LEVERAGING LARGE SCALE DATA FOR VIDEO RETRIEVAL

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Anıl Armağan

August, 2014

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Pınar Duygulu Şahin (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Öznur Taştan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Sinan Kalkan

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

LEVERAGING LARGE SCALE DATA FOR VIDEO RETRIEVAL

Anıl Armağan
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. Pınar Duygulu Şahin
August, 2014

The large amount of video data shared on the web resulted in increased interest on retrieving videos using usual cues, since textual cues alone are not sufficient for satisfactory results. We address the problem of leveraging large scale image and video data for capturing important characteristics in videos. We focus on three different problems, namely finding common patterns in unusual videos, large scale multimedia event detection, and semantic indexing of videos.

Unusual events are important as being possible indicators of undesired consequences. Discovery of unusual events in videos is generally attacked as a problem of finding usual patterns. With this challenging problem at hand, we propose a novel descriptor to encode the rapid motions in videos utilizing densely extracted trajectories. The proposed descriptor, trajectory snippet histograms, is used to distinguish unusual videos from usual videos, and further exploited to discover snapshots in which unusualness happen.

Next, we attack the Multimedia Event Detection (MED) task. We approach this problem as representing the videos in the form of prototypes, that correspond to models each describing a different visual characteristic of a video shot. Finally, we approach the Semantic Indexing (SIN) problem, and collect web images to train models for each concept.

Keywords: Large Scale Video Retrieval, Multimedia Event Detection, Unusual Videos, Semantic Indexing.

ÖZET

BÜYÜK ÖLÇEKLİ VERİLERİN VİDEO ERİŞİMİNDE KULLANIMI

Anıl Armağan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Asst. Prof. Dr. Pınar Duygulu Şahin
Ağustos, 2014

Günümüzde kullanımı büyük oranda artan video verileri araştırmacıları bu verilerden elde edilebilecek ipuçlarını kullanmaya yöneltmiştir. Çünkü yazısal ipuçlarının günümüzde görsel ipuçları kadar başarılı sonuçlar veremediği gözlemlenmiştir. Bu soruna büyük ölçekli resim ve video verilerini çıkarımız için kullanarak, videolardaki önemli karakteristik bilgileri bularak yaklaşıyoruz. Bu tezde üç farklı konuya odaklanılmaktadır. Bunları olağan dışı olaylardaki ortak motifleri bulmak, geniş ölçekli multimedya olay tespit edilmesi ve videoların anlamsal dizinlenmesi olarak isimlendirebiliriz.

İstenmeyen olayların gerçekleşmesinin bildiricisi olduğu için, olağan dışı olayların erken tespit edilmesi gerekli görülmektedir. Bu konuya genellikle sıradan olayların motiflerinin bulunması ile yaklaşılmaktadır. Elimizdeki bu zorlu problemi çözümlemek için videolardaki hızlı hareketleri yakalayabilen orijinal bir tanımlayıcıyı, piksel yörüngelerinden yoğun aralıklar ile çıkartılarak sunulmaktadır. Sunulan tanımlayıcı, yörünge parça seleleri, olağan dışı videoları sıradan videolardan ayırt etmek için kullanılmaktadır. Daha sonra olağan dışı olarak belirlenen videoların fotoğraf kareleri ile gösterimi için diğer bir yöntem kullanılmaktadır.

Daha sonra TRECVID video erişim değerlendirmesinin bir parçası olan Multimedya Olay Tespiti olarak adlandırılan problemi ele almaktayız. Bu probleme videoları prototipler ile temsil ederek yaklaşmaktayız. Prototipler olayların farklı görsel karakteristik özelliklerini temsil eden modellerdir. Son olarak, TRECVID'in bir diğer parçası olan Anlamsal Dizinleme problemine, İnternet'ten topladığımız resimleri kavramları modellemek için kullanarak yaklaşmaktayız.

Anahtar sözcükler: Geniş Ölçekli Video Erişimi, Multimedya Olay Tespiti, Sıradışı Videolar, Anlamsal Dizinleme.

Acknowledgement

First and foremost, I owe my deepest gratitude to my supervisor, Asst. Prof. Dr. Pınar Duygulu Şahin, for her encouragement, motivation, guidance and support throughout my studies.

Special thanks to Asst. Prof. Dr. Öznur Taştan and Asst. Prof. Dr. Sinan Kalkan for kindly accepting to be in my committee. I owe them my appreciation for their support and helpful suggestions.

I would like to thank to my parents, my father Mümtaz, my mother Faize and my brother Burak Armağan for always being cheerful and supportive. None of this would have been possible without their love. I am tremendously grateful for all the selflessness and the sacrifices you have made on my behalf.

I consider myself to be very lucky to have the most valuable friends Fuat, Arif, Oğuz, Fatih, İrem and Didem. I would also like to thank to my special office mates Fadime, Caner, Eren, Ahmet and Ilker for sharing their knowledge and supporting me all the time.

This thesis is partially supported by TUBITAK project with grant no 112E174 and CHIST-ERA MUCKE project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Indexing and video retrieval have been receiving increasing interest from the computer vision researchers. Rate of multimedia content shared and produced on the Internet is extremely high and the large data sources create the opportunity to exploit information from large scale data to be used for the sake of video retrieval. For example, YouTube reports that 72 hours of videos are uploaded to its servers every minute [1]. This excites the researchers to use the large scale data to exploit the information for video retrieval and indexing [3, 4, 5].

In this thesis, we address the video retrieval problem from a general to a more specific case. We address the problem of detecting unusual events by finding the usual patterns in unusual videos unlike other studies that the usual videos for learning and label the outliers as unusual videos in classification stage [6, 7].

TREC Video Retrieval Evaluation (TRECVID) community has great contribution on Multimedia Event Detection (MED) where more complicated events are taking place for detection, e.g. attempting a bike trick. Usually not all segments of a video are important, therefore; we try to find the segments that are worth to be evaluated first and use those segments to define our models which we call prototypes. We use the prototypes to detect the event of a video.

[1] www.youtube.com/yt/press/statistics.html

Another TRECVID task is Semantic Video Indexing (SIN) Automatic assignment of semantic tags to videos can be used for filtering and ranking in retrieval process. In this part, we use web images to learn the semantic tags and assign it to videos.

1.1 Motivation

Understanding complex events in unconstrained video data can be challenging. Synthetic datasets that are collected in constrained environments are not good representatives of real world actions. In an unconstrained environment, a video may include more than one scene, activity and event. Also each event may be defined by its subevents with collection of many objects and other concepts, e.g. a celebration event may include drinking, clapping or dancing actions. However, continuing on the celebration example, in a video depicting the celebration, there would be some people sitting during some small segments of the video instead of dancing. Therefore, what makes a video is not the whole video itself. Instead, we believe that the essence of the video is the combination of shorter segments in it.

1.1.1 Unusual Video Detection

People tend to pay more attention to unusual things and events, and it seems that it is generally amusing to watch them happening as proven by the popularity of TV shows like America's Funniest Home Videos, where video clips with unexpected events are shown. The so called "fail compilations" that refer to the videos that have collections of unusual and funny events are also among the most popular videos shared in social media, such as Youtube or Vine. In spite of their growing amount, there has not been sufficient attention to such videos in computer vision community.

Consider the video frames shown in Figure 1.1. If a user was presented with these videos, they would probably want to watch the ones on the top row before the ones at the bottom. Yet, what makes these videos more appealing to the audience? The *unusual events* taking place in these videos are likely to have an effect on the preference,

compared to the events that we expect to see every day. On the other hand, what makes something unusual? In most of the cases it is difficult to answer this question. Our observation is that unusual videos share some common characteristics among them like rapid motions.

Although the problem of detecting unexpected events has been addressed recently, the focus is mostly on surveillance videos for capturing specific events in limited domains. Our focus is not to detect the unusual activity in a single video, but rather to capture the common characteristics of being unusual. Moreover, we do not limit ourselves to surveillance videos but rather to the realistic videos shared in social media, in their most natural form with variety of challenges.

The data collected from web is weakly-labeled. While a video in the training set is labeled as *usual* or *unusual*, we do not know which part contains unusualness. We cannot even guarantee that a video labeled as unusual definitely contains an unusual part or a video labeled as usual does not contain an unusual part, since we query based on subjective and noisy user tagging. Our goal in such a setting is to discover the hidden properties of unusual videos from the data itself.

1.1.2 Multimedia Event Detection (MED)

Multimedia data, specifically video data in our case, on Internet is growing exponentially. The video data need to be searched, filtered and sorted according to the their content for efficient video retrieval. To be able to learn and describe the video content we need high level content descriptors [8].

We can define an event as a complex activity that occurs at a specific place and time [9]. An event may include people interaction with objects, other people or an event may consist of a number of human actions and activities.

Events are ubiquitous in real life. We can easily encounter them in daily life or on the Internet. For example, while playing a football match, watching this match on the TV or when joining your best friend's birthday party. All these events are captured somehow with different media devices. What makes us call all of these as events are

Figure 1.1: Videos on the top row contain unusual events while the videos on the bottom row do not contain any unusualness. On (a), the subject disappears and falls into the ground while walking, meanwhile the couple on (c) performs a usual walking action without any unexpected events. Similarly, subject standing on (b) collapses during an interview while two subjects on (d) perform a normal interview. Regardless of the action that the subjects are performing, our aim is to distinguish these videos.

the captured information in real life.

Recently there have been many studies that use fusion techniques for multi-model event detection [10, 11, 12]. In this study, we built our methods based upon the idea of prototypes. The prototypes are the initial models that defines some characteristics of an event. These prototypes are learned from the segments of the videos that we define a segment as a small part of a video.

1.1.3 Semantic Indexing (SIN)

Since the number of videos that people encounter every day is so high, people start using it as a communication tool. Most video search engines like Vine or Vimeo uses text or tag based search to show users what is intended to be searched. Text based retrieval is generally not very efficient for video retrieval since a video may contain more than an event or the text of the video might be wrongly annotated. We want to have relevant results from our multimedia queries.

Automatic assignment of semantic tags for high level concepts is needed for categorization of videos for retrieval tasks. Instead of using the video itself, we can use the frames that form a video separately. If we can define and learn all aspects of a concept, then we can use these models for automatic tagging of semantic tags. For this purpose, we use web images that we collect from Bing Image search engine for each concept model.

1.2 Our Contributions

1.2.1 Unusual Video Detection

While event and activity recognition have been widely studied topics [13], the literature is dominated with the studies on ordinary actions. Some of the early studies that attack the problem of detecting irregular or unusual activities assume that there are only a few

regular activities [14, 15]. However, there are various number of activities in real life.

We aim to discover what is commonly shared among the unusual videos. Our main intuition is that there should be a characteristic motion pattern in such videos, regardless of the ongoing actions and where the event happens. Unusual videos may contain a person falling down or some funny cat videos. We propose a novel descriptor, which we call *trajectory snippet histograms*, based on the trajectory information of little snippets in each video, and show that it is capable of revealing the differences between *unusual* and *usual* videos. We also use the proposed descriptor to find the discriminative spatio-temporal patches, which we refer to as *snapshots*, that explain what makes these videos unusual.

1.2.2 Multimedia Event Detection (MED)

We propose four innovative methods for feature extraction to be used in event detection for video retrieval. First three methods use clustered training data for learning. All of the four methods are used on significant segments of a video, that we call shots, note that a video may consists of more than a shot. All of the approaches except the fourth method, stand on the information learned from clusters, we name our methods as, Cluster Similarity Histogram, Cluster Id Histogram, SVM Histogram and Examplar-SVM-direct.

First approach uses the distances of the shots to each cluster center and uses them to create a feature vector for each shot. Then the feature vector of shots are combined into a vector by using average pooling or maximum pooling approaches to represent each video as a feature vector.

Second method that we propose finds the closest cluster centroid to each shots and uses this similarity information to create a histogram based on cluster ids that each shot is assigned to.

In the third and the fourth methods we use Support Vector Machines (SVMs) for learning. The third method uses SVMs to learn models from each cluster created. On the prediction phase of each shot, we use the confidence values of all learned models

for each shot's prediction. The prediction is done by using all models that are learned. Fourth method uses the famous Exemplar-SVM [16] to learn models without the need for clustering. The confidence values for each shot is kept as in the previous method. For both methods, we use average and max pooling approaches to combine shot vectors into a video feature vector that represents the whole video.

1.2.3 Semantic Indexing (SIN)

Instead of using high level concept models for automatic assignment of semantic tags, we use a simpler approach by learning concepts from web images and try to increase the quality of our models by re-ranking the images that will be used for model learning by a Multiple Instance Learning (MIL) [17] approach. The web images are re-ranked based on a MIL approach called [18] where the algorithm leverages the candidate object regions in a weakly unsupervised manner.

The rest of the thesis is organized as follows.

Chapter 2 consists of four parts. First the state of art descriptors used in this thesis are explained and the background information is given for each three chapter including Unusual Video Detection, MED and SIN.

Chapter 3, the method that extracts *trajectory snippet histograms* for detecting unusual videos and finding common patterns is introduced. Evaluation results of our method and the patches where the unusualness happen is given in this chapter.

Chapter 4 explains the data of MED task used in 2014, introduces four methods for event detection in multimedia videos, and their evaluations.

Chapter 5, the dataset used for semantic indexing of concepts is explained and, revision of a Multiple Instance Learning algorithm called MILES is made. Also the details of how we adapt MIL for image ranking for model learning is presented and evaluated in this chapter.

Chapter 6 concludes the thesis with a summary and discussions of the presented approaches with possible future directions.

Chapter 2

Background

In this chapter, we will introduce the state-of-the-art features that we used in our methods in Section 2.1. Then other studies in the literature will be given for each chapter in Section 2.2.

2.1 State of the Art Descriptors

In this section, we describe some of the low level visual features used in our studies. We will focus on three state-of-the-art features, namely Scale Invariant Feature Transform (SIFT), Opponent Sift (OpSift), Histograms of Oriented Gradients (HOG), Dense Trajectory Features and Fisher Vectors that we used to form the Improved Trajectory Features.

