive abilities.

After a coarse evaluation of the above three importance measures of a key frame x, its overall importance to $\mathbb{X}^e$ can be calculated as:

$$F(x, \mathbb{X}^e) = \frac{VI(x, X_q^e) \times EI(x, \mathbb{X}^e)}{GI(x, \mathbb{X}^e)} \tag{6.4}$$

The above equation can be explained as follows: the frequently occurring visual patterns in the training video set have stronger representative abilities, while the visual patterns appearing independently in many event-relevant videos are relatively less important for the event description. The potential semantic patterns are contained in the key frames with high importance, so, as mentioned above, we call them **visual states** of events, based on which the training set of the binary embedding is formed. In our experiments, we found that the first 60% of the key frames are sufficient to train reliable hash models.

In Figure.6.2 we give an overall explanation of the frame importance, where two events *feeding an animal* and *grooming an animal* share similar visual patterns (e.g., *dog*) in many frames, but their representative frames are different. For example, the visual patterns *the dog is opening its mouth and eating the food* and *a hand is combing the fur of a dog* represent the unique static

properties of the two events. Such kind of difference could be distinguished and high-lightened in the procedure of visual state evaluation. The frames that show the appropriate patterns (gestures in the two events) indicate they have more powerful descriptive abilities, thus their importance values are high. As a result, the two visual states should be far away from each other in binary embedding. By contrast, in Figure 6.1, the two visual states *cutting a tomato* and *loaves in a dish* are representative to the event *making a sandwich*, thus they should have the similar data representations in binary embedding.

By applying importance measures and introducing the concept of visual state, the proposed VSBE model has two significant advantages: first, event-representative key frames are selected to reduce the size of the training set and meanwhile remove noise; second, the importance of the frames is that they convey prior knowledge, based on which semantic constraints are constructed.

Having evaluated the key frames from the training videos, we can formulate some constraints to facilitate the visual state coding. Assume two frames x, x' are from the same event $\mathbb{X}^e$, our target is to let them have a coupled correlation. Let N be the total number of the selected key frames from all training videos, $ind(x)$ be the index of frame x, and $u_i^+ = [u_{i,1}^+, u_{i,2}^+, \ldots, u_{i,N}^+]^\top$ be the i-th intra-event constraint vector. If two frames x and x' belong to the same event $\mathbb{X}^e$,

$$\begin{aligned} u_{i,ind(x)}^+ &= F(x, \mathbb{X}^e) \times F(x', \mathbb{X}^e) \\ u_{i,ind(x')}^+ &= -F(x, \mathbb{X}^e) \times F(x', \mathbb{X}^e) \end{aligned} \tag{6.5}$$

Similarly, let $u_j^- = [u_{j,1}^-, u_{j,2}^-, \ldots, u_{j,N}^-]^\top$ be the inter-event constraint vector. Both u_i^+ and u_j^- are column vectors. If $x \in \mathbb{X}^e$, $x' \in \mathbb{X}^{e'}$, $\mathbb{X}^e \neq \mathbb{X}^{e'}$ and $x' \in \mathcal{N}(x)$, the values in u_j^- could be set as:

$$\begin{aligned} u_{j,ind(x)}^- &= F(x, \mathbb{X}^e) \times F(x', \mathbb{X}^{e'}) \\ u_{j,ind(x')}^- &= -F(x, \mathbb{X}^e) \times F(x', \mathbb{X}^{e'}) \end{aligned} \tag{6.6}$$

From the above two equations we can see that all the pair-wise constraints are weighted by the importance measures of visual states, which we call the **weighted scheme**. We will demonstrate that the weighted scheme plays a significant role in binary embedding, which outperforms the

approach that equally constructs the pair-wise constraints, which we call the **equal scheme**, i.e., to simply set $u^+_{i,ind(x)} = u^+_{j,ind(x)} = 1$ and $u^+_{i,ind(x')} = u^-_{j,ind(x')} = -1$ in Eq. (6.5) and Eq. (6.6) respectively.

In order to have a compact representation of the constraints, we concatenate the column vectors and form two constraint matrices $U^+ = [u^+_1, u^+_2, \ldots, u^+_{M^+}] \in \mathbb{R}^{N \times M^+}$ and $U^- = [u^-_1, u^-_2, \ldots, u^+_{M^-}] \in \mathbb{R}^{N \times M^-}$, where M^+ and M^- are the numbers of the intra-event and inter-event constraints, respectively.

In the next subsection, we will introduce how to utilize the two constraint matrices to generate the binary codes that potentially incorporate the event-relevant visual states.

6.2.3 A visual state binary embedding model

In this subsection, we introduce a visual state binary embedding (VSBE) model to precisely encode the possible visual states. Without any constraints, spectral hashing[95] is one of the most basic hashing methods, and the objective function is formulated as:

$$\begin{aligned} &\min_Y \frac{1}{2} \sum_{i,j=1}^{n} \|Y_i - Y_j\|^2 A_{ij} = tr(Y^\top L Y) \\ &s.t.\ Y \in \{1, -1\}^{n \times r} \\ &\quad \mathbf{1}^\top Y = 0 \\ &\quad Y^\top Y = I \end{aligned} \tag{6.7}$$

In the above formulation, A is the adjacency matrix, and the value of A_{ij} is calculated as:

$$A_{ij} = \begin{cases} \exp \frac{-\|x_i - x_j\|^2}{\sigma} & x_i \text{ and } x_j \text{ are } k\text{-nearest neighbours} \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

D is the diagonal matrix such that $D\mathbf{1} = A\mathbf{1}$, where $\mathbf{1}$ is a column vector with all elements set as 1. The Ratio Laplacian matrix L is calculated as:

$$L = D - A \tag{6.9}$$

The affinity matrix A is built on the visual level rather than the semantic level. In unconstrained videos, semantically similar descriptors may have vastly different visual representations. Therefore, considering the visual similarity without any constraints is unable to reflect the semantic correlations among different visual states. However, L is still necessary for the VSBE model because through our observation, most of the training videos have low visual variance, and the adjacent matrix A makes the frames in the same video have the similar binary embedding.

Our target is to let the visual states from one event have similar binary codes. At the same time, the binary codes should have the discriminative ability to integrate the visual states of the corresponding events. In order to achieve this, the proposed model should simultaneously be visually consistent and incorporate the semantic constraints.

Considering the matrices U^+ and U^- presented in the previous subsection, it is easy to verify that $U^+U^{+\top}$, $U^-U^{-\top}$ and L have the same properties:

1. They are symmetric matrices, thus all of them are semi-positive definite;

2. $U^+U^{+\top}\mathbf{1} = U^-U^{-\top}\mathbf{1} = L\mathbf{1} = 0$, i.e.,The smallest eigen values of the three matrices are 0, with the corresponding eigen vector $\mathbf{1}$;

Besides L, we impose pair-wise constraints to put the semantically similar visual states into similar buckets. So the objective function is defined as:

$$\min_{Y} tr[Y^\top(L + \lambda(U^{+\top}U^+ - U^{-\top}U^-))Y] \tag{6.10}$$

where λ is an enforcement parameter to apply the pairwise constraints, and Y is the binary matrix. By applying U^+, the model is able to encode similar visual states with large visual variances from the intra-class videos to the similar binary embeddings. Similarly, U^- can offset the negative effect caused by visually similar but semantically different visual states from inter-class videos. Compared with other semi-supervised binary embedding algorithms like semi-supervised circulant binary embedding [105], the VSBE is more flexible to incorporate the "soft" constraints.

Note that in order to keep the matrix $L + \lambda(U^{+\top}U^+ - U^{-\top}U^-)$ semi-definite, the number of intra-event constraints should be equal to or greater than the number of inter-event constraints. In our experiment, for each selected key frame, we randomly choose 5 key frames from the same

event videos to construct U^+, and choose the nearest 5 neighbour frames from the other event-relevant videos to construct U^-.