2.1.1 Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) has been proposed by Lowe [19] and used in wide range of areas such as object recognition, 3D modelling, image stitching, video tracking, etc. SIFT allows the key-points (interest points, salient points) detected in an image to have a representation invariant to translation, scaling and rotation.

Lowe uses Difference of Gaussians (DoG) function to determine key-points. DoG is applied to a series of smoothed and resampled images and maxima and minima of the results are used to determine the key points. Then, low responses are filtered from the set of candidate key-points. Orientation of a key-point is assigned based on the dominant orientation of gradients around the key-point. Key-points are described by the distribution of gradients for 4x4 subregions in 8 bins, resulting in 128 length feature vector.

SIFT descriptors are generally used with Bag of Words (BoW) model in computer vision [20]. To represent an image with BoW model, an image is treated as a document. Features are quantized to generate a codebook, and images are represented by the histogram of words from the codebook.

2.1.2 Opponent Scale Invariant Feature Transform (OpponentSift)

Only the intensity channel is considered and evaluated within the SIFT descriptors. An extension to original SIFT descriptors is proposed by Sande and the power of color based descriptors are proved in [21].

The definition of opponent space is given by the Eq. 2.1 O_1 and O_2 channels contain the red-green and yellow-blue opponents and O_3 is the third channel, where the intensity information is encoded as the classical HSV model. Since O_1 and O_2 do not keep intensity information, this channels are invariant to light changes. Opponent Sift descriptors are extracted by computing SIFT [19] descriptors on each channel independently. Experimentally, Sande found that an Opponent SIFT descriptor based on color-opponent channels leads to the best performance for object detection. We use Opponent SIFT features for automatic semantic indexing of videos together with SIFT [19] and Histogram of Oriented Gradients (HOG) [22] which will be explained later in this section.

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix} \tag{2.1}$$

2.1.3 MoSIFT

Another variation of SIFT [19] descriptors is MoSIFT descriptors [23]. MoSIFT descriptors are first proposed and used by Chen et al. for human action recognition in the domain of real world surveillance videos.

What makes MoSIFT descriptors more special than the previous approaches [24, 25, 26] which use temporal components for the appearance descriptors to extend spatial descriptions is its performance to explicitly encode the local motion besides the local appearance information.

MoSIFT feature descriptors are based on SIFT descriptors and this makes it robust to small deformations through grid aggreagation. With such advantages, MoSIFT descriptors are widely used in action recognition and event detection domains [27, 28, 29]. We use MoSIFT descriptors to be our base features for multimedia event detection.

2.1.4 Histograms of Oriented Gradients (HOG)

Introduced by Dalal and Triggs in [22], Histogram of Gradients (HOG) is a popular feature descriptor that is used widely in computer vision domain. It captures gradient structures that are the characteristics of local shape. HOG method finds gradient orientations on a dense grid of uniformly spaced cells on an image, and quantizes gradients into histogram bins. Local shape information is well described by the distribution of gradients in different orientations.

2.1.5 Dense Trajectory Features

Trajectory based features have been shown to be successful in different applications. Recently, in [30] relying on large collections of videos, a simple model of the distribution of expected motions is built using trajectories of keypoints for event prediction.

The dense trajectories has been presented in [31] for recognition of complex activities. We extend the use of dense trajectories to detection of unusualness through a novel descriptor that encodes the motion of trajectories.

2.1.6 Fisher Vectors

Fisher Kernel (FK) emits the advantages of generative and discriminative approaches. FK representation is proposed with the classical bag-of-visual words (BOVW) representation by Perronnin in [32]. It learns more statistics about the data by going beyond the count statistics which is used by the BOVW representation.

Perronnin et al. uses Gaussian Mixture Model to model the visual vocabulary to be able to compute the gradient of the log-likelihood that represents an image. The representation is the concatenation of partial derivatives and describes in which direction the parameters of the model should be modified to best fit the data [33]. The resulting representation is called Fisher Vector (FV) which generally gives better results then the BOV representation and it does not need the supervision as BOV does with the supervised visual vocabularies.

Perronnin uses FK with SIFT descriptors [19] in [32] but we exploit FV representation with the Dense Trajectory Features to improve the performance with a better description of the shots for MED. We name the resulting vectors as *Improved Dense Trajectory Features* Perronnin exploit FV for image classification task and many other studies use this representation within several other domains, eg. segmentation of images [34], image retrieval [35], object recognition [36] or event recognition [37]. We use FV representation for event detection with by exploiting FK on Dense Trajectory Features.

2.2 Related Work

2.2.1 Unusual Video Detection

While activity recognition has been a widely studied topic [13], the literature is dominated with the studies on ordinary actions. Some of the early studies that attack the problem of detecting irregular or unusual activities assume that there are only a few regular activities [14, 15]. However, there are various number of activities in real life.

Surveillance videos have been considered in several studies with the aim of preventing undesired events that are usually the unexpected ones. In [38] dominant and anomalous behaviors are detected by utilising a hierarchical codebook of dense spatio-temporal video volumes. In [39] detecting unusual events in video is formulated as a sparse coding problem with an atomically learned event dictionary forming the sparse coding bases. In [40], normal crowd behavior is modeled based on mixtures of dynamic textures, and anomaly is detected as outliers. Recently, prediction based methods gained attention, as in [7] which focuses on predicting people's future locations to avoid robot collusion and [41] which considers effect of physical environment on human actions for activity forecasting. However, most of these methods are limited with domain specific events for surveillance purposes in constrained environments. We are interested in revealing the unusualness in a much more broader domain focusing on web videos that are considered in the literature for complex event detection[6, 42], but not sufficiently for anomaly detection.

For finding common and discriminative parts, Singh et al. [1] shows that one can successfully detect discriminative patches on images with different categories. In [43], Doersch extends this idea by finding geo-spatial discriminative patches to differentiate images from one city to another. More recently, Jain et al. [2] showed that it is also possible obtain discriminative patches from videos using exemplar-SVMs originally proposed in [16]. In [44], a method for temporal commonality discovery is proposed to find the subsequences with similar visual content.

2.2.2 Multimedia Event Detection (MED)

MED is one of the main tasks of TREC Video Retrieval Evaluation (TRECVID) since 2010. The challenge of MED has been proven by many studies with the exponentially growing number of available videos on the Internet.

The purpose of the task is searching multimedia recordings for a given event specified by an event kit, which can be the name of the event and it's description. The final aim is to rank each clip in the collection of videos [45]. There are various number of activities in real life, therefore; an event can be defined as a complex activity occurring at a specific place and time. An event may include interaction of people, human actions and activities.

Through the MED task of TRECVID many studies are published in this domain by computer vision researchers. However, most of the work aim to build a complete video retrieval system, and therefore; the studies are based on combination of different methods that are placed upon different cues [10, 11, 46] The gained information from each method is combined with different fusion techniques. The cues may be visual, textual or audio.

Over et al. [10] uses Sift, Color based Sift, Mel-Frequency Cepstral Coefficients (MFCC), and improved trajectory features which are the FV representation of the original trajectory features as the low level features. On the other hand, as the high level features, [10] uses BoW model of Optical Character Recognition (OCR), Automatic Speech Recognition (ASR)

Besides the features used above, [11] uses low level features, semantic features and other concept detectors like ObjectBank [47] for object detection or the concept detectors learned from Sun Scene database [48] for scene understanding. On the object based MED, another study presented at TRECVID13 [9] uses the object based relative location information as a new feature [49]. In an approach based on semantic saliency, event specific event belief regions are used to capture semantic saliency.

Different from the rest, IBM does not use many features instead they use only FV representation of MoSift [23] features to present two approaches retrospective and

interactive event detection [50]. In the retrospective part they use temporal dependencies to enhance the event detection results which is called temporal modeling and [50] presents a method for the interactive event detection part with the motivation that some events are correlated. For example, the events "people meeting" and "pointing each other" can happen successively. They assume looking at such events together is more beneficial than checking one at each time. Another approach called MultiModal Pseudo Relevance Feedback (MMPRF) which is presented in [46] uses the feedback information gathered from previous steps to learn the events better.

Two of our methods are strongly based on Support Vector Machines (SVMs). We adapt SVMs and Exemplar SVMs [16] for feature extraction of a video from it's shots and in the final detection phase. Both SVMs and Exemplar SVMs methods are used in an unsupervised manner on the research development set of MED, instead of sampling positive instances, we sample them randomly. We adapt Exemplar SVMs capability of learning what an instance does not look like.

In this work we are not interested in fusion techniques as many of the other studies presented in MED task of TRECVID instead we present new methods to build more discriminative prototypes for event detection. We are inspired from the MoSift experiments presented by [50] and used the FV representation of trajectory features. Many studies are interested in frame based or clip based approaches but we are interested in snippet (small segments the video) and shot based MED. In our knowledge this is the first snippet and shot based work presented for event detection.

2.2.3 Semantic Indexing (SIN)

Automatic assignment of semantic tags is an important task to represent visual or multi-modal concepts. In this task instead of shots or snippets, keyframes of the video is used to model, note that a keyframe can be considered as a very short length snippet of a video. Semantic indexing can be used for filtering, categorization and in search for video retrieval.

SIN has been studied in the context of TRECVID and also by many other researchers. The number of collection and the number of concepts to be evaluated is one of the main challenges in this task since if the size of the collection and concepts is large it is harder to assign the tags. Another challenge in the task is the number of relevant keyframes to the concepts, therefore; we need to learn highly discriminative models for defining the concepts.

Some studies show that there is no magical solution for the problem [51], and therefore; the use of multiple descriptors and multiple classification methods is unavoidable [52]. The possible solutions are the number of descriptors to be used, parameter tuning quality, and processing time but the question to ask is which direction should we head to among those possible options.

[51, 52] use different combinations of feature extraction, feature processing, low level processing methods and show the effectiveness of those methods for visual big data processing. The success of Neural Network and recently Deep Learning based methods have proven, [53] shows the success of Convolutional Neural Networks (CNN) based methods on SIN. Eurecom in [12] shows the using high number visual features increases mean average precision (MAP) results comparing the current results with their previous year's results in [54]. Eurecom et al. 2013 uses a user based approach by considering the uploader of the video and their credibility to contribute the resulted reranking of the concept keyframes among the videos.

In this study we do not prefer to use high number of different descriptors, low level processing or classification methods. Instead of these computational methods, we believe in the representative power of images is the key to the success for a concept to learn the discriminative models. Therefore; we make experiments on the re-ranking of images to make the models learn better with Multiple Instance Learning (MIL) method called MILES [17] as used in [18], please note that the details of the MILES based method in [18] will be given in Chapter 5

In supervised learning, the learner operates over single instances and determines the labels of unseen single instances. Multiple instance learning (MIL) is a variation of supervised learning which differs in the source the learner receives. As opposed to supervised learning MIL methods operates over groups of instances. In this type of

learning, groups of instances named as bags and each bag contains multiple instances. In binary case; for supervised learning, instances labeled as negative or positive, on contrary in multiple instance learning the labels of single instances are not known. In multiple instance learning framework the only label is given to the bags where a bag is labeled as positive if it contains at least one positive instance, otherwise the bag is labeled as negative if all the instances in it are negative.

Multiple-instance learning paradigm was introduced by Dietterich et al. [55] in this name. In their work they provide a solution to the problem of drug activity prediction. The drug molecules may appear in different shapes by rotation of internal bonds and the shape determines the potency of a drug. So a molecule may adopt different shapes and only some of them are the true shapes to decide that the drug has potential. This is a completely suitable problem to be represented in a MIL framework where a bag contains multiple instances which are different shapes of a molecule and there is no information about the labels of each shape of molecule in bags. The label of being an ”active” or ”inactive drug is giving to the drug molecule bag. If at least one of the instances in the bag is the correct shape then the molecule is labeled as ”active but it is not known which one of them is the correct shape. Dietterich et al. [55] name their algorithm as the axis-parallel rectangle (APR) method.

Since then, many researchers have studied to formulate multiple-instance learning. Maron and Perez [56] introduce a probabilistic generative framework named Diverse Density and study a computer vision problem which is learning a simple description of a person from images. Zhang and Goldman [57] propose their work EM-DD by combining the expectation-maximization (EM) with Diverse Density. Different from this generative solutions for multiple instance learning Andrews et al. [58] propose their discriminative novel algorithms called MI-SVM and mi-SVM where they modify one of the supervised learning method Support Vector Machines to multi-instance problems. Wang and Zucker [59] adopt the k-nearest neighbor algorithm, [60, 61] adopt neural networks, [62], [63] adopt decision trees, Deselaers and Ferrari [64] adopt graphical models for multi-instance representations. Additionally, there are algorithms convert multi-instance representations to standard supervised learning problems MILES [17], MILIS [65]. We refer the interested readers the recent surveys on these topics of the MIL by Amores [66] Foulds and Frank [67].

MIL based methods have been commonly studied in computer vision. Multi-instance representation is suitable to many vision problems and it requires less labeling than supervised learning since the only label required are the bag labels. Some of the fields that the researchers study MIL in computer vision are image categorization [56, 68, 69, 17, 70], face detection [57, 71], object recognition and detection [72, 71], tracking [73, 74], web image retrieval and re-ranking [75, 76, 77, 18].

Chapter 3

Unusual Video Detection

3.1 Method

When large number of unrestricted web videos are considered, it is difficult, if not impossible, to learn all possible usual events that could happen, and to distinguish unusual events as the ones that are not encountered previously. We attack the problem from a different perspective, and aim to discover the shared characteristics among unusual videos.

Our main intuition is that unusual events contain irregular and fast movements. These are usually resulted from causes such as being scared or surprised, or sudden actions like falling. To capture such rapid motions we exploit dense trajectories as in [31], and propose a new descriptor that encodes the change in the trajectories in short intervals, that we call as *snippets*. In the following, first we summarize how we utilize dense trajectories, and then present our proposed descriptor *trajectory snippet histograms*, followed by description of our method for *snapshot* discovery.

3.1.1 Finding Trajectories

We utilize the method described in [31] to find trajectories. This method samples feature points densely in different spatial scales with a step size of M pixels, where M=8 in our experiments. Sampled points in regions without any structure are removed since it is impossible to track them. Once the dense points are found, optical flow of the video is computed by applying the Farnebäck's method [78]. Median filtering is applied to optical flow field to maintain sharp motion boundaries. Trajectories are tracked upto D frames apart, to limit drift from the original locations. Static trajectories with no motion information or erroneous trajectories with sudden large displacements are removed. Finally, a trajectory with duration D frames is represented as a sequence $T = (P_t, ..., P_{t+D-1})$ where $P_t = (x_t, y_t)$ is the point tracked at frame t. Unlike [31] where $D = 15$ to track trajectories for 15 frames, in order to consider trajectories with fast motion, we set D to 5. This length provides a good trade-off between capturing fast motion, and providing sufficiently long trajectories with useful information [79].

3.1.2 Calculating Snippet Histograms

We use the extracted trajectories to encode the motion in short time intervals, namely in *snippets*. Figure 3.1 depicts the overview of our method. First, for each trajectory T, we make use of the length of the trajectory (l), variance along x-axis (v_x), and variance along y-axis (v_y) to encode the motion information for a single trajectory. Trajectories with longer lengths correspond to faster motions, and therefore velocity is encoded with the length of the trajectory in one temporal unit. We combine it with the variance of trajectory along x and y-coordinates, to encode the spatial extension of the motion.

Let T be a trajectory in a video that starts on frame t and is tracked for a duration of D frames. Let m_x and m_y be the average positions of T on x and y coordinates, respectively. For each trajectory, the variance on x and y coordinates and the length of each trajectory is calculated as:

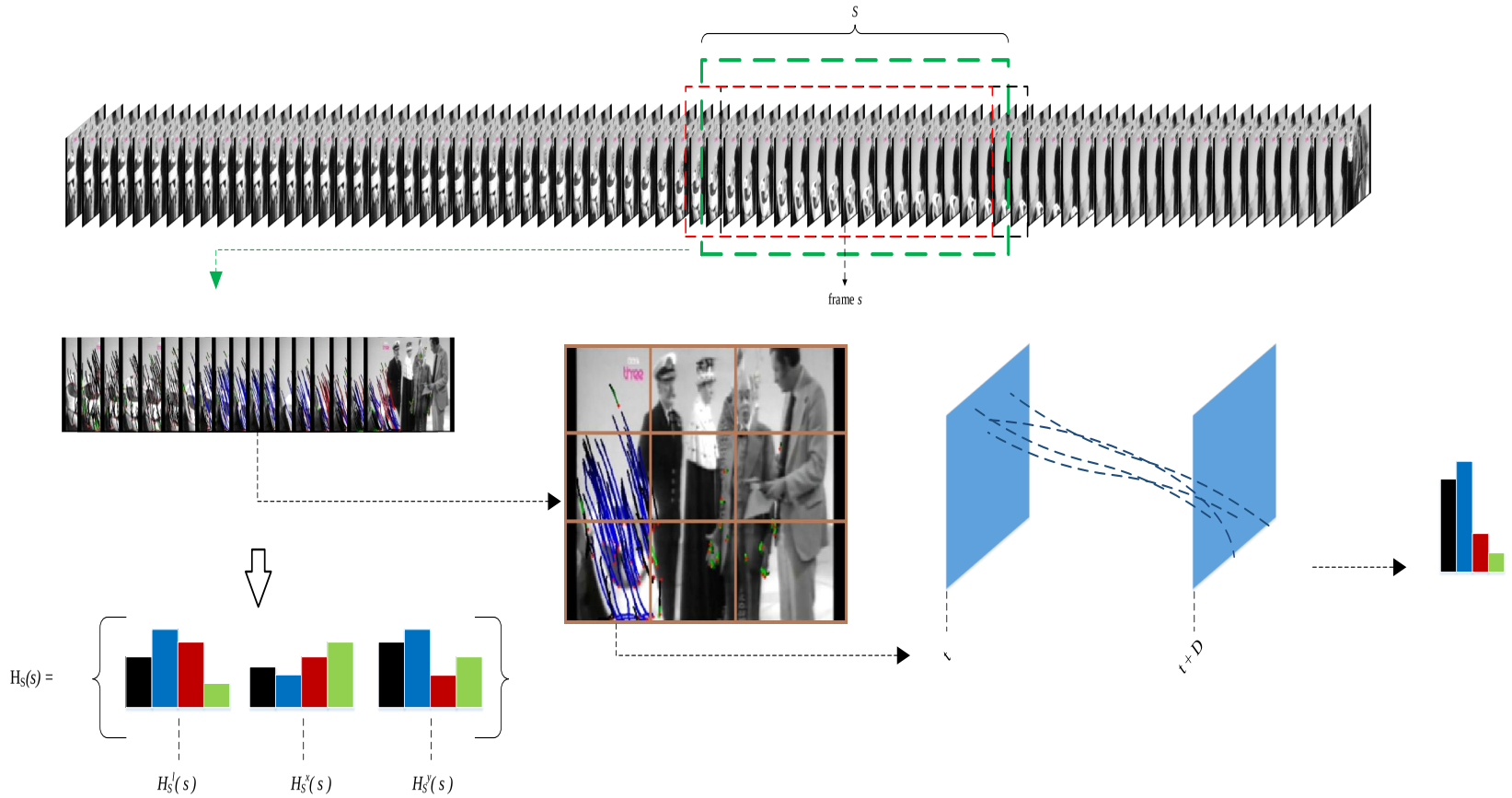


Figure 3.1: For each snippet S centered at frame s in the video, we extract the trajectory length, variance on x and variance on y values of the frames to construct a histogram of trajectories in snippets. Each frame is divided into $N \times N$ grids, and only trajectories that are centered at those grids contribute to their histogram. This process is repeated for each s in the video in a sliding window fashion.

$$
\begin{aligned}
m_x &= \frac{1}{D}\sum_{t}^{t+D-1} x_t, v_x = \frac{1}{D}\sum_{t}^{t+D-1} (x_t - m_x)^2 \\
m_y &= \frac{1}{D}\sum_{t}^{t+D-1} y_t, v_y = \frac{1}{D}\sum_{t}^{t+D-1} (y_t - m_y)^2, \\
l &= \sum_{t}^{t+D-1} \sqrt{(x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2}
\end{aligned}
\tag{3.1}
$$

Note that, videos that are uploaded to online sources, such as Youtube, can have varying frames per seconds, as most of them are collections of short video clips made by the uploader and have different formats. In order to extract motion information from the same time interval on any video, regardless of their frames per second rate, we use seconds as our basic temporal unit. Therefore, our snippets actually correspond to video sequences of lengths in seconds. In the following, we assume that snippets of length seconds are mapped to snippets of length in frames, in order to ease the description of the method.

After calculating the trajectory features for each trajectory T, at each position $t = 0 \ldots V$, where V is the length of the video, we combine them in snippets. For each snippet, we form *trajectory snippet histograms* to encode the corresponding motion pattern through extracted trajectories.

Consider a snippet S that is centered at frame s. We consider all trajectories extracted between $s - \|S\|/2 \leq t \leq s + \|S\|/2$, where t is the ending frame of the trajectory. To spatially localize the trajectory information, we divide the frames into $N \times N$ spatial grids, and compute histograms for the trajectories whose center points m_x and m_y reside at the corresponding grid. We create 8 bin histograms separately for l, v_x and v_y by quantizing corresponding values.

Let's consider l, the length of the trajectories, first. Variances in x and y dimensions, v_x and v_y, follow a similar process. Let $H_S^l(t)$ be the trajectory snippet histogram for snippet S constructed from the length l of the trajectories that end at frame t. It is a vector obtained through concatenating the individual histograms for each spatial grid.

$$
H_S^l(t) = (H_S^l(t)_{[1,1]}, \ldots H_S^l(t)_{[1,N]}, \ldots H_S^l(t)_{[N,N]}) \tag{3.2}
$$

where $H_S^l(t)_{[i,j]}, 0 \leq i, j \leq N$,, is the 8-bin histogram of trajectory lengths, for the trajectories that end at frame t and have m_x and m_y values falling into the $[i, j]^{th}$ grid. For snippet S, which is centered at frame s, we combine the individual histograms for each t, in a single histogram.

$$H_S^l = \sum_{t=s-(\|S\|/2)}^{s+(\|S\|/2)} H_S^l(t) \tag{3.3}$$

We repeat the same procedure for v_x and v_y to obtain histograms H_S^x and H_S^y respectively. Finally, we combine all of this information for a snippet S as:

$$H_S = (H_S^l, H_S^x, H_S^y) \tag{3.4}$$

At the end we have a descriptor of $8 \times 3 \times N \times N$ dimensions for each frame s of the video. These descriptors are calculated for each snippet by a sliding window approach.

In order take overall video motion in consideration, we find the minimum and maximum values of trajectory length, variance on x-coordinate and variance on y-coordinate of all the trajectories in a video. We then divide each of them into 8 bins between their minimum and maximum values, b_l, b_{vx}, and b_{vy} respectively. After finding our bin border, we start calculating our features. For a given snippet length of s in seconds, we first find its equivalent frame length, snippet frame interval l, by considering frame per second information of the video. This value changes depending on the video, and the reason why we are using *seconds* as the input and not frame number is that we would like to capture *snippets*, a period of intervals in seconds. Videos that are uploaded to online sources, such as Youtube, can have varying frames per seconds, as most of them are collections of short video clips made by the uploader. Therefore, by accepting the input as a seconds, and finding l, we extract motion information from the same time interval of videos, regardless of their frames per second. For each frame f in the video, we look at the motion that covers a length s seconds, including the motion that is preceding it and the motion that comes after it. More precisely, we look at the range of trajectories from $f - \frac{l}{2}$ to $f + \frac{l}{2}$ frames. We quantize their trajectory length

into 8-bins using b_l, and similarly quantize their x and coordinate variance using b_{vx}, and b_{vy} respectively. Alternatively, we can represent our formulation as the following:

3.1.3 Classification of usual and unusual videos

We exploit the trajectory snippet histograms for separating unusual videos from usual videos. After extracting the features from each snippet, we use the Bag of Words approach and quantize these histograms into words to generate a *snippet codebook* describing the entire video clip. Then, we train a linear SVM classifier [80] over the training data.

3.1.4 Snapshot discovery

Our goal is then to find the parts of video where the unusual events take place. We call these snippets as *snapshots* corresponding to unusual spatio-temporal video patches. Snapshots may include more than a single action. Some videos may contain unusual events where people are falling, while others may contain events where people are scared or shocked, or funny motion movements. Also, these patches should only describe actions from unusual events, not any other usual actions.

We address the problem of finding snapshots as finding *discriminative patches* in a video and follow the idea of [2]. However, in our case, a snapshot may include more than a single action unlike [2]. For example, an unusual video may contain actions like people are falling, while other videos may contain events where people are scared or shocked, or funny motion movements. Also, these patches should only describe actions from unusual events, not any other usual actions. We utilize trajectory snippet histograms to solve this problem.

First, on the training set, for each snippet we find the n-nearest neighbors using trajectory snippet histogram as the feature vector. We check the number of nearest neighbors from usual and unusual videos, and eliminate the snippets with having more neighbors from usual videos. Remaining snippets are used to construct initial models, and an examplar-SVM [16] is trained for each model.

Next, we run our trained models to retrieve similar trajectory snippet histograms for each model. We rank models using two criteria. The first criterion is *appearance consistency*. This is obtained by summing up the top ten SVM detection scores for each model. The second criterion is *purity*. This is calculated by finding the ratio of retrieved features from the unusual videos to the ones from the usual videos. For each model, we linearly combine its *appearance consistency* and *purity* scores. Finally, we rank each model based on the scores, and set the top-ranked models as our unusual video patches.

Alternatively, we also apply an approach very similar to the work in [1] with small differences in implementation. Instead of finding nearest neighbors in the beginning of the algorithm, we cluster the data in the training set into $n/4$ clusters where n is the number of instances. These cluster centers become our initial models, and we test them in the validation set. Models that have less than four firings in the validation set are eliminated, and we train new models using the firings for each model. We test newly trained models in the training set, and follow the same iteration for 5 times. We score each model using their *purity* and *discriminativeness* measures, and retrieve top T models. This method was originally proposed for still images, using HOG features. However, we are easily able to extend into videos by using trajectory snippet histograms as features.

3.2 Experiments

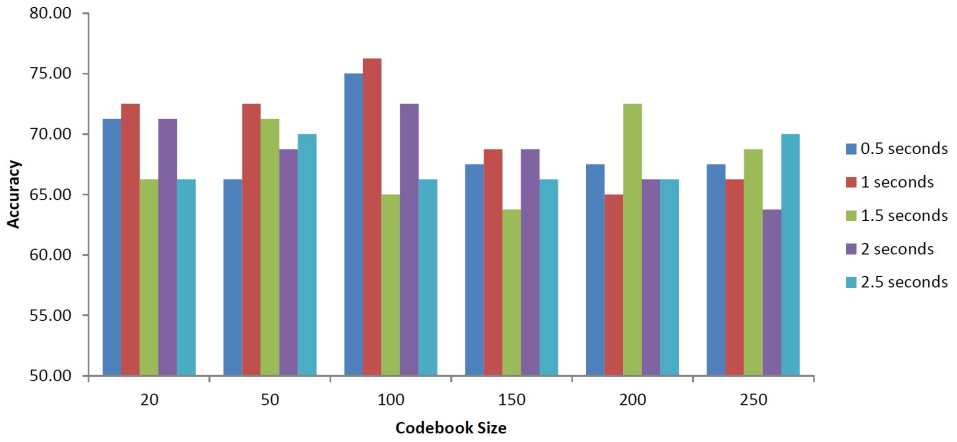
We perform two experiments using our method. The first experiment is the classification of usual and unusual videos, and the second experiment is finding and extracting unusual snapshots from videos.