Given a key frame $x \in \mathbb{R}^{1\times d}$ that contains an event-relevant visual state, we seek to get a set of linear mapping functions $f_l(x) = xw_l + b_l$, $l = 1, 2, \ldots, r$, where $w_l \in \mathbb{R}^{d\times 1}$ is the learned weight vector. So the binary embedding functions can be expressed as $h_l(x) = sign(f_l(x)) = sign(xw_l + b_l)$, $l = 1, 2, \ldots, r$. To have a more compact representation, suppose $z \in \mathbb{H}$ is the binary code of x, $W = [w_1, w_2, \ldots, w_r] \in \mathbb{R}^{d\times r}$ is the weight matrix, and $b = [b_1, b_2, \ldots, b_r]$ is the bias vector. The binary code $z \in \{-1, 1\}^{1\times r}$ of frame x can be expressed as:

$$z = \begin{bmatrix} h_1(x) \\ \vdots \\ h_r(x) \end{bmatrix}^\top = \begin{bmatrix} sign(xw_1 + b_1) \\ \vdots \\ sign(xw_r + b_r) \end{bmatrix}^\top = sign(xW + b) \tag{6.11}$$

We intend to discover the relevance of essential visual states contained in the key frames, and learn the hash functions in a joint optimization framework. Suppose the key frames from the training videos are represented as a matrix $X = [x_1; x_2; \ldots; x_n] \in \mathbb{R}^{n\times d}$, where each row is a d-dimensional vector. The objective function is given by:

$$\begin{aligned} &\min_{Y,W} tr[Y^\top (L + \lambda(U^{+\top}U^+ - U^{-\top}U^-))Y] + \\ &\quad \alpha\|XW - Y\|_2^2 + \beta\|W\|_2^2 \\ &s.t.\ Y \in \{1, -1\}^{n\times r} \\ &\quad Y^\top Y = I \end{aligned} \tag{6.12}$$

The above function has two balance parameters α and β. Note that the problem is equivalent to the graph partitioning, and the constraint $Y \in \{1, -1\}^{n\times r}$ makes it an NP-hard problem. So we follow the method [95] by removing the binary constraint for practical optimization. Hence the optimization problem becomes:

$$\begin{aligned} &\min_{Y,W} tr[Y^\top (L + \lambda(U^{+\top}U^+ - U^{-\top}U^-))Y] + \\ &\quad \alpha\|XW - Y\|_2^2 + \beta\|W\|_2^2 \\ &s.t.\ Y^\top Y = I \end{aligned} \tag{6.13}$$

With the removal of the binary constraint, the optimal Y can be achieved by eigenvalue decomposition in the real number space. In the binary embedding procedure, we can use Eq. (6.11) to keep the centring property of the hash codes.

6.2.4 Solution

To solve the joint optimization problem (6.13), we first set the derivative of it *w.r.t.* W to 0, and then we get:

$$W = (\alpha X^\top X + \beta I)^{-1}\alpha X^\top Y \tag{6.14}$$

Let $P = (\alpha X^\top X + \beta I)^{-1}\alpha X^\top$ so $W = PY$. By substituting W in Eq. (6.13), it becomes:

$$\begin{aligned} &\min_{Y} tr[Y^\top (L + \lambda(U^{+\top}U^+ - U^{-\top}U^-))Y] + \\ &\quad \alpha\|XPY - Y\|_2^2 + \beta\|PY\|_2^2 \\ &s.t.\ Y^\top Y = I \end{aligned} \tag{6.15}$$

The above problem is equivalent to:

$$\begin{aligned} &\min_{Y} tr[Y^\top (L + \lambda(U^{+\top}U^+ - U^{-\top}U^- + \\ &\quad \alpha(XP - I)^\top (XP - I) + P^\top P))Y] \\ &s.t.\ Y^\top Y = I \end{aligned} \tag{6.16}$$

Let $V = L + \lambda(U^{+\top}U^+ - U^{-\top}U^- + \alpha(XP - I)^\top (XP - I) + P^\top P)$. The optimal Y could be calculated by computing the first r eigenvectors corresponding to the r smallest eigenvalues of matrix V.

Algorithm 2: The algorithm for VSBE

Input: The selected key frame feature matrix X, constraint matrices U^+ and U^-, hash bit r, enforcement parameter λ, and balance parameters α and β.

Output: Local optimal hash mapping matrix of W, the bias vector b, and the visual states matrix Y.

Randomly initialize W;

Randomly initialize Y and orthogonalize it;

Compute the affinity matrix according to Eq. (6.8);

Compute matrix L based on Eq. (6.9);

repeat

Update W based on Eq. (6.14);

Compute $V = L + \lambda(U^{+\top}U^+ - U^{-\top}U^- + \alpha(XP - I)^\top(XP - I) + P^\top P)$;

Compute Y by eigen decomposition of V;

until Convergence;

Compute b by calculating the median numbers of each column of Y;

Return W, b and Y.

When we get the optimal Y, the bias vector $b = [b_1, \ldots, b_r]^\top$ can be simply computed by calculating the median numbers of each column of Y. The overall algorithm for VSBE is described in Algorithm 2. Similar to the VHDT algorithm proposed by Ye. et al [104], the VSBE algorithm is solved in an iterative way. However, the time complexity of VSBE is much lower. Given the pre-computed matrix P, the Laplacian matrix L, and the iterations t, the eigen decomposition complexity is $\mathcal{O}(n^3)$, and the computation of W needs $\mathcal{O}(n \times r)$, so the general time complexity for VSBE is $\mathcal{O}(t(n^3 + n \times r))$. Specifically, since we only use the selected frames of the training videos, the training time is even shorter than using all the key frames. For example, if we use the top 60% of selected frames according to section 6.2.2, the optimization procedure only takes about 1/5 time compared with that using all frames.

6.2.5 Video-level distance calculation

In our proposed framework, we first select a set of key frames from the positive training videos, which contain the possible visual states that could sketch different static properties of the pre-defined events, so the sequence of frames for a video that contains event-relevant visual states can also be kept as the discriminative information. With the assumption that the frames within a video have the low variances, we only need to re-construct the training set using a limited number of frames, which could greatly reduce the training time and save memory.

Then we use the proposed VSBE model in subsection 6.2.3 and subsection 6.2.4 to encode all the video key frames, since we do not know the event representative abilities of both query videos and the database videos. As a result, videos are represented by compact binary matrices, which will be stored in the database for query purposes.

Assume a binary query video is represented as a binary matrix $Z^Q = [z_1^Q; z_2^Q; \ldots; z_m^Q] \in \mathbb{H}^{m \times r}$, with each row being an r-bit binary code generated by Eq. (6.11), and the testing video is $Z^T = [z_1^T; z_2^T; \ldots, z_{m'}^T] \in \mathbb{H}^{m' \times r}$. The distance of two frames' binary codes z_a^Q and z_b^T is

$$d(z_a^Q, z_b^T) = \sum_{l=1}^{r} H(z_{al}^Q, z_{bl}^T) \tag{6.17}$$

where $H(\cdot, \cdot)$ is the Hamming distance.

There are three methods to compute the distance between two videos: sliding-window based distance, frame-wise distance and DTW (dynamic time warping) distance.

The sliding-window based distance considers the temporally-aligned information of the video frames. Assume $m < m'$, the distance between two videos Z^Q and Z^T can be calculated by the maximum bit value subtracting the minimum sub-sequence similarity:

$$\begin{aligned} d_{sw}(Z^Q, Z^T) &= \min\{r - \frac{1}{m} \sum_{a=1}^{a+m} d(z_a^Q, z_b^T)\} \\ b &= a - m + 1, \ldots, m' - m + 1 \end{aligned} \tag{6.18}$$

It can be easily seen that the distance is shortest when two videos are both spatially and temporally aligned. However, such alignment between two videos is too rigid, because the visual patterns in different videos are often presented in different temporal scales.

The frame-wise distance between two videos Z^Q and Z^T is calculated as:

$$d_{fw}(Z^Q, Z^T) = \frac{1}{mm'} \sum_{a=1}^{m} \sum_{b=1}^{m'} d(z_a^Q, z_b^T) \tag{6.19}$$

The frame-wise distance totally ignores the temporal information, and it needs the pair-wise frame comparison in the retrieval phase, thus it is slower than the sliding-windows based computation.

DTW is often used to measure similarity between two temporal sequences which may vary in time or speed [1]. In the video retrieval task, it first computes the Hamming distance matrix between two videos Z^Q and Z^T, then applies the dynamic programming to find the shortest path. The calculation of DTW is the slowest among the three distance measures, but it can keep both spatial and temporal information of the videos.

6.3 Experimental Evaluation

In this section, we conduct the binary embedding and event retrieval experiment on the TRECVID MED video collection to evaluate the performance of our proposed method VSBE. Then we give an analysis regarding the experimental results.

6.3.1 Data preparation

The NIST TRECVID provides a series of multimedia event detection (MED) datasets for competition every year. Here we use the following subsets in our experiment: 1)MED11 event kits(E001-E015), 2)MED12 event kits (E021-E030), 3)MED11 DEV-T, 4)MED12 Kindred cityscape collection, 5)MED13 AdHoc event kits (E031-E040), and 6)Progress test collection.