Datasets: Videos used in our experiments are downloaded from Youtube, and irrelevant ones are removed manually. We constructed two different datasets. The first set, *Set 1*, has "domain specific" videos. These videos are collected by submitting the query "people falling" for positive videos, and "people dancing", "people walking", "people running" and "people standing" for negative videos. The goal of this set is to test the effectiveness of our method on visually similar usual and unusual videos with

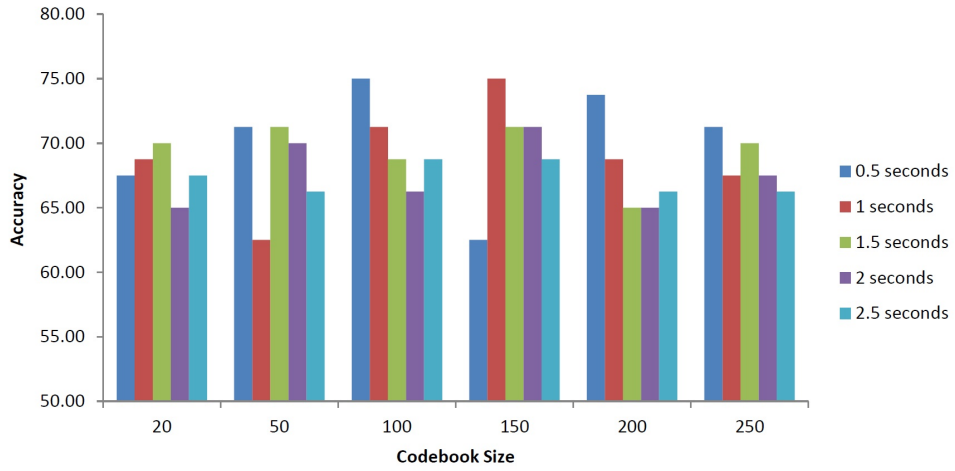
low inter-class variations. The second set, *Set 2*, is a more challenging set which consists of videos from variety of activities. Positive videos for this set are retrieved using the query "funny videos", and negative videos are randomly selected. Therefore, there is no restriction on the types of events taking place in videos of *Set 2*. Both sets have 200 positive and 200 negative videos. For each set, we randomly select 60% of videos for the training set, and the remaining 40% for the test set. Both training and test sets are balanced, meaning they have the same amount of positive and negative videos.

Unusual versus usual video classification: On the task of separating usual videos from unusual videos, we used the snippet codebooks generated from the trajectory snippet histograms. We use BoW approach to quantize descriptors and conduct experiments using different codebook sizes. We also try different snippet lengths. As seen in Figure 3.2(a), for *Set 1*, using a smaller snippet length gives better results. Note that positive videos in that set consist of people falling, and it makes sense that such action can be seen in snippets of half a second, or one second. Our highest accuracy is 76.25% using a snippet of 1 second and a codebook of size 100. In *Set 2*, since videos can contain any action, we try to learn a more broad definition of unusualness. This is a harder task, but using our descriptor we can still obtain good results, maximum being 75% with snippets of sizes 0.5 and 1 seconds, and codebook size of 100 and 150 words respectively (see Figure 3.2(b)).

We compare the proposed descriptor based on trajectory snippet histograms with the state-of-the-art descriptors extracted from dense trajectories as used in [31], namely trajectory shape, HOG [22], HOF [81] and MBH [82]. We quantize the features using Bag-of-Words approach. We evaluate codebooks with different sizes, and report the results with highest accuracy values. As shown in Figure 3.3, the proposed descriptor is competitive with and mostly better than the other descriptors when compared individually. It is not surprising to see that on *Set 1* for "people falling" HOG alone gives the best performance, since the shape information is an important factor for this task. In order to test how much strength is gained with combining different features, we combine all the other descriptors, and also include the snippet histograms as well. The results show that, snippet histograms alone can beat the combination of all other descriptors on *Set 2*, and with the combination of others it becomes the best in both



(a) Set 1 - People Falling



(b) Set 2 - Funny Videos

Figure 3.2: Comparison of performances for trajectory snippet histograms with different snippet lengths and codebook sizes. For both sets, we obtain better results using smaller time snippets.

sets. These results show the effectiveness of the proposed descriptor that encodes the motion information in a simple way in capturing the unusualness on many different type of videos.

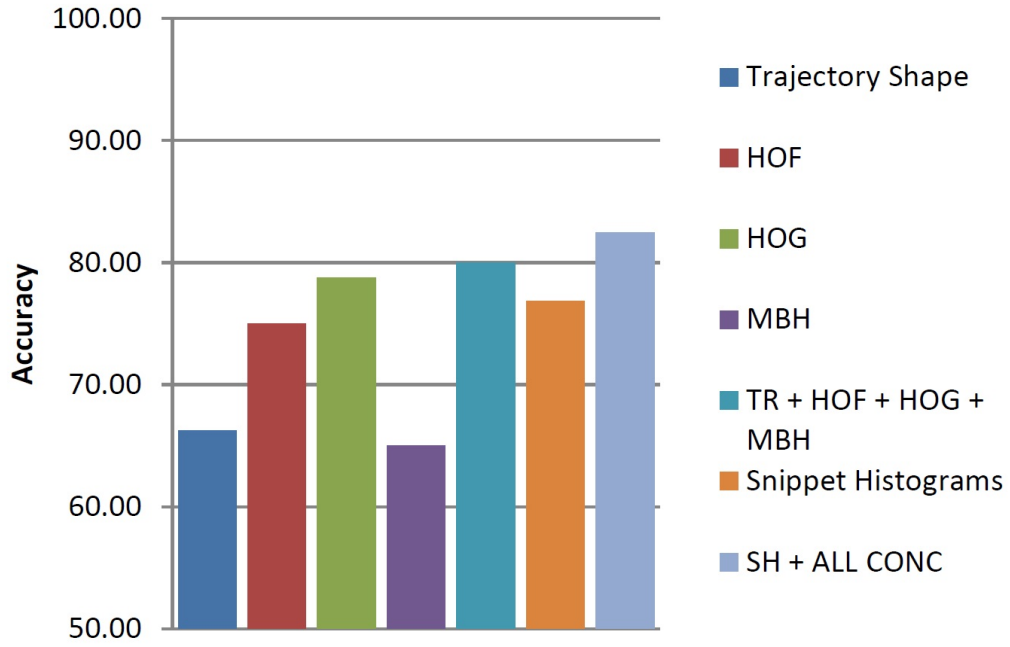
As another feature which has been successfully utilized for other problems in the literature, we exploit HOG3D feature [83] on the task of separating usual and unusual videos. However, we could only achieve 73.75% performance on *Set 1* and 65.00% performance on *Set 2* with this feature.

Our main goal is to detect unusual videos that may contain many actions, not just one action. This problem can be more challenging for traditional descriptors made for action recognition, since different actions may have different shape and appearance information. By considering complex appearance and shape information, traditional descriptors increase intra-class variance dramatically to model different actions into a single class, and this may cause many problems for classification.

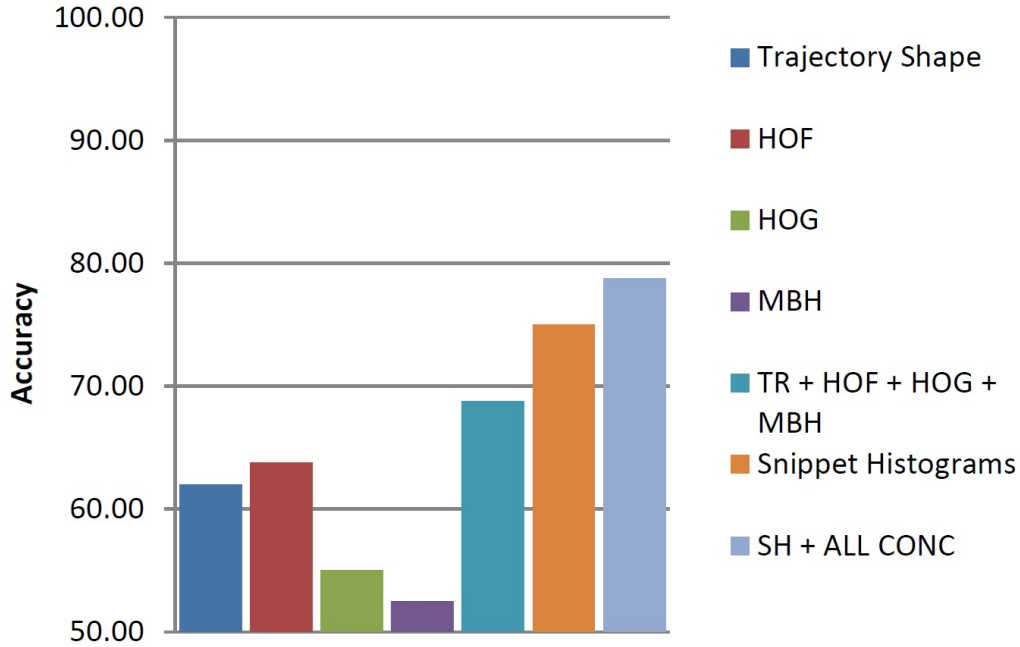
As we can see in the second part of Figure 3.3(b), we perform much better using snippet histograms in *Set 2*, which contains unusual videos from many different actions. As expected, among the traditional descriptors, the best accuracy is obtained by using HOF, which considers more optical flow information than other gradient information. However, using even simpler motion statistics, as we do in trajectory snippet histograms, performs even better in classifying a video as usual or unusual. This shows that to learn about unusuality, we only need to take simple trajectory statistics into consideration, as other appearance and motion information can add extra noise.

Discovery of Unusual Video Patches: With the encouraging results in separation of unusual and usual videos, we then use trajectory snippet histograms to find snapshots as the discriminative video patches in unusual videos. Unlike [2], we do not consider only a subset of spatial grid to find *mid-level discriminative patches*, but consider the trajectory snippet histograms of the entire spatial grid. Over a sliding window approach, with overlapping windows of length s, we detect the *discriminative* snippets. Therefore, the output is short snapshots of video where an unusual event occurs.

As seen from some of the snapshots shown in Figure 3.5 and 3.6, most of the



(a) Set 1 - People Falling



(b) Set 2 - Funny Videos

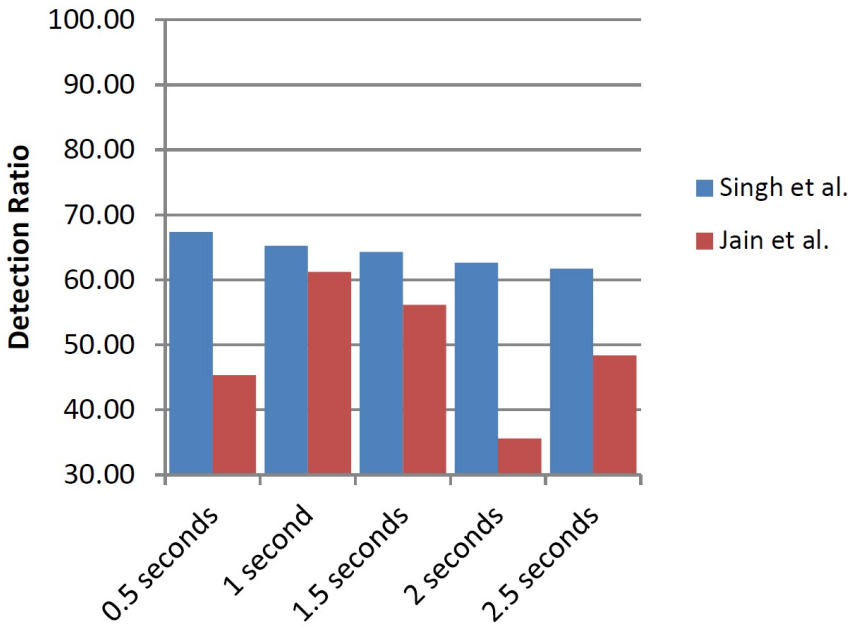
Figure 3.3: Comparison of our method with state-of-the-art descriptors. As we can observe, the performance of trajectory snippet histograms is better than other descriptors on (b), and it's concatenation with other descriptors gives us the best results in both sets.

snapshots represent motion patterns with sudden movements. These movements are the results of unexpected events, such as being scared, running into something, being hit by something or falling down. Note that our detector was also able to detect an accidental grenade explosion, which also has sudden movements and long trajectories.

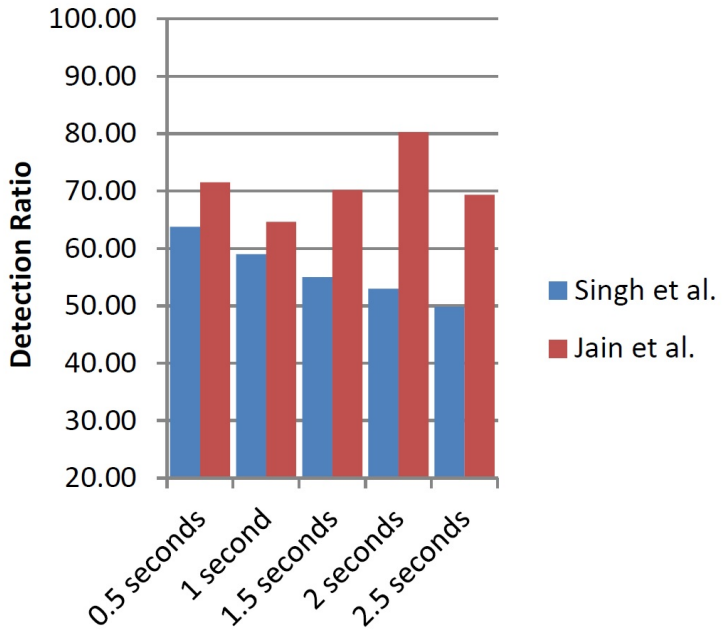
Since the ground truth for snapshots are not available, and difficult to obtain, we use a similar setting as in [43] to quantitatively evaluate the performance of detection. For each snapshot model, of how many times it was fired in positive videos out of all firings is found. As seen in Figure 3.4, again the results are better on *Set 2*, compared to *Set 1*.

We compare our descriptor with the HOG3D [83] feature used in [2] using the same setting. We obtain 25.19% on *Set 1* and 30.81% on *Set 2* using the HOG3D feature.

Most of the detected HOG3D snapshots had already been detected by snippet histograms, except for a few like those in the third column of Figure 3.7. This particular snapshot probably confused snippet histograms as there are people moving around the whole spatial grid. HOG3D descriptors localize features in x and y coordinates, therefore it was able to ignore the noise around the main subject and capture only its motion.



(a) Set 1 - People Falling



(b) Set 2 - Funny Videos

Figure 3.4: The percentage of firings in positive sets for discriminative snapshots. While using trajectory snippet histograms with [1] gives us better results for *Set 1*, [2] works better in *Set 2*.

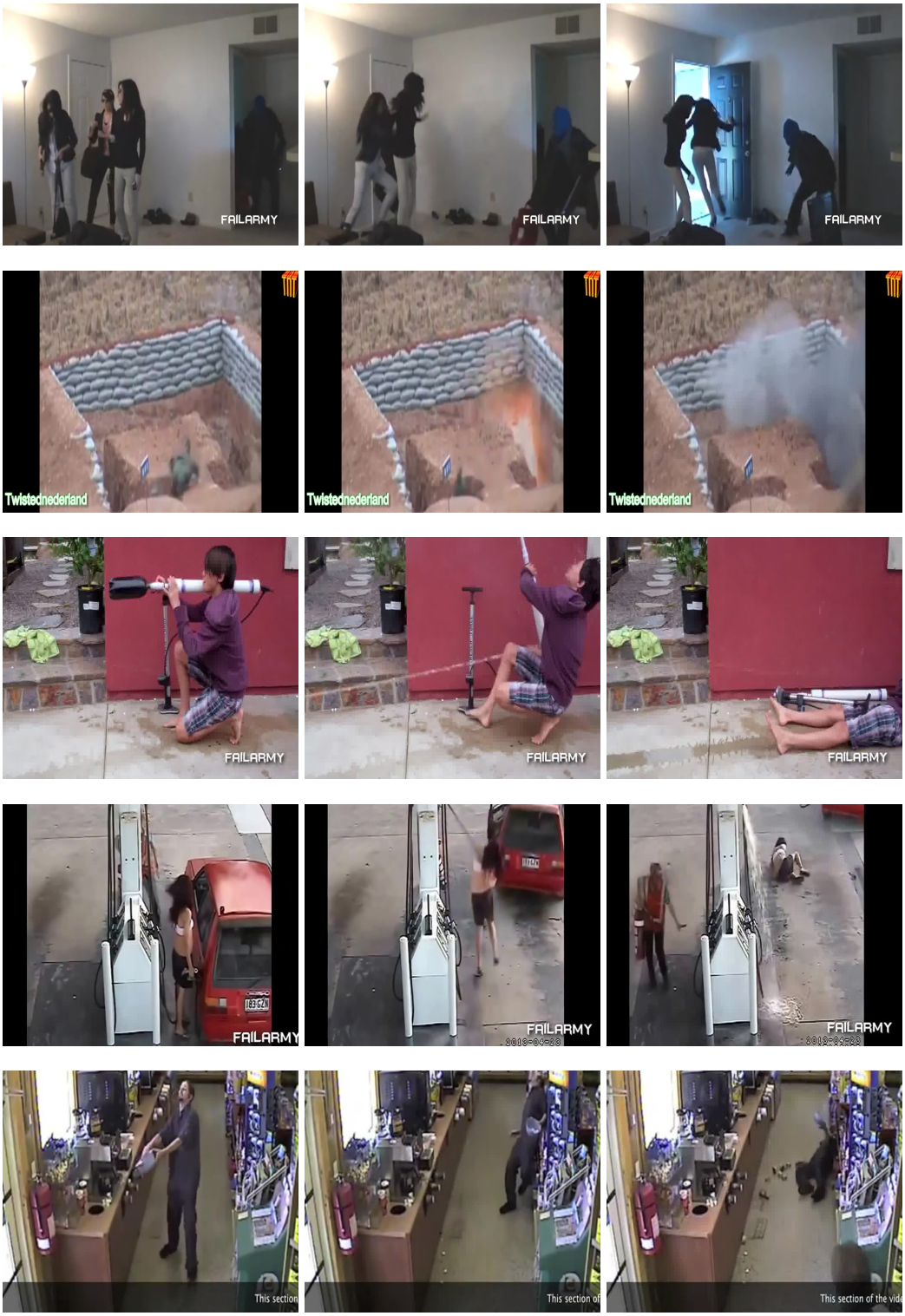


Figure 3.5: Frames from some of the detected unusual video patches using snippet histograms. As we can see most of the frames contain sudden movements.

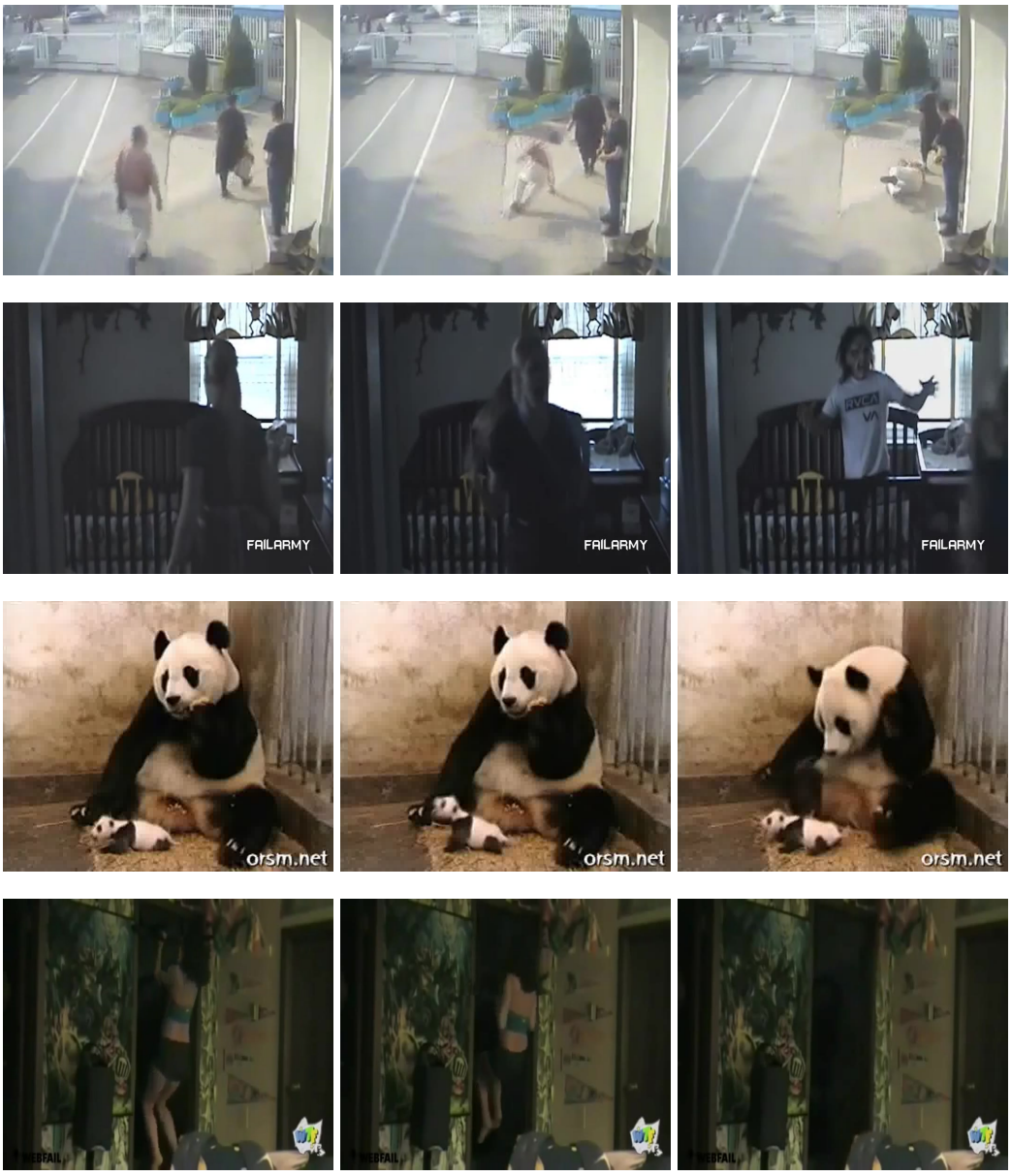


Figure 3.6: Frames from some of the detected unusual video patches using snippet histograms. As we can see most of the frames contain sudden movements.

Figure 3.7: Frames from some of the detected unusual video patches using HOG3D features. Frames on the first two columns were also detected using snippet histograms, while the frames on the third column were only detected by HOG3D features.

Chapter 4

Multimedia Event Detection (MED)

An event is defined by the videos that consist of the concepts with some shared characteristics. With this information in hand, if we find the parts that define each concept, we can model the concepts to separate an event from the rest.

4.1 Prototypes

With the observation that some of the semantic concept detectors are helpful in discriminating events if they fire consistently even if they are wrong, we decided to learn prototypes that are not necessarily semantic but commonly appearing in the data set.

We define *prototypes* as the models corresponding to mid-level representations of the videos. A prototype is a model that represents a concept or a characteristic of a concept. For example, a prototype can be as simple as a feature corresponding to the centroid of a cluster, or models learned from the clusters. These *prototypes* may capture different characteristics of semantic concepts or may correspond to an unnameable property that is shared among different concepts or events. They could be obtained from low-level visual features, as well as from audio, note that in this study we do not use audio of the videos.

4.2 Snippets and Shots

A video does not always consist of only a concept. A concept can be defined by different number of sub-concepts. Therefore, we follow the approach in which we need to consider the parts of a video that may contain a concept or a sub-concept which are used to define one of the characteristics of an event separately.

Inspired from the snippet idea introduced in Chapter 3, we make use of the small segments of the video instead of considering a video as a whole. With this representation, a video can be described by the *prototypes* that we learn from the segments.

In this chapter, we follow two different approaches to extract segments from the video. If a video is cut into segments with small fixed length, we call them snippets. Our observations showed that a video can be cut into parts that are not necessarily fixed length pieces. Therefore, the length of a segment can be dynamic according how much it differs from the other shots of a video. We call the dynamic size segments as shots. Note the difference between the snippets and the shots, a snippet is a fixed length segment of a video while length of a shot does not necessarily has the same with the other shots.

The main idea is, given a feature representation for each snippet or shot in the video, to cluster the corresponding segments into groups. Each group is then used as a prototype. Next, each segment is described in the form of prototypes. The entire video is then represented as the combination of all snippets with pooling techniques. See Figure 4.1 for an illustration of prototypes with based on shots.

4.2.1 Snippet Extraction

Extraction of the snippets from a video is done with a fixed length of 60 frames. We divide the video into small pieces where each segment consists of 60 frames. An illustration of snippets is shown in Figure 4.2.

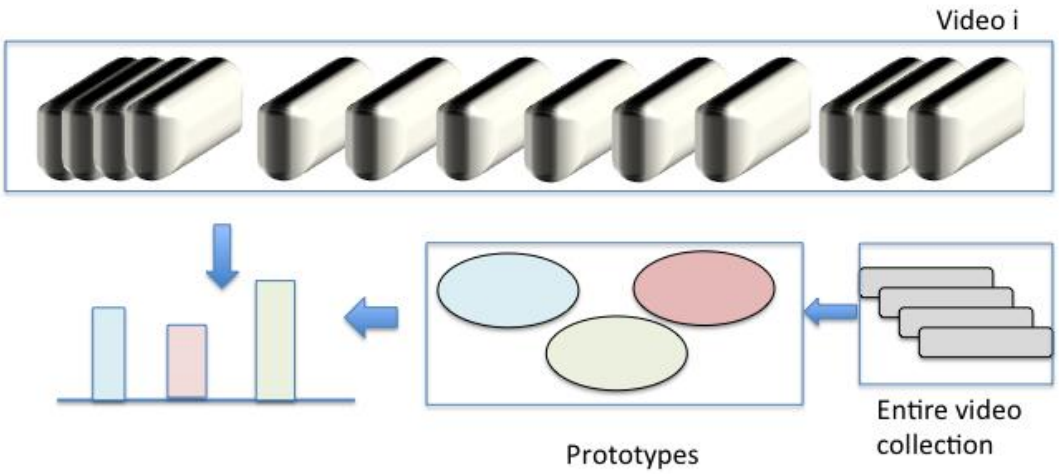


Figure 4.1: Illustration of Prototype extraction based on shots.

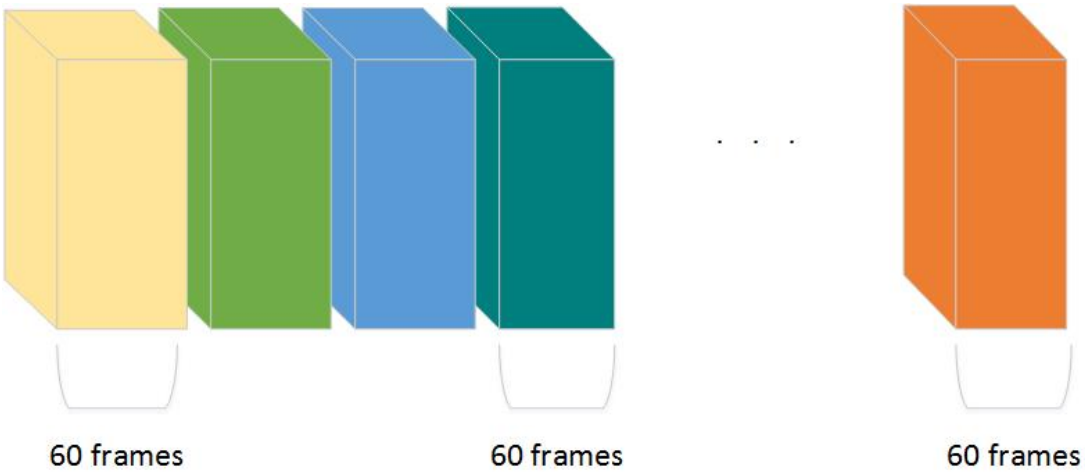


Figure 4.2: Illustration of Snippet extraction. Snippets are extracted from each 60 frames of a video without overlapping.