The data setting is described as follows:

- The videos from the positive sets (MED11 event kits, MED12 event kits and MED13 Ad-Hoc event kits) are split into three subsets, with approximately 100, 50, 50 positive videos respectively. For each event class, the 100-video set is used for model training, one of the

[1] https://en.wikipedia.org/wiki/Dynamic_time_warping

50-video set is as query video set, and the rest of the labelled videos are injected into the testing set;

- The MED12 Kindred cityscape collection does not contain any event-relevant videos, i.e., all videos from this subset are negative. These videos are injected into the testing set;
- All the videos in MED11 DEV-T are used for testing;
- The videos in the Progress test collection are used for testing;

We conducted two groups of experiments on the dataset. The first group (G1) is on the subsets 1) to 5) as introduced above, because all videos in these subsets are well labelled. As a result, in the G1 experiment there are 3,500 videos (35 pre-defined events) for hash function training, 1,750 for query, and the rest of the 24,384 videos are for testing.

Then we mixed the Progress test collection 6) into the G1 test set, and conducted a retrieval on the 122,503 testing videos as the second group (G2) using the model learned in G1.

We extracted the key frames from the videos by the toolbox FFmpeg [2], and applied two kinds of visual features (SIFT and pre-trained convolutional neural network [34][3]) from each key frame of the videos. We adopted the VLAD quantization proposed in [33] to encode the SIFT descriptor. Then we concatenated the two visual features as 12,288 dimensional vectors. Finally, we conducted PCA and mapped the 12,288 dimensional vectors to 512 dimensional ones in order to reduce the noise, as well as the computational complexity.

6.3.2 Experiment settings

We used the mean average precision (MAP) to evaluate the retrieval performance. MAP is a single-valued measurement that reflects the ranking of the retrieval on testing data. The higher the MAP value, the more the model is in favour.

In our proposed VSBE model, there are three parameters α, β and λ for tuning, and we conducted a 5-fold cross validation on the training set. We first fixed the value of λ as 1, and tuned α in

[2]https://www.ffmpeg.org/

[3]We use the output of the fc6 layer (fully connected layer) in the AlexNet with the output dimensionality 4,096.

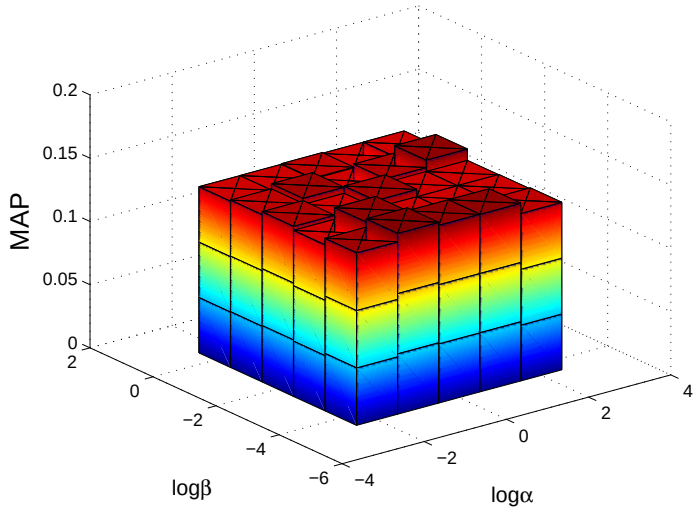


FIGURE 6.3: The sensitivity of α and β in the VSBE model

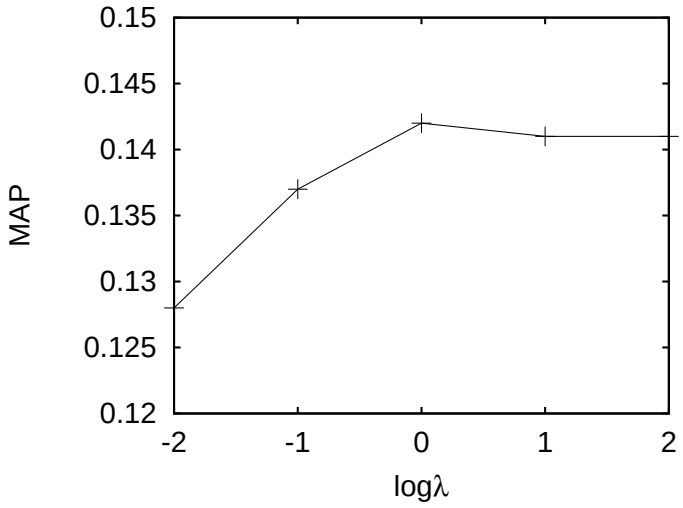


FIGURE 6.4: The sensitivity of λ in the VSBE model

$\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, β in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ in the grid-search approach respectively. The variation of the MAP with respect to these parameters is shown in Figure 6.3. Then we fixed α and β, and tuned λ in range $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. The sensitivity curve of parameter λ is shown in Figure 6.4. Finally, we set $\alpha = 0.1$, $\beta = 0.0001$ and $\lambda = 1$ without any change in the experiment.

To observe the performance affected by different code lengths, we conducted binary encoding with five different bits by setting the value of r to 16, 32, 64, 128 and 256 respectively.

In the G1 experiment we used the MAP to evaluate the retrieval performance of different retrieval models. The MAP is calculated by averaging all AP values for the query videos in each class. We also plotted the recall curve with different binary hash bits, which is used to reflect the retrieval effectiveness. In the G2 experiment, we do not have the whole label set, so for each event class we selected the top ranked 100 videos, manually label them. The retrieval accuracy is the proportion of the true event-relevant samples in the top-ranked videos in retrieval.

Our experiment was conducted on a PC with Intel(R) Core(TM) i7-3770 @3.40GHz CPU with 8 processors, 16GB RAM, and 64-bit Windows 7 Professional operating system.

6.3.3 Effect of key frame selection

In order to achieve the best performance of our proposed VSBE, in subsection 6.2.2 we designed a simple method to evaluate the descriptive abilities of the key frames in training videos, and select a subset of key frames specific to each event. As far as we can see, it is the first attempt to refine the key frames for video event analysis. There are other key frame selection methods proposed for some tasks [84, 52], which need manual labels as supervised information, so they are not suitable in our experiment settings. A comparable selection method is category-specific video summarization proposed in [67], which can assemble the video sequences by performing the SVM to get the scores. Given a set of videos represented as key frame sequences, we first conducted the frame selection using the following methods: 1)random selection (RS) of the key frames; 2)visual similarity based discriminative technique (VSD) [15]; 3)category-specific video summarization (VSS) [67]; 4)selection of the key frames through the importance of video-level (VI) in Eq.(6.1), event-level (EI) in Eq.(6.2) and global-level (GI) in Eq.(6.3) described in Section 6.2.2, respectively; and 5)the integration of the three importance measures (VI-EI-GI) in Eq.(6.4).

We conducted the experiment to compare the performance on the training set when using different proportions of selected key frames, and plotted the MAP curves as shown in Figure 6.6. From the figure it can be easily seen that by simultaneously considering our proposed measurement in Section 6.2.2 and conducting the key frame evaluation in the VSBE model, we only need to

select about 60% of the key frames for hash function training to achieve the best MAP. Therefore, this procedure can significantly reduce the computational burden when dealing with large-scale training video set. When the selection rate is close to 1, all the selection methods are similar.

A case study on the comparison of these frame selection methods is shown in Figure 6.5. From the figure we can see that the quality of the selected key frames directly affect the hash model training. For example, the VSD [15] method can only select the most distinctive frames that describe every temporal aspects of the videos. In this case study, the video is intuitively relevant to the event *parade*, and its visual states can be well described by some frames that contain the pattern *several people are walking together*. At the same time, some noisy parts such as video titles, animals and other irrelevant objects should be excluded in the training procedure. Our proposed visual state evaluation is class specific, which can measure the frame importance at three different levels, so it can effectively filter out the irrelevant parts of the videos.

6.3.4 Comparison of different binary embedding methods

We compare our proposed VSBE with different binary embedding models:

- **Spectral Hashing** (SPH) [95]: One of the most basic unsupervised models for hash function learning, which utilizes the graph Laplacian eigenvectors to generate compact binary codes of data points by assuming the data are uniformly distributed in high-dimensional feature space.

- **Binary Codes with Bilinear Projections** (BPBC) [25]: A bilinear rotation based algorithm for learning binary codes for high dimensional vectors.

- **Circulant Binary Embedding** (CBE) and its semi-supervised extension (SS-CBE) [105]: A binary embedding model created by mapping the data points with a circulant matrix.