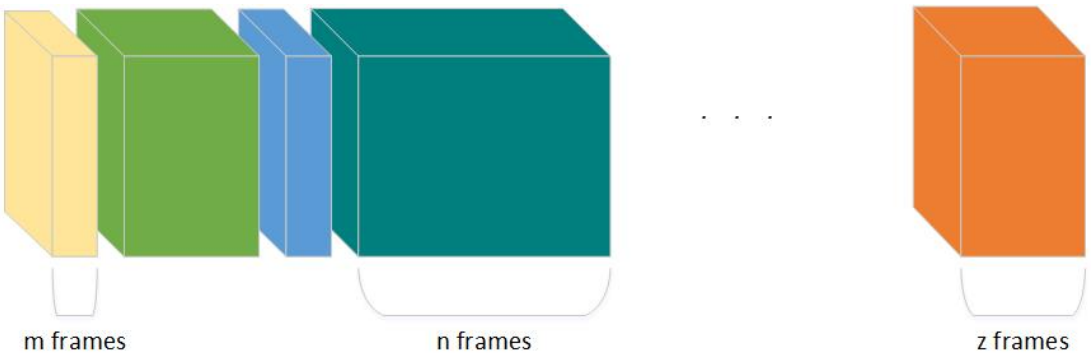


Figure 4.3: Illustration of Shot extraction. Each shot of a video may contain different number of frames.

4.2.2 Shot Extraction

The main purpose of the shot extraction process is to find the scenes in a video that are significantly different than the previous scene shown in the video. In order to find shot boundaries, we calculate the HSV color histogram for every five frames (which we will call a scene) and then we subtract the histogram from the histogram of the previous scene. If the subtracted value is larger than some threshold, and the previous shot boundary is more than three scenes away from the current one, then this scene will be a shot boundary. Therefore, using the current parameters, each shot will be at least 15 frames long. We determined the threshold value based on the average global histogram difference. The average global histogram method is defined by the difference between of the current scene and the previous scene. If the difference is larger than the average difference of histograms for all previous scenes, then this current scene is a shot boundary. A representation of the extracted shots can be found in Figure 4.3.

4.3 Initial Prototype Selection Procedure

In this section, we will first give general method description for choosing our initial prototypes. This selection process applies for our first three methods but it does not apply for the fourth method.

We are interested in finding the patterns that define the concepts where those concepts define an event or more than an event. Therefore, we are trying to reduce the similarity of the concepts to make them more reliable and precise. In order to do so, we apply clustering on the training set. We adopt the well known k-means clustering approach to find the centroids that are going to be our candidate prototypes.

Cosine distance metric is used to calculate the distance between two vectors. The cosine metric is defined as the Euclidean dot product of vectors. Let a and b the vectors whose cosine similarity is described as

$$a \cdot b = \|a\| \, \|b\| \cos\theta \tag{4.1}$$

Let V be the set of n videos in the set. $V = \{V_1, V_2, ..., V_{\text{n-1}}, V_{\text{n}}\}$, where V_{i} is the i$^{\text{th}}$ video in the video set. Then we can define a specific video in the set as V_{i} and $V_{\text{i}} = \{s_1{}^{\text{i}}, s_2{}^{\text{i}}, ..., s_{\text{p-1}}{}^{\text{i}}, s_{\text{p}}{}^{\text{i}}\}$, where $\text{s}_{\text{j}}{}^{\text{i}}$ is the j^{th} segment of i^{th} video which has p number of segments. Using k-means clustering algorithm we find k number of centroids from the training set. Let C be the cluster centroids found by k-means. Then, $C = \{c_1, c_2, ..., c_{\text{k-1}}, c_{\text{k}}\}$ where c_{i} is the i$^{\text{th}}$ centroid vector.

The usage of initial prototypes depends on the methods that are described in Section 4.4. The methods aim to create more reliable and efficient feature representations of video segments.

The new feature vectors of shots are combined with maximum or average pooling approaches to represent a video by a feature vector. We can define the maximum pooling and the average pooling approaches with the following Eq. 4.2 and Eq. 4.3, respectively.

$$f_t^i = max_j(fs_{j,t}^i) \tag{4.2}$$

$$f_t^i = avg_j(fs_{j,t}^i) \tag{4.3}$$

where j is the segment index of the i^{th} video for the t^th prototype for both equations

and fs is feature value for segment while f is the feature value for the video.

With the representation of each video by a feature vector, we are able to learn an SVM model for event detection. The extracted histograms of videos in the set are used to learn an SVM model for the final classification by cross validation.

4.4 Methods

We propose four different methods for event detection on videos. Each of the four method is introduced below. Except the fourth method, the methods are applied after finding the initial prototypes with MoSIFT Features, Dense Trajectory Features or Improved Dense Trajectory Features from each segment of the videos.

4.4.1 Cluster Similarity Histograms

The centroids are used to find the similarity of each segment in a video and are used to create Cluster Similarity Histograms. The extracted Cluster Similarity Histograms are the feature vectors that describe a video based on its segments' similarity to the prototypes, cluster centroids. Illustration of Cluster Similarity Histograms method can be found in Figure 4.4.

We use each prototype (cluster centroids) to calculate their similarity to the segments of a video and the distances are used to create the Cluster Similarity Histograms. Then, according to the definition of Cluster Similarity Histograms, we can define a distance vector of a segment to the cluster centroids as $D_{\mathrm{j}}{}^{\mathrm{i}} = \left\{ d_{\mathrm{j,1}}{}^{\mathrm{i}}, d_{\mathrm{j,2}}{}^{\mathrm{i}}, ..., d_{\mathrm{j,z\text{-}1}}{}^{\mathrm{i}}, d_{\mathrm{j,z}}{}^{\mathrm{i}} \right\}$, where $D_{\mathrm{j}}{}^{\mathrm{i}}$ is the distance histogram of j^{th} segment of i^{th} video to all cluster centroids, $d_{\mathrm{j,z}}{}^{\mathrm{i}}$ is the cosine distance of the j^{th} segment of the i^{th} video to the z^{th} cluster centroid.

At the end of the process introduced above, we extracted similarity histograms for each segment of a video. The next step is to use the extracted similarity histograms and combine them to represent a video by Cluster Similarity Histograms. To achieve this, we follow the pooling approaches. We use two pooling techniques called maximum

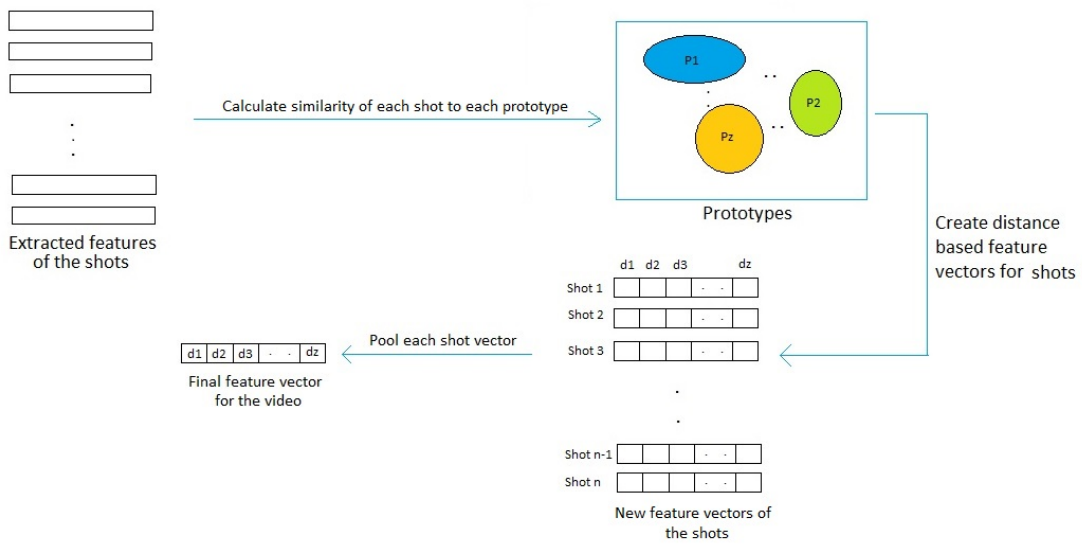


Figure 4.4: Illustration of Cluster Similarity Histogram Method for event detection.

pooling and average pooling. The maximum pooling approach is adapted by finding the segment that has the maximum distance to each cluster centroid c_i in our cluster set C. The average pooling approach is adapted by finding average similarity of all segments of a video to each cluster centroid c_i in our cluster set C. The similarities of the segments found with the average pooling or the maximum pooling techniques are concatenated to obtain the final feature representation for a video. Let F^i is the final feature vector of the i^{th} video. We define $F^i = \{f_1{}^i, f_2{}^i, ..., f_{z\text{-}1}{}^i, f_z{}^i\}$, where $f_t{}^i$ is the similarity value of the i^{th} video to the t^{th} where $t \in \{1, ..., z\}$ cluster centroid that is found by the maximum or average pooling approaches.

4.4.2 Cluster Id Histograms

Cluster Id Histogram method adopts the created prototype clusters based on their ids. Unlike from the Cluster Similarity Histograms method described in Section 4.4.1, this method uses the prototype cluster similarity information from a different perspective. In this method we use the information gained from the ids of prototype clusters that are close to the video segments and create a histogram based on the prototype cluster ids that each segment is assigned to.

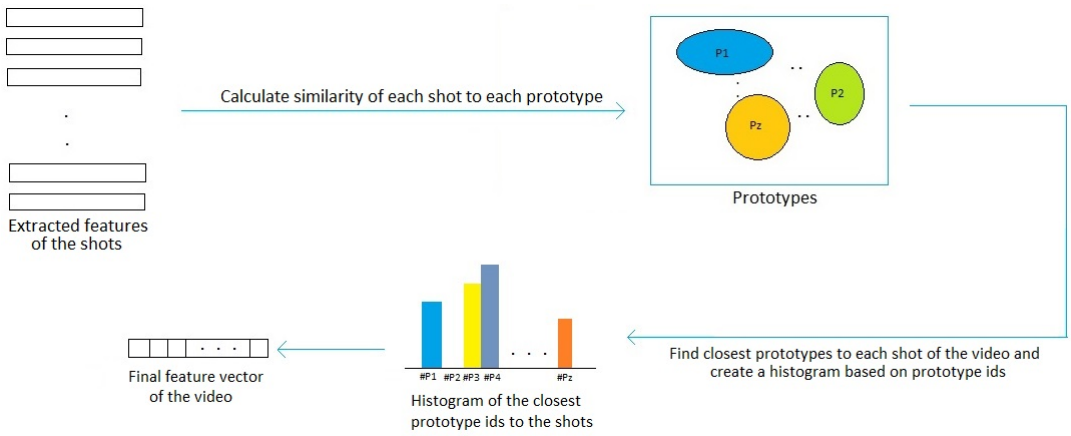


Figure 4.5: Illustration of Cluster Id Histogram Method for event detection.

The approach based on prototype cluster ids allows the dimension of our representation to have as many dimensions as required. We can create histograms as many bins as we want in order to represent the data the best. Other than just changing the bin count of histograms we create the id histograms based on two approaches. First approach is the classical histogram creation where the bin's count is increased whose index corresponds to the closest prototype cluster centroid. Second, cluster id histogram creation is based on soft assignment of prototype to the video segments. In the second approach, we use the information mined from average distance of a prototype cluster centroid to the all segments of a video and assign the cluster id to all segments of the video that have the distance smaller than the average distance. An illustration of the method can be found in Figure 4.5.

4.4.3 SVM Histograms

Support Vector Machines (SVMs) [80] are widely used machine learning technique in machine learning for supervised learning of models to find the patterns in the given data and recognize them in the classification or regression stage. One of the advantages that we also benefit from using SVMs is that they are able to extend the patterns that are not linearly separable by transformations of original data to map into a new space by using kernel functions.

In the previous methods described in Section 4.4.1 and Section 4.4.2, we use SVMs for final classification of the data by learning supervised models. Unlike previous methods, SVM Histograms method benefits from SVMs in two ways: First usage is same as the previous methods in which SVMs are used for supervised learning of models for classification. Additional usage of SVMs with SVM Histograms methods is that SVMs are used for feature creation based on clusters where we learn unsupervised SVM models. The clusters are used as prototypes in previous methods but now we use the clusters to create new prototypes with SVM models. The new prototypes are considered as candidate concepts and they are called improved prototypes. The improved prototypes are obtained from using clusters that are used as the cues with the unsupervised learning for representing the video segments.

Let n is the number of clusters we learned from the training data. Each cluster yields a prototype. We use all the clusters and their centroids as candidate concepts for the event detection. Then, for each cluster an SVM model is learned to be used in describing video segments for event detection. Learned SVM models are used for describing the data by using them to create feature vectors for each video. For each video segment, we use the learned prototypes to predict the segment and use the confidence value of the all n predictions to create the new feature vector for a segment. The created feature vector of segments for a video is used to describe the overall video as in previous methods using average or maximum pooling techniques. An illustration of the SVM Histograms method can be found in Figure 4.6

4.4.4 Exemplar SVM Direct

Exemplar SVMs are proposed by Malisiewicz in [16] for object detection. Their capability of learning what an instance does not look like with providing many negative examples comparing to a positive example has shown in [16]. The main idea is to learn models based on a single exemplar instance and many negative instances. We aim to use the exemplars as our prototypes. The difference from the previously described methods is the selection of prototypes in this phase. Instead of using clustering, we use random exemplars for this purpose.

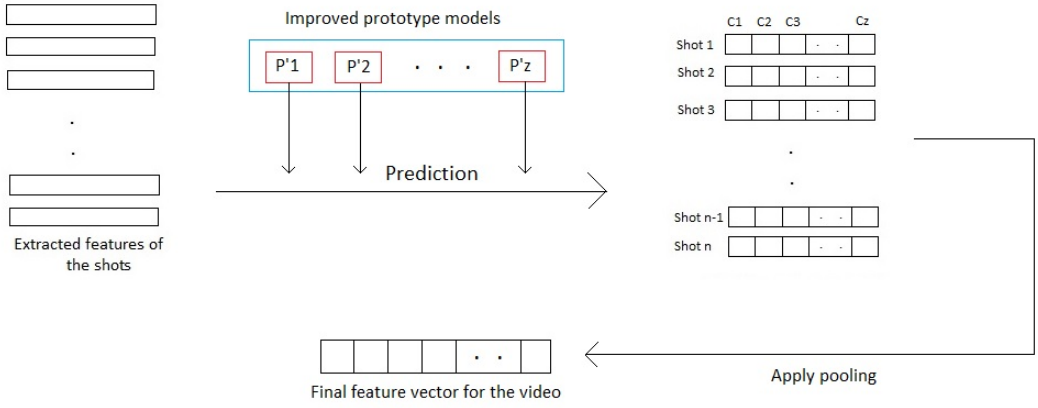


Figure 4.6: Illustration of SVM Histograms Method for event detection.

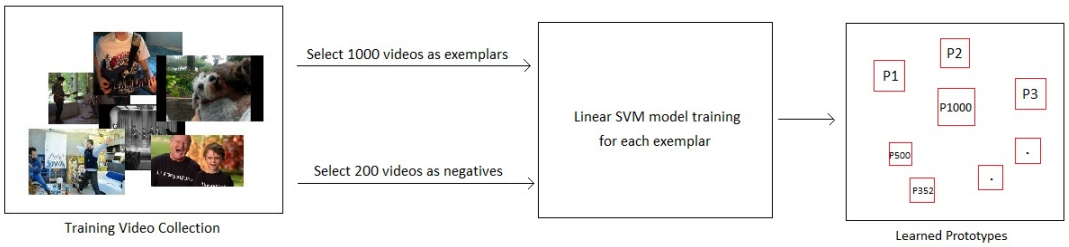


Figure 4.7: Illustration of Exemplar Method for event detection.

We use Exemplar SVM idea with a different approach that we use random instances as our positive exemplars. We randomly select n videos and consider these videos as the prototypes to be used in Exemplar SVMs model learning. As the prototype selection, we randomly select m videos for negative instance set creation.

We learn n linear SVM models with using one of the selected n exemplars and the collected negative set. Those Exemplar SVM models are used in the same way we use the classical SVMs in Section 4.4.3 by predicting the new instance and using each models' confidence values for feature vector creation. Then, a final SVM model is learned for classification as in other proposed event detection methods previously. Illustration of the method is given in Figure 4.7

4.5 Evaluation

We perform experiments with different feature types with the shots and with the snippets on prototypes by applying our previously described methods. We conduct simple experiments to have a baseline for prototypes and the baseline for using only low level features.

4.5.1 Data Sets

We mainly use TRECVID's MED 2014 (MED14) data set [9] in our experiments. As a starting point we also used a non-overlapping set containing 9746 videos of the MED14 data set to be able to see the effectiveness of our methods and to decide on the performance of the feature representations that we use.

4.5.1.1 MED Research Set

MED Research Set is used by the participants of MED task as a training set for unsupervised learning. We do not use any annotated video in the set and use it to learn our prototypes. It consists of 10161 videos without annotation.

4.5.1.2 MED 9746 Set

MED 9746 set is an older dataset for event detection task and it contains 9746 videos with annotation. The videos belong to 18 different events, eg. attempting a board trick, birthday party. Therefore, it has been used as a pre-test dataset before running experiments on a larger set. In some of the methods that we used to experiment are presented, 3014 videos are used for training and the rest 6642 videos are used for testing.

4.5.1.3 MED14 Set

MED14 set is a larger test set for the MED task for which we have event labels for each video. There are 31980 videos in this set and it contains the 20 events decided by the TRECVID ogranizers for evaluation that are used for the evaluation of the methods as it is stated by the task organizers.

4.5.2 Feature Extraction

We investigate two descriptors used widely in event detection in the literature. First one is an extension to well known SIFT [19] descriptors called MoSift [23] and the second feature we used is the Dense Trajectory Features [31]. For both features, we adopted the BoW model. Then, we have adopted the Improved Dense Trajectory Features which are based on FV representation of the classical Dense Trajectory features.

4.5.2.1 MoSIFT Feature Extraction

MoSIFT features are presented as an extension to SIFT descriptors in the domain of human action recognition. The main difference between SIFT descriptors is its capability of finding spatially distinctive interest points with substantial motions. These interest points can be considered as a subset of the interest points found by SIFT because MoSIFT applies the constraint which allows the interest points only if there is sufficient amount of optical flow around the points. Note that the usage of the optical flow constraint allows MoSIFT to be able to capture the interest points that describe the movement with a magnitude and direction. MoSIFT features represent the interest points with 256 dimensions that are the concatenation of two 128 dimensional histograms. The former one is for appearance and the latter one is for the optical flow.

We adopt the MoSIFT features with BoW model. We extract 256 dimensional descriptors of interest points in a video segment. Please note that a segment refers to a snippet or a shot of the video. We use the defined descriptors of our training set for learning the codebook for BoW model. We extract 4096 words for the codebook.

Then each segment can be represented as a histogram of 4096 words which leads us to represent a video with *n* histograms of segments. Here, *n* is the number of segments the video has. The way that we integrate each segment feature to have a video feature vector is described later in this chapter.

4.5.2.2 Dense Trajectory Features

We utilize the Dense Trajectory Features introduced in [31] and also mentioned in Chapter 3. Different from the implementation in Chapter 3, we keep the settings used in [31] where step size of M=8 pixels, trajectories are tracked upto D=15 frames to track trajectories for 15 frames. However, in this phase we do not consider the spatial relationship of points while sampling them. Once the dense points are found, optical flow of the video is computed by applying the Farnebäck's method in [78].

Tracking the points with four different features described in [31], HOG [22], Histograms of Oriented Optical Flow (HOF) [81], Motion Boundary Histograms (MBH) [82] and the trajectory shape information that describes the shape of the pixel. As we did on MoSIFT features, we applied the BoW model to each of these four features and created a histogram of 4096 dimensions with a codebook of 4096 words. To integrate each of the extracted histograms, we simply concatenate them horizontally and this results in a 16384 dimensional feature vector for each segment of the video.

We use Dense Trajectories with FV representation which we call the Improved Dense Trajectory Features. The dimension of raw feature of Dense Trajectory is 426 in the original implementation [31]. And the dimension of non-spatial FV representation is 109056. We first apply the PCA to shrink the raw feature vector to 213-dimensions. Then, we used a 256-size GMM codebook to encode the fisher vector. We get the 109056=213*256*2 dimensional FVs. Since the represented dimension of each segment is too high, we used PCA dimensionality reduction and reduced the dimension of the feature vectors to 9000 dimensions. The dimension size is selected by looking at the variance in eigenvalues.

4.5.3 Representations & Experiments

The first phase of our experiments is the decision of the feature types that we are going to use. Please note that we use Mean Average Precision (MAP) metric for evaluation of the experiments. MAP is defined for a set of queries is the the average precision scores for each query.

4.5.3.1 MoSIFT Features with Snippet Representation

In the initial experiments we start with the MoSIFT features where the dimension of the feature vector is 4096. The MoSIFT features are extracted from the video, based on the snippets representation as described in Section 4.2.1 We use the length of the snippets s as 60 frames. So, the number of snippets extracted from each video depends on length of the video.

We create the prototypes by applying k-means clustering on the MED 9746 set for which we know the event labels. In the creation of prototypes we followed two approaches. We create the clusters for each event and for all data. For the event based clustering we obtained 50, 100 and 200 clusters for 18 events on 9746 set. For the all data based clustering we created 50, 100, 200, 400, 800 and 1600 clusters by using all the training data.

To see the effectiveness of the MoSIFT and both event and all data clusterings approaches, we replace each video snippets with the closest cluster centroid. Then, average pooling is applied to obtain the feature vector of the video. As a baseline we use the original BoW MoSIFT features of snippets and applied average pooling on the snippet features. The obtained MAP results can be seen in Table 4.1 and the results of the detailed experiments with two clustering types with respect to different number of cluster counts can be found in Table 4.2.

It can be realized from the MoSIFT experiments that we are not able to beat our baseline. There is almost no difference between obtaining clusters by using all training data or by using each event's training data separately. Since obtaining event based

Table 4.1: MAP values of MoSIFT Snippet Representation for all data and event based data clustering using 9746 data set. Average Pooling approach is applied to obtain video feature vector. Best MAP values are selected for each type of the method. k represents the cluster count.

Clustering Type		
No Clustering	All Data Clustering (k=800)	Event Clustering (k=200)
0.256	0.112	0.124

Table 4.2: MAP Values of Snippet based MoSIFT experiments showing the difference between clustering using all training data and clustering using each event separately, MED 9746 set is used. k represents the cluster count. Average Pooling approach is applied to obtain video feature vector.

		Clustering Type	
		All Data Clustering	Event Data Clustering
Cluster Count	k=50	0.072	0.108
	k=100	0.107	0.109
	k=200	0.091	0.124
	k=400	0.0977	-
	k=800	0.112	-
	k=1600	0.109	-

clusters are more costly, we decided to obtain clusters using sampling on all training data from now on. Cluster count seems to have a positive effect on the MAP results but there is no observable change in the results with higher cluster counts. The difference of MAP values between the baseline and the usage of prototypes with Snippet Based MoSIFT methods is too large, therefore; we present the experiments with the Dense Trajectory Features.

4.5.3.2 Dense Trajectory Features with Snippet Representation

We use the Dense Trajectories with HOG, HOF, MBH and trajectory shape information separately with a BoW model of 4096 dimensions. The resulting feature vector for a snippet is 16384 dimensions. As in the previous experiment, we use the snippet approach where each snippet length is 60 frames. We repeat the same experiments with the same baseline method but with higher number of cluster counts. We replace each snippet's feature vector with the closest cluster centroid's feature vector. To obtain

Table 4.3: MAP values obtained on MED 9746 data set for the baseline methods using MoSIFT and Dense Trajectories with snippet representation of segments. Average Pooling approach is applied to obtain video feature vector.

Pooling Type / Feature Type	MoSIFT	Dense Trajectories
Average	0.256	0.370
Maximum	0.252	0.366

Table 4.4: MAP values for the comparison of Dense Trajectory features with different pooling approaches and different cluster counts. Results are obtained on 9746 set with using instances sampled on all training set for clustering.

Cluster Count / Pooling Type	Average	Maximum
k=5000	0.187	0.246
k=10000	0.194	0.256

the video feature vector from the snippet vectors, we also use the maximum pooling approach besides the average pooling approach.

The resulted MAP values for baseline methods show us the Dense Trajectory Features are more representative than the MoSIFT features on snippet case. The resulting MAP values can be seen in Table 4.3

We further experimented with higher number of cluster counts where k is set to 5000 and 10000 with Dense Trajectory Features by applying both average and maximum pooling approaches with replacing the snippet features with the closest centroid feature. The results can be seen in Table 4.4

We see from the baseline comparison experiment that the Dense Trajectory features work much better than the MoSIFT features. Also extraction of prototypes does not decrease the MAP values as much as it does in MoSIFT experiments. Therefore, we use the Dense Trajectory features for further experiments and use both pooling approaches since we do not observe an important change between average and maximum pooling.

Table 4.5: MAP values for the baseline results of Improved Trajectory Features. Results are obtained on MED14 set with the original features and features obtained with PCA, the dimensions are 109056 and 9000, respectively. The results of average and maximum pooling for 9000 dimensions, and also results of maximum pooling for 109056 dimensions are not available yet. These results will be added when available.

Dimension/Pooling Type	Average Pooling	Max Pooling
9000	0.0010	0.0015
109056	0.0013	0.0019

4.5.3.3 Improved Trajectory Features with Shot Based Representation

After observing the potential of the Dense Trajectory Features with the snippet representation, we further experiment with the Improved Trajectory Features with the shot representation where we consider the important segments of a video based on the change in color histograms of the segments. Improved Trajectory Features are extracted using FV representation of the Dense Trajectory Features. The original FV representation has 109056 dimensions and the reduced dimensionality of the Improved Trajectory Features is 9000. We have experimented with both number of dimensions and created baseline results for each to see the information loss due to the use of PCA. The baseline MAP results can be found in Table 4.5

Another type of experiment that we have conducted some experiments to see how well the clusters represent the data. Therefore, we created baseline results for our prototype based methods by using cluster centroids. We calculated the distance between each feature vector of shots and the cluster centroids by cosine distance. Then, each feature vector is replaced with the closest cluster centroid vector. Applying the maximum and average pooling ideas, we created the feature vector for the video. The vectors of MED14 dataset are used to train an SVM model with chi-square kernel. The obtained baseline result with these features can be found in Figure 4.8. We show the results obtained with 200, 500 and 1000 extracted clusters.

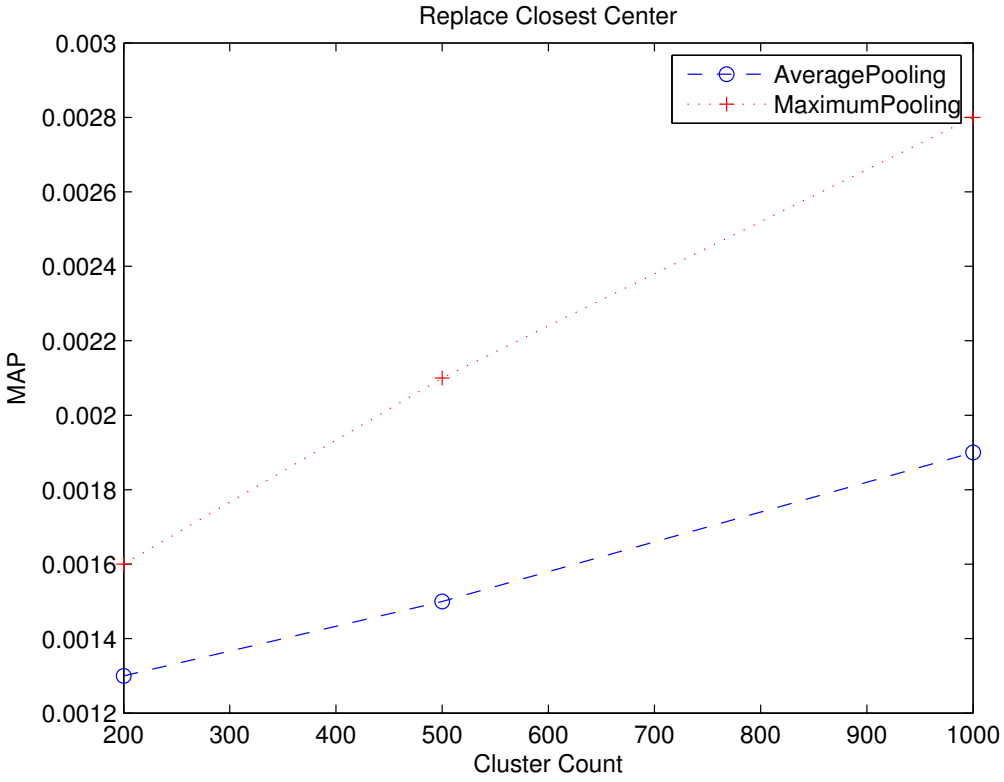


Figure 4.8: MAP results obtained with replacing the each shot's feature vector with the closest cluster centroid feature vector and applying the pooling techniques.