- **Video Hashing with both Discriminative commonality and Temporal consistency** (VHDT) [104]: An inductive structural hashing model specially designed for large-scale video retrieval, which can explore both the discriminative local visual commonality and temporal consistency.

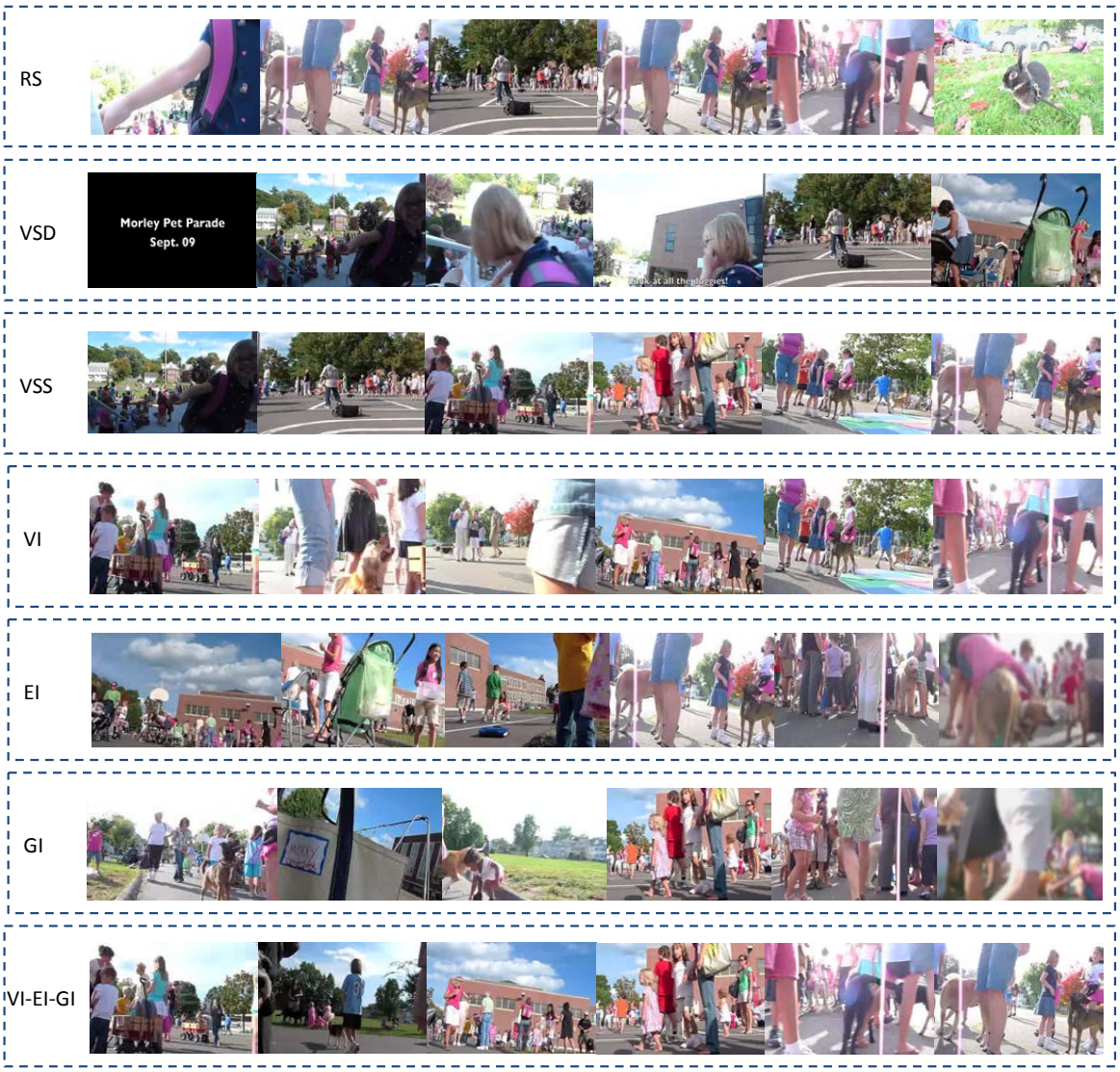


FIGURE 6.5: A visualization of different key frame selection methods

- **Visual State Binary Embedding** (VSBE): Our proposed model in this work. We use both *equal* and *weighted* schemes as mentioned before building the pair-wise constraints, and mark them as VSBE-E and VSBE-W respectively.

Note that the first three models (SPH, BPBC and CBE) are mainly used for near-duplicate image search, but VHDT is specifically designed for unconstrained video event retrieval. In addition, since the above models are used to encode all the key frames in the videos, the time costs in the retrieval phase are all the same. However, the accuracy of retrievals is quite different.

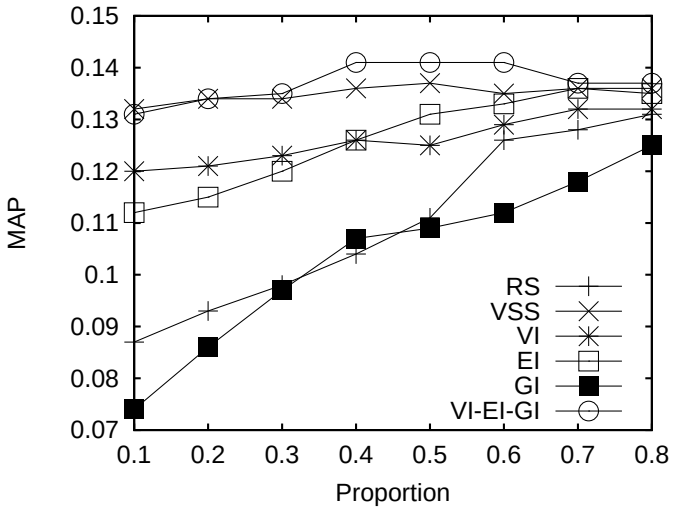


FIGURE 6.6: The effect of different key frame proportions on the training set

6.3.5 Training time and convergence study

We first compared the training time on VHDT and VSBE since both are specifically designed for video event hashing, and we plotted the bar chart in Figure 6.7. The VHDT model does not have any key frame selection procedure as a pre-processing stage, so it needs much more training time. When the number of events and training videos is small, the learning speed of VHDT is acceptable. However, when there is a large number of training videos, it is extremely time and memory consuming to train the hash functions. Since our proposed VSBE model only uses about half of the key frames in model training, and we adopt a different update strategy, the optimization is much faster. In our experiment, there are 3,500 videos in total which belong to 35 pre-defined events, but only tens of minutes were spent in model training. Compared with VHDT, our VSBE model is much more efficient in training hash functions for videos.

As is mentioned in subsection 6.2.4, the objective function (6.13) is optimized by updating W and Y alternatively. The iterative optimization for VSBE converges within 5 iterations, which indicates that the objective function values shrink very quickly.

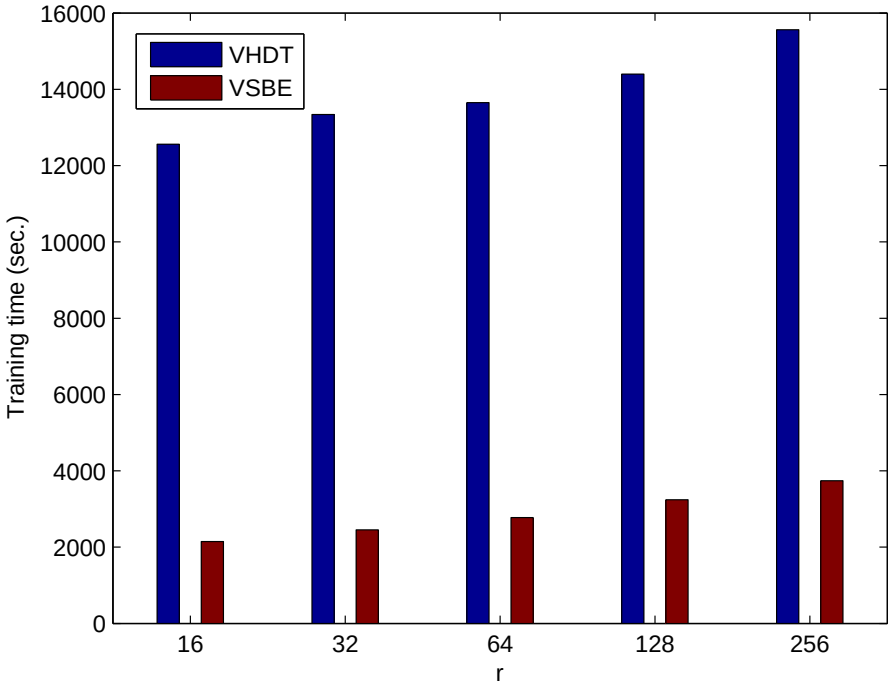


FIGURE 6.7: The comparison of training times between VHDT and VSBE

6.3.6 Event retrieval comparison

We first conducted the retrieval experiment ($r = 128$) on the G1 test set using three different distance calculation approaches introduced in section 6.2.5. In video retrieval tasks, the binary embedding can significantly accelerate the search speed, with the tolerable information loss, so we conducted the video event retrieval in both Euclidean space and Hamming space, to compare the retrieval speed and accuracy. For the retrieval in Euclidean space, we just removed the binary embedding procedure (Eq. 6.11) described in 6.2.3. The MAP values and retrieval times are reported in Table 6.1. We can see that in the binary Hamming space, the video retrieval is much faster than in the real number space. Although the sliding-windows based distance calculation in the retrieval phase is more time efficient, the accuracy is lower than that of frame-wise calculation due to the over rigid temporal alignment. The DTW distance achieves the best accuracy, but its improvement is marginal, and it has a very high computational complexity. This phenomenon indicates that the temporal evidence of the frames has limited contribution to the event video

TABLE 6.1: Retrieval performance using different distance calculation methods ($r = 128$)

	Euclidean space		Hamming space	
	MAP	Time	MAP	Time
Sliding-window	0.0184	42m	0.0155	78.4s
Frame-wise	0.0397	227m	0.0368	243.1s
DTW	0.0411	875m	0.0384	536.3s

retrieval accuracy.

Video event retrieval can also be considered as a specific type of "lazy" classification if we know the pre-defined class of the query video, i.e., the k-nearest neighbour (KNN) algorithm. The neighbours are taken from a set of samples for which the category is known, which can be thought of as the training set for the algorithm, though no explicit training step is required. When the data samples are encoded to binary representations, the computational efficiency is greatly boosted. We compared the VSBE model and the linear SVM, using the same training and testing settings as G1, to see their differences. Note that the video representations in the two models are different. Unlike the binary matrix representation of videos for retrieval, we just used the flat vector representation in the real number space for SVM. The linear SVMs were trained in the one-versus-all way on the 1,750 query videos. The per-class AP values are displayed in Fig.6.8.

From the figure we can see the AP values of VSBE are lower than SVM in general, and the reasons are two-folds: first, the information loss in the binary embedding is inevitable; second, the lazy classifiers usually have worse classification performance than SVMs. Consequently, the binary embedding model can hardly achieve the same accuracy compared to the supervised event detection model. However, such retrieval performance is acceptable because in real application scenarios, we mainly care about the search speed and top-ranked results, rather than the overall retrieval accuracy.

The MAP of event retrieval on the G1 test set is displayed in Fig. 6.9, and the retrieval accuracy on the G2 test set is displayed in Fig. 6.10, respectively. In Fig. 6.9, the error bars are also plotted due to the random constraints and random parameter initialization.

We can see that our proposed VSBE model with weighted scheme (VSBE-W) outperforms the other models. SPH, BPBC and CBE are unsupervised binary embedding methods, and they are

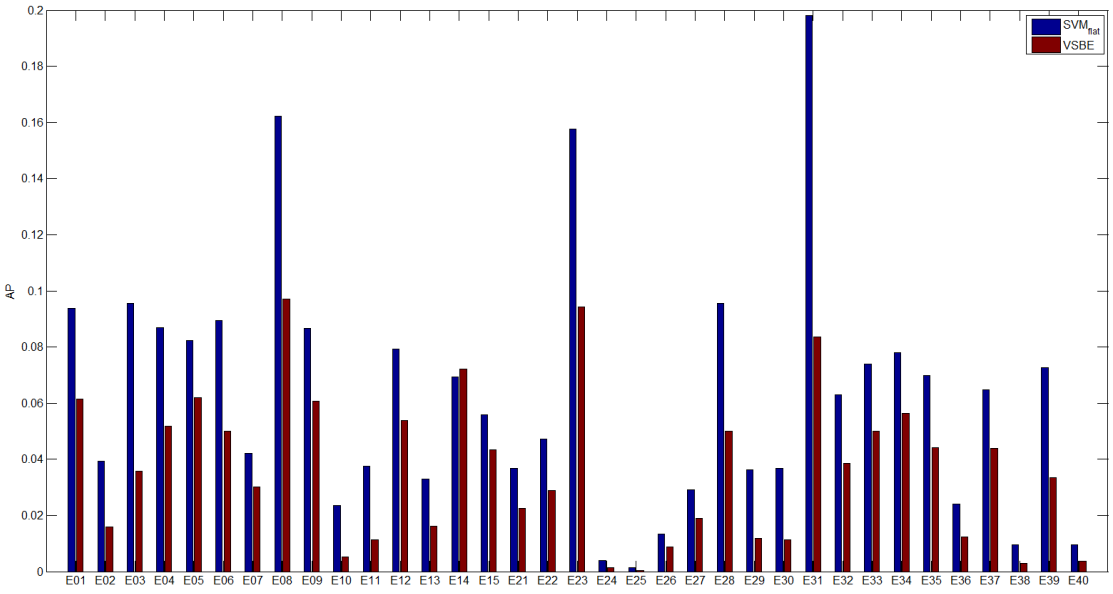


FIGURE 6.8: AP comparisons between retrieval and detection model on G1 test set

not designed for video event retrieval, especially in unconstrained cases, so their event retrieval performances are unsatisfactory. Their semi-supervised extensions can improve the performance to some extent by incorporating the constraints. However, the SS-CBE is not flexible enough to integrate the weighted scheme to train the hash functions, thus it does not achieve the best performance. By adopting the equal scheme, the retrieval performance of VSBE-E is comparable to VHDT, the state-of-the-art model. When adding the weighted scheme introduced in Section 6.2, the retrieval accuracy could be further boosted by VSBE-W, which achieves the best performance. Compared with VHDT, the visual state evaluation and key frame selection in the VSBE model can simultaneously remove noise and reduce computational complexity. To get a better retrieval performance, we also noticed that the hash bits and MAP simultaneously increase when $r \leq 128$, and the retrieval time does not have a dramatic change.

The recall curves of different bits on the test set are plotted in Figure 6.11, which also proves our proposed VSBE model performs best.

In the event retrieval task, the quality of binary embedding can be evaluated by intra-class and inter-class covariance. Similar to clustering models, it is expected that after the mapping procedure, the data points with smaller intra-event covariance and greater inter-event covariance are in favour. With the integration of the semantic information, the mapping function in Eq. (6.11) tries to make

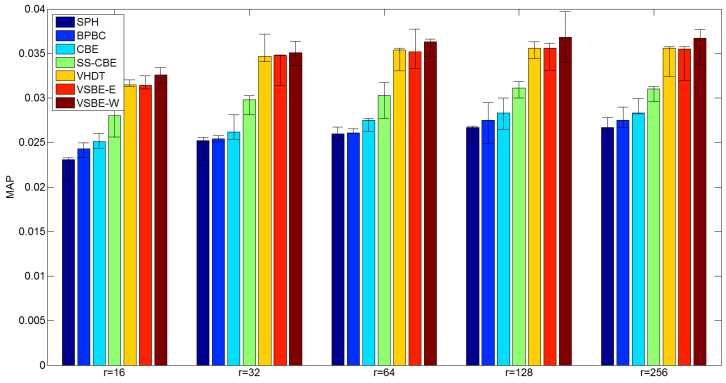


FIGURE 6.9: MAP comparisons of different encoding models on the G1 evaluation

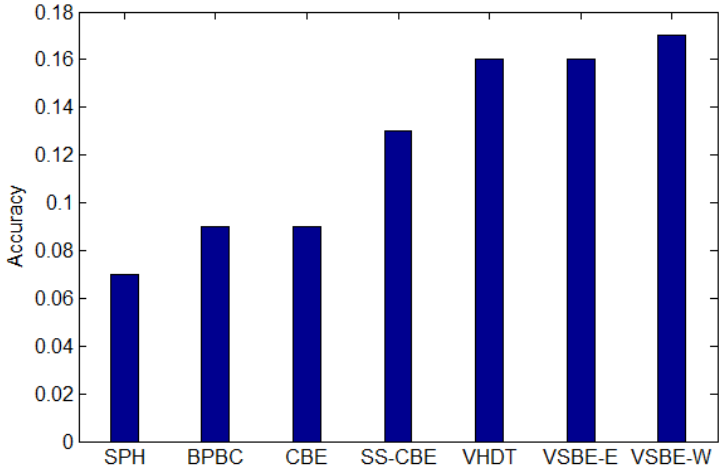


FIGURE 6.10: Top ranked 100 accuracy comparisons of different encoding strategies in the G2 evaluation ($r = 128$)

the visual states in the same event be close with each other in the Hamming space. At the same time, the visual states from different events are supposed to be far away from each other. In our VSBE model, we use Eq. (6.5) and Eq. (6.6) to control the intra-event covariance and inter-event covariance respectively.

To visualize the data mapping of different encoding strategies before binary embedding, we randomly selected several key frames from three different events, and applied different learning models. Then we conducted PCA to reduce the dimension to 2, plotting the scatters in Figure 6.12. From the figure we can see that compared with other mapping strategies, the VSBE with weighted scheme can better discriminate the visual state clusters.

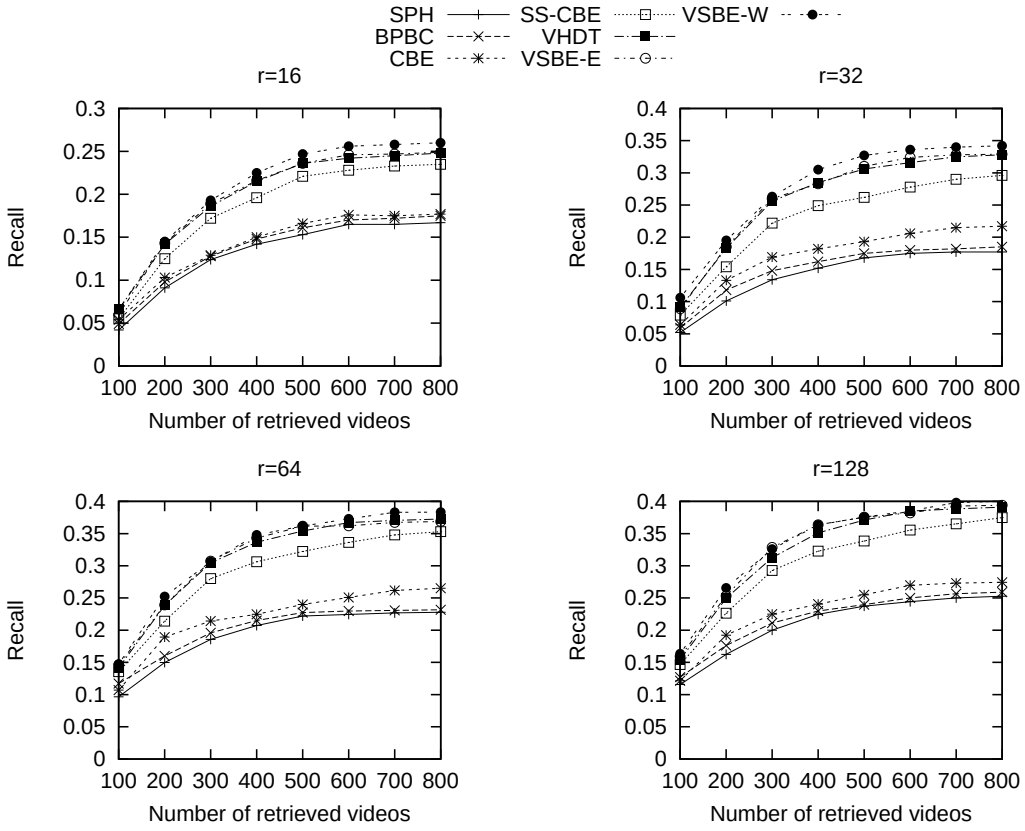


FIGURE 6.11: The recall curves of different hash bits settings

6.3.7 Discussion

From the experiments we can see that the integration of prior knowledge in hash function training can achieve two targets: first of all, in the training phase it can help us select the most representative frames to sketch the visual states, and filter out the redundant and noisy frames, so the computational complexity can be effectively reduced; second, in the retrieval phase, the videos represented by the key frames from the query videos need to be binary encoded to preserve both static and dynamic properties. Although we need the pair-wise Hamming distance calculation, the retrieval efficiency is still very high. From the experimental results, we can see that our VSBE model can both boost the training and ensure the retrieval accuracy. Although it is a semi-supervised method, which contains some uncertainties in label information, the VSBE model performs even better than

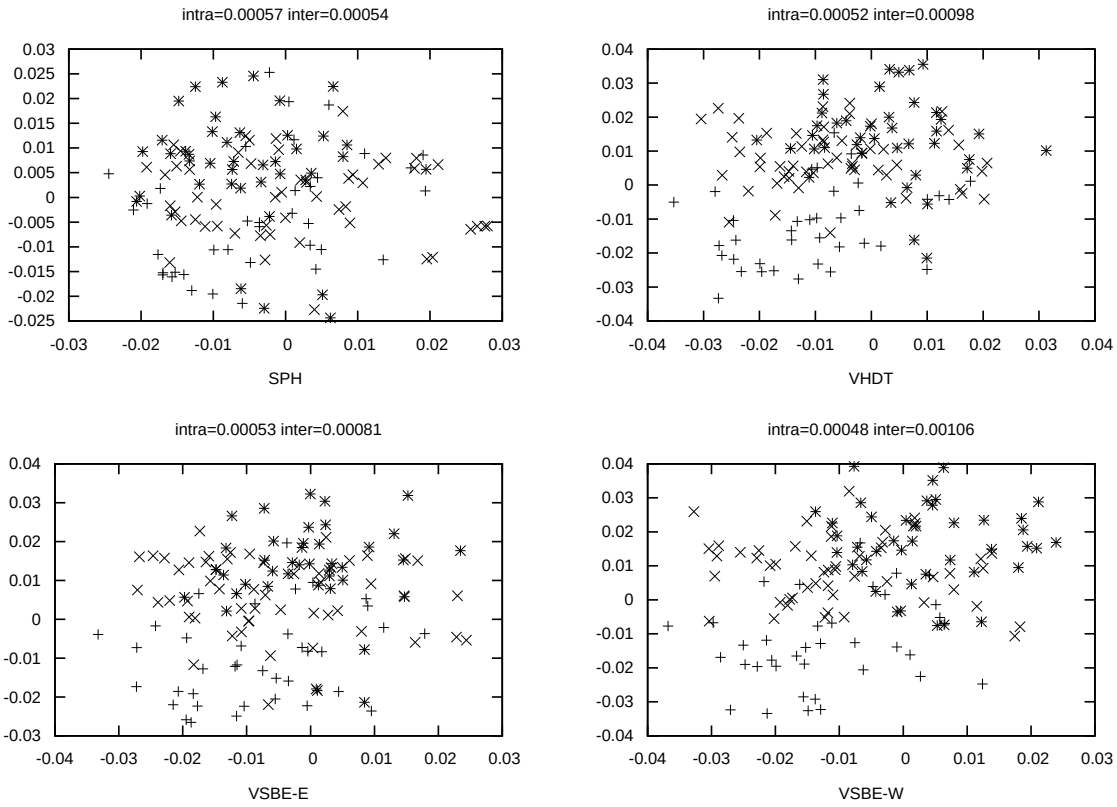


FIGURE 6.12: The 2D visualization of different mappings

the supervised VHDT method.

Through our observation of the videos of the challenging TRECVID MED dataset, we also noticed that both visual and semantic structures are quite complex. Some events are sensible to temporal properties while others are not, so the retrieval performance varies in different event classes. In the future, we plan to build the model by analysing the spatial-temporal properties and evaluating the importance of different features (including different visual and audio features), to further boost the retrieval performance at the event level.

We also noticed that although our proposed VSBE model simultaneously considers the static and dynamic properties of the videos, the information loss of the video representation is still severe after binary embedding, especially when there are a large number of null videos (i.e., irrelevant to any pre-defined events) in the testing set. Another issue is that the VSBE model is inflexible when there are new event categories, i.e., the model cannot be incrementally trained.

6.4 Summary

In this chapter, we proposed a novel binary embedding for scalable event retrieval in large unconstrained video databases. First, we evaluated the representative ability of the key frames from the event-relevant videos, and select the top ranked frames to sketch visual states. Then we constructed the pair-wise constraints as prior knowledge to embed the visual states into binary codes at the semantic level. Finally, we proposed the VSBE algorithm and its iterative solution. The experimental results on the challenging TRECVID MED dataset demonstrated that our proposed VSBE model can both accelerate the training procedure, and boost retrieval accuracy.

Chapter 7

Conclusion and future work

7.1 Conclusion

In this thesis, we first reviewed the relevant techniques for effective video event analysis, including the visual features, deep learning techniques, and the recently proposed machine learning based models for event detection, recognition and retrieval.

We mainly focused on how to build effective models for video event analysis in different cases:

- In Chapter 3, we proposed a generic model MMA for complex video pattern recognition. This model is based on semi-supervised learning and transfer learning, which can learn a mapping matrix Θ to transform video data from the original feature space to a lower-dimensional space. This transformation can simultaneously minimize the data distribution between labelled training videos and unlabelled auxiliary videos, and maximize the margin between different categories. We conducted the experiment on two public datasets, and proved the superiority of the proposed MMA model.

- In Chapter 4, we proposed a two-step deep learning model for MED tasks. In the first training phase, we used the GRU model to train a recurrent auto-encoder for video re-construction. This is an unsupervised learning procedure, so we could use both labelled and unlabelled training videos to learn the temporal patterns as prior knowledge. In the second phase, we only used the labelled training videos to fine-tune the parameters of the model. The

experiment on the challenging TRECVID MED 14 dataset proved that our proposed spatial-temporal deep model can outperform the state-of-the-art MED method.

- In contrast to the previous two chapters, where we built the content based models, in Chapter 5, we proposed a context based framework for web video event recognition. In this framework, we first collected the pages to construct a event knowledge base, which consists of a set of event semantic graphs (ESGs). Then we utilized this event knowledge base to enrich the partial textual information accompanying the web videos. Finally we proposed a two-view adaptive regression model (TVAR) that can utilize the enriched semantic information to help training of the recognition model. We evaluated our approach on a public dataset EVVE and another dataset UQE50, which is a large-scale one collected by ourselves. The experiment showed that our proposed framework can effectively correlate the semantic information and the visual features, thus achieving very promising results.

- In Chapter 6, we proposed a binary embedding method for video event retrieval. For this we first designed a class-specific frame evaluation method, based on the importance measures at video-level, event-level and global-level respectively. By evaluating the importance, we only selected a limited number of frames for further processing. After that we proposed a visual state binary embedding model (VSBE) that can integrate the semantic information into the hash model training procedure. One advantage of our proposed approach is that it can effectively alleviate the negative effect caused by noisy and redundant video frames. At the same time, it can effectively reduce the computational complexity, so it is suitable for the hash model training with large-scale video datasets. The experiment was conducted on the unconstrained video dataset TRECVID MED, and proved the effectiveness of our proposed method.

7.2 Future work

In the future, we plan to continue to explore our research work on video event analysis along the following directions:

- We plan to apply the event detection and recognition techniques on some practical applications such as monitoring indoor activities for public health services. Such tasks require us to quickly analyse the real-time video data. Most of the existing models have very high computational complexities, so it is difficult to directly use them in real-world applications. Besides optimization of the software and hardware systems, one research direction is aimed at establishing how to build efficient models to efficiently process the video data to meet the requirements of real-time circumstances.

- We also plan to integrate video event analysis with big data techniques. In the big data era, we can collect multimedia datasets from different data sources. These datasets are highly unstructured and heterogeneous, but we can try to discover any potential correlations from the patterns. As distinct from traditional video analysis approaches that are mainly based on supervised learning, the analysis of ubiquitous data is mainly unsupervised. As such a kind of knowledge discovery is consistent with human cognition, we plan to design novel models that can use big data techniques to effectively analyse large-scale videos.

- The technique of deep learning is becoming increasingly popular in multimedia content analysis, and we plan to integrate some of the latest deep models for complex video event analysis. For example, the CNN feature and its variants have achieved very promising performance on image data. However, the deep-learned features do not perform well in 3D video data. One of our future tasks is to investigate and build deep models to better sketch the spatial-temporal properties of videos for effective event analysis.

- Finally, we plan to integrate the deep learning model into hash function training for scalable video event retrieval. Recently, deep learning techniques have revealed a very powerful descriptive capability in multimedia content analysis. We noticed that in the traditional binary embedding procedure, there is a serious information loss when integrating the semantics. Applying the deep models in hash model training can effectively make the binary encoding approximate the original data distribution.

References

[1] Trecvid med 14. http://www.nist.gov/itl/iad/mig/med14.cfm, 2014.

[2] A. Abdel-Hakim and A. Farag. Csift: A sift descriptor with color invariant characteristics. In *IEEE CVPR*, volume 2, pages 1978–1983, 2006.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *ACM SIGIR*, pages 37–45, 1998.

[4] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. on Neural Networks*, 5(2):157–166, 1994.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[6] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. In *NIPS*, pages 129–136, 2007.

[7] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schlkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):49–57, 2006.

[8] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel. Efficient co-regularised least squares regression. In *ICML*, pages 137–144, 2006.

[9] X. Chang, Y.-L. Yu, Y. Yang, and A. G. Hauptmann. Searching persuasively: Joint event detection and evidence recounting with limited supervision. In *ACM MM*, pages 581–590, 2015.

[10] L. Chen, L. Duan, and D. Xu. Event recognition in videos by learning from heterogeneous web sources. In *IEEE CVPR*, pages 2666–2673, 2013.

[11] M.-y. Chen and A. Hauptmann. Mosift: Recognizing human actions in surveillance videos. 2009.

[12] Y. Cheng, Q. Fan, S. Pankanti, and A. Choudhary. Temporal sequence modeling for video event detection. In *IEEE CVPR*, pages 2235–2242, 2014.

[13] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv:1409.1259*, 2014.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555*, 2014.

[15] M. Cooper and J. Foote. Discriminative techniques for keyframe selection. In *IEEE ICME*, pages 4–7, 2005.

[16] C. Cortes, M. Mohri, and A. Rostamizadeh. L2 regularization for learning kernels. In *UAI*, pages 109–116, 2009.

[17] G. Doran and S. Ray. A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, 97(1):79–102, 2013.

[18] L. Duan, I. Tsang, and D. Xu. Domain transfer multiple kernel learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[19] M. Faraki, M. T. Harandi, and F. Porikli. More about vlad: A leap from euclidean to riemannian manifolds. In *IEEE CVPR*, 2015.

[20] J. D. Farquhar, D. R. Hardoon, H. Meng, J. Shawe-Taylor, and S. Szedmak. Two view learning: Svm-2k, theory and practice. In *NIPS*, 2005.

[21] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *IEEE CVPR*, pages 2568–2577, 2015.

[22] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization for approximate nearest neighbor search. In *IEEE CVPR*, pages 2946–2953, 2013.

[23] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *IEEE ICCV*, pages 221–228, 2009.

[24] N. Gkalelis and V. Mezaris. Video event detection using generalized subclass discriminant analysis and linear support vector machines. In *ACM ICMR*, pages 25–32, 2014.

[25] Y. Gong, S. Kumar, H. A. Rowley, and S. Lazebnik. Learning binary codes for high-dimensional data using bilinear projections. In *IEEE CVPR*, pages 484–491, 2013.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[27] Y. Guo. Convex subspace representation learning from multi-view data. In *AAAI*, 2013.

[28] A. Habibian, T. Mensink, and C. G. Snoek. Videostory: A new multimedia embedding for few-example recognition and translation of events. In *ACM MM*, pages 17–26, 2014.

[29] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[30] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] Y. Hu, Z. Jin, H. Ren, D. Cai, and X. He. Iterative multi-view hashing for cross media indexing. In *ACM MM*, pages 527–536, 2014.

[32] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[33] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *IEEE CVPR*, pages 3304–3311, 2010.

[34] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *aarXiv:1408.5093*, 2014.

[35] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *International Journal of Multimedia Information Retrieval*, 2(2):73–101, 2013.

[36] 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 *ACM ICMR*, pages 1–8, 2011.

[37] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.

[38] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, pages 2342–2350, 2015.

[39] Y. Kalantidis and Y. Avrithis. Locally optimized product quantization for approximate nearest neighbor search. In *IEEE CVPR*, pages 2329–2336, 2014.

[40] V. Kantorov and I. Laptev. Efficient feature extraction, encoding, and classification for action recognition. In *IEEE CVPR*, pages 2593–2600, 2014.

[41] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE CVPR*, pages 1725–1732, 2014.

[42] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *IEEE CVPR*, volume 2, 2004.

[43] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, page 3, 2013.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[45] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *IEEE ICCV*, 2011.

[46] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *IEEE CVPR*, pages 2251–2258, 2014.

[47] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[48] X. Li, C. Shen, A. Dick, and A. van den Hengel. Learning compact binary codes for visual tracking. In *IEEE CVPR*, pages 2419–2426, 2013.

[49] D. X. Liu, W. R. Xu, and J. N. Hu. A feature-enhanced smoothing method for lda model applied to text classification. In *NLP-KE*, pages 1–7, 2009.

[50] J. Liu, Z. Huang, H. Cai, H. T. Shen, C.-W. Ngo, and W. Wang. Near-duplicate video retrieval: Current research and future trends. *ACM Comput. Surv.*, 45(4), 2013.

[51] J. Liu, Y. Yang, Z. Huang, Y. Yang, and H. T. Shen. On the influence propagation of web videos. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 2014.

[52] L. Liu, L. Shao, and P. Rockett. Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognition*, 46(7):1810–1818, 2013.

[53] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *IEEE CVPR*, pages 2074–2081, 2012.

[54] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE ICCV*, volume 2, pages 1150–1157, 1999.

[55] Z. Lu and Y. Peng. Unified constraint propagation on multi-view data. In *AAAI*, 2013.

[56] Y. Luo, D. Tao, C. Xu, D. Li, and C. Xu. Vector-valued multi-view semi-supervised learning for multi-label image classification. In *AAAI*, 2013.

[57] C. Ma and C. Liu. Two dimensional hashing for visual tracking. *Computer Vision and Image Understanding*, 135:83–94, 2015.

[58] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. G. Hauptmann. Knowledge adaptation for ad hoc multimedia event detection with few exemplars. In *ACM MM*, pages 469–478, 2012.

[59] Z. Ma, Y. Yang, N. Sebe, K. Zheng, and A. Hauptmann. Multimedia event detection using a classifier-specific intermediate representation. *IEEE Trans. on Multimedia (TMM)*, 15(7):1628–1637, Nov 2013.

[60] Z. Ma, Y. Yang, Z. Xu, S. Yan, N. Sebe, and A. Hauptmann. Complex event detection via multi-source video attributes. In *IEEE CVPR*, pages 2627–2633, 2013.

[61] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller. A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks. pages 1996–2000, 2015.

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

[63] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, and R. Prasad. Multimodal feature fusion for robust event detection in web videos. In *IEEE CVPR*, pages 1298–1305, 2012.

[64] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. *NIPS*, 23:1813–1821, 2010.

[65] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE CVPR*, pages 1–8, 2007.

[66] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[67] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, pages 540–555. 2014.

[68] Z. Qi, M. Yang, Z. M. Zhang, and Z. Zhang. Multi-view learning from imperfect tagging. In *ACM MM*, pages 479–488, 2012.

[69] V. Ramanathan, P. Liang, and L. Fei-Fei. Video event understanding using natural language descriptions. In *IEEE ICCV*, 2013.

[70] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei. Learning temporal embeddings for complex video analysis. *arXiv:1505.00315*, 2015.

[71] J. Revaud, M. Douze, C. Schmid, and H. Jégou. Event retrieval in large video collections with circulant temporal encoding. In *IEEE CVPR*, pages 2459–2466, 2013.

[72] F. Shang, L. Jiao, and F. Wang. Semi-supervised learning with mixed knowledge information. In *ACM SIGKDD*, pages 732–740, 2012.

[73] F. Shen, W. Liu, S. Zhang, Y. Yang, and H. Tao Shen. Learning binary codes for maximum inner product search. In *IEEE ICCV*, 2015.

[74] F. Shen, C. Shen, W. Liu, and H. Tao Shen. Supervised discrete hashing. In *IEEE CVPR*, 2015.

[75] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[76] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *ACM SIGMOD*, pages 785–796, 2013.

[77] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[78] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *IEEE CVPR*, pages 1250–1257, 2012.

[79] M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418, 2011.

[80] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. pages 4489–4497, 2015.

[81] R. Trichet, R. Nevatia, and B. Burns. Video event classification with temporal partitioning. In *IEEE AVSS*, pages 1–6, 2015.

[82] M. Van Erp, L. Vuurpijl, and L. Schomaker. An overview and comparison of voting methods for pattern recognition. In *Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 195–200, 2002.

[83] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(56):544 – 557, 2009.

[84] S. Vijayanarasimhan and K. Grauman. Active frame selection for label propagation in videos. In *ECCV*, volume 7576, pages 496–509, 2012.

[85] K.-W. Wan, W.-Y. Yau, and S. Roy. Metadata enrichment for news video retrieval: a graph-based propagation approach. In *ACM MM*, pages 373–376, 2013.

[86] F. Wang, Z. Sun, Y.-G. Jiang, and C.-W. Ngo. Video event detection using motion relativity and feature selection. *IEEE Trans. on Multimedia*, 16(5):1303–1315, 2014.

[87] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *IEEE CVPR*, pages 3169–3176, 2011.

[88] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. In *ICML*, pages 352–360, 2013.

[89] H. Wang and C. Schmid. Action recognition with improved trajectories. In *IEEE ICCV*, pages 3551–3558, 2013.

[90] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large-scale search. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(12):2393–2406, 2012.

[91] J. Wang, W. Liu, S. Kumar, and S.-F. Chang. Learning to hash for indexing big data–a survey. *Proc. of IEEE*, 104(1):34–57, 2016.

[92] J. Wang, H. T. Shen, J. Song, and J. Ji. Hashing for similarity search: A survey. *arXiv:1408.2927*, 2014.

[93] J. Wang, J. Wang, N. Yu, and S. Li. Order preserving hashing for approximate nearest neighbor search. In *ACM MM*, pages 133–142, 2013.

[94] S. Wang, Z. Ma, Y. Yang, X. Li, C. Pang, and A. Hauptmann. Semi-supervised multiple feature analysis for action recognition. *IEEE Trans. on Multimedia*, 16(2):289–298, 2014.

[95] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2009.

[96] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[97] C. Wu, J. Zhu, D. Cai, C. Chen, and J. Bu. Semi-supervised nonlinear hashing using bootstrap sequential projection learning. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 25(6):1380–1393, 2013.

[98] X. Wu, H. Wang, C. Liu, and Y. Jia. Cross-view action recognition over heterogeneous feature spaces. In *IEEE ICCV*, 2013.

[99] L. Xu. Data smoothing regularization, multi-sets-learning, and problem solving strategies. *Neural Networks*, 16(5):817–825, 2003.

[100] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *IEEE CVPR*, pages 1798–1807, 2015.

[101] Z. Xu, Y. Yang, I. Tsang, N. Sebe, and A. G. Hauptmann. Feature weighting via optimal thresholding for video analysis. In *IEEE ICCV*, 2013.

[102] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. *arXiv:1510.01442*, 2015.

[103] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe, and A. Hauptmann. Multi-feature fusion via hierarchical regression for multimedia analysis. *IEEE Trans. on Multimedia (TMM)*, 15(3):572–581, 2013.

[104] G. Ye, D. Liu, J. Wang, and S.-F. Chang. Large-scale video hashing via structure learning. In *IEEE ICCV*, pages 2272–2279, 2013.

[105] F. X. Yu, S. Kumar, Y. Gong, and S.-F. Chang. Circulant binary embedding. In *ICML*, 2014.

[106] S.-I. Yu, L. Jiang, Z. Mao, X. Chang, X. Du, C. Gan, Z. Lan, Z. Xu, X. Li, Y. Cai, et al. Informedia@ trecvid 2014 med and mer. In *NIST TRECVID Video Retrieval Evaluation Workshop*, 2014.

[107] M. D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv:1212.5701*, 2012.

[108] D. Zhang, J. He, and R. Lawrence. Mi2ls: Multi-instance learning from multiple information sources. In *ACM SIGKDD*, pages 149–157, 2013.

[109] D. Zhang, J. He, Y. Liu, L. Si, and R. Lawrence. Multi-view transfer learning with a large margin approach. In *ACM SIGKDD*, pages 1208–1216, 2011.

[110] J. Zhang and J. Huan. Inductive multi-task learning with multiple view data. In *ACM SIGKDD*, pages 543–551, 2012.

[111] J. Zhu, K. Wang, Y. Wu, Z. Hu, and H. Wang. Mining user-aware rare sequential topic patterns in document streams. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 28(7):1790–1804, 2016.

[112] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao. Linear cross-modal hashing for efficient multimedia search. In *ACM MM*, pages 143–152, 2013.