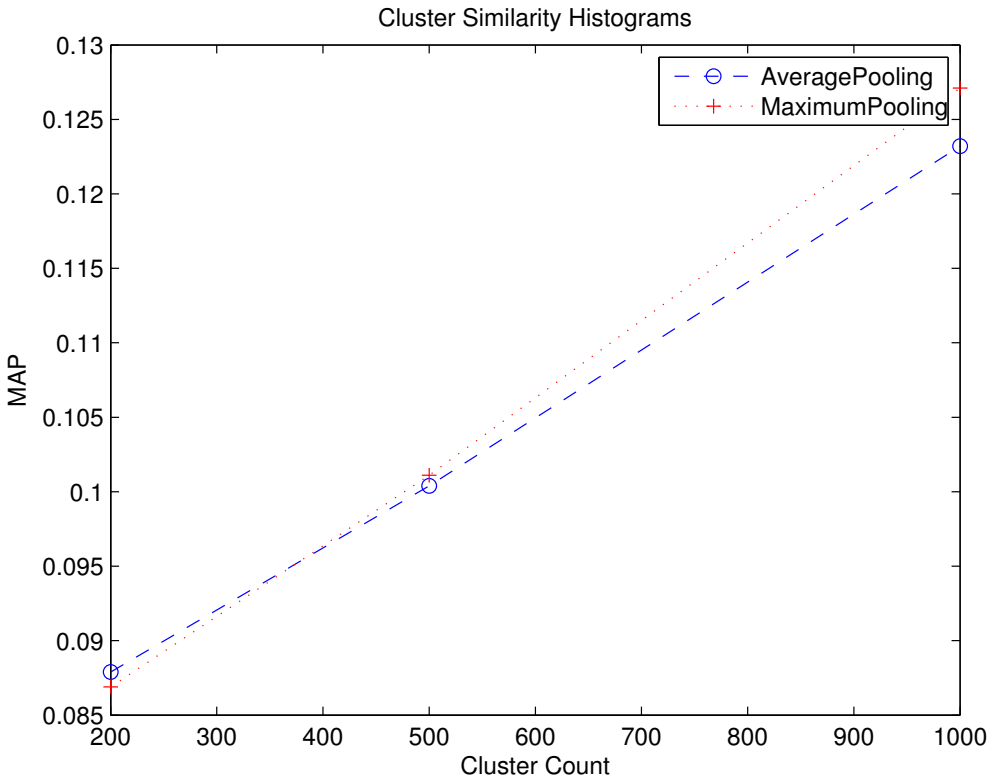


Figure 4.9: MAP results of Cluster Similarity Histogram on MED14 set. A comparison of MAP results depending on the number of prototypes used and the pooling technique is made.

4.5.3.4 Results of Cluster Similarity Histogram Method

We created prototype with k-means clustering where cluster counts are 200, 500 and 1000. The cluster centroids that define the prototypes are used to find the similarities of the video shots using the cosine distance metric. The distances of the shots to all prototypes are used to create the feature vector of the shot. To create a feature vector for the video, we used pooling techniques. The represented video features are used to train an SVM model with chi-square kernel. The MAP results of the method on MED14 set can be found in Figure 4.9.

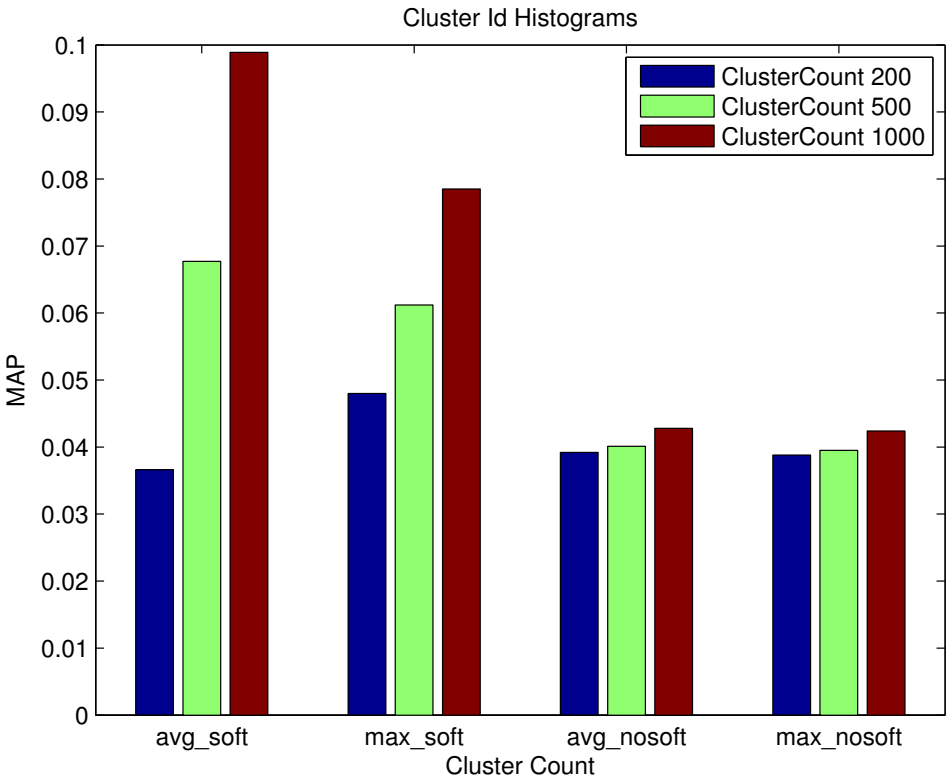


Figure 4.10: MAP results of Cluster Id Histograms on MED14 set. A comparison of MAP results depending on the number of prototypes used, the pooling technique and the histogram creation with the soft assignment method is made.

4.5.3.5 Results of Cluster Id Histograms Method

Cluster Id Histograms is the representation of similarities of the shots to the prototypes in a different perspective. The number of prototypes used in the experiments same with the previous experiment where the number of clusters are 200, 500 and 1000. We created a histogram for a video based on the prototype ids that are closest to the video's shots. The histograms are created with the naive histogram creation method and also soft-assignment method is applied where we consider the closer prototypes than the average distance of a shot to each prototype. The comparison of MAP values for soft assignments with two pooling approaches with respect to different number of clusters is shown in Figure 4.10.

4.5.3.6 Results of SVM Histograms Method

SVM Histograms uses prototypes in a different perspective than the previous methods. This method uses the initial prototypes learned with the k-means clustering where the number of clusters is 1000 to learn the improved prototypes with the SVM models. The improved prototypes are the trained SVM models with RBF kernel for each initial prototypes (clusters). The models are trained on MED Research set.

The improved prototypes are used to obtain the confidence values for each segment after prediction and use them for feature generation. The number of dimensions of segment feature vector is equal to the number of improved prototypes which is 1000. The used segments are extracted based on the shot extraction methods and all the experiments are done on the shots.

The learning of improved prototypes are not too costly, but using them for prediction on shots are. Since the complexity of this method is much higher than the previous methods we only conducted experiments with 1000 improved prototypes. The number of shot counts for each video is varies and if we consider the total number of shots to get the confidence values for 1000 SVM models, the number of predictions is too high.

As the previous experiments, we provide MAP results obtained from the method on MED14 set with using maximum or average pooling approaches for the video feature vector. The resulted feature vectors are used to train another multi-class SVM model with chi-square kernel. The MAP results of SVM Histograms method is shown in Figure 4.11.

4.5.3.7 Results of Exemplar SVM Direct Method

Exemplar SVM Direct method is different from the SVM Histograms method in terms of prototype learning approach and the used SVM kernel. In this method, we do not improved prototypes based on initial prototypes learned with clustering. Instead, we learn the prototypes with selecting 1000 random exemplar videos in the training set and selecting 200 videos to be considered as negatives. We train a linear SVM model for each exemplar with the same settings of the original study [16] where c=100 and

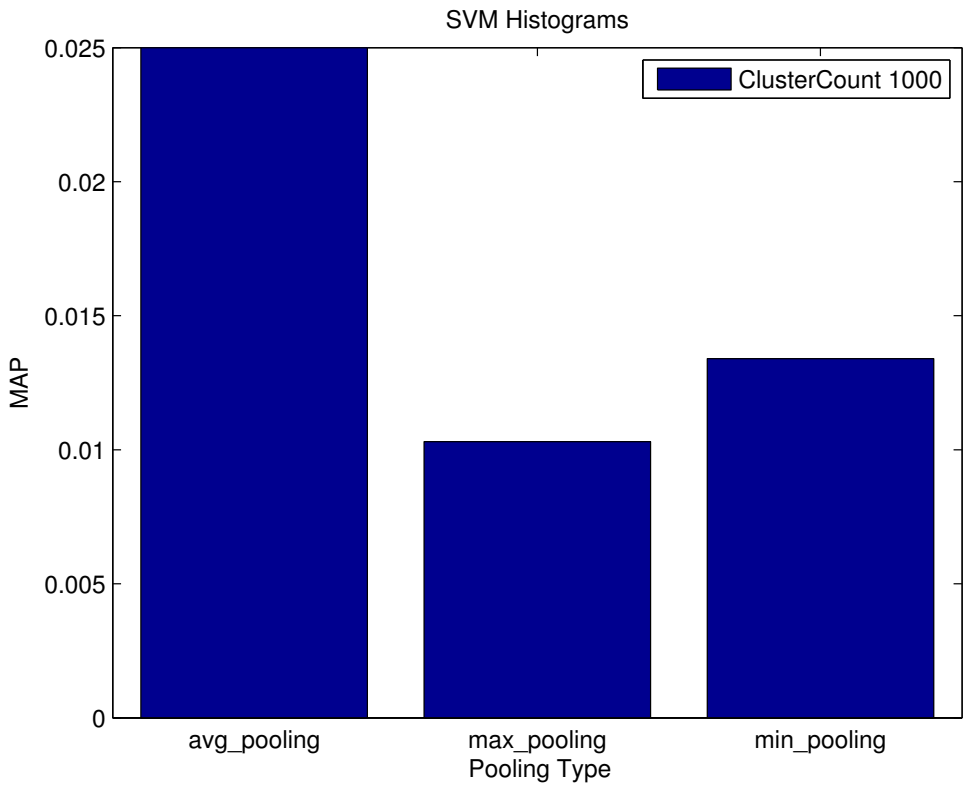


Figure 4.11: MAP results of SVM Histograms method on MED14 set. A comparison of MAP results depending on the pooling type used for video feature vector creation is compared.

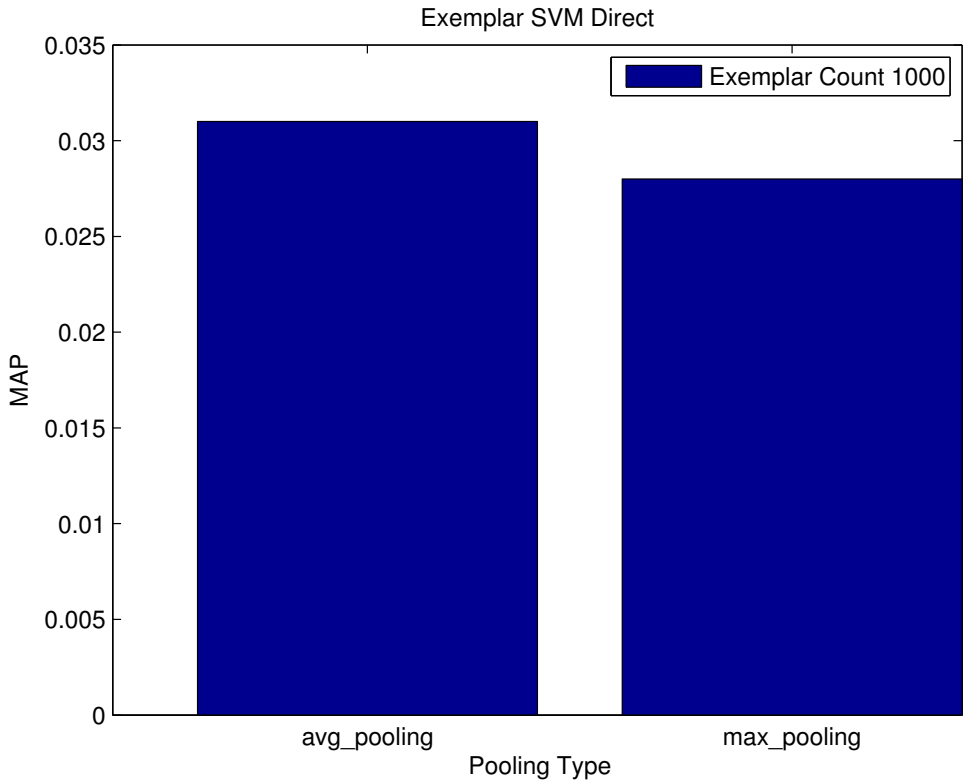


Figure 4.12: MAP results of Exemplar SVM Direct method on MED14 set. A comparison of MAP results depending on the pooling type used for video feature vector creation is compared.

the weight to the exemplar is 100. The MAP values for the Exemplar SVM Direct method can be found in Figure 4.12.

4.5.4 Discussion

We experimented with different feature types. We got the best results with Improved Dense Trajectory features comparing with the MoSIFT, the Dense Trajectory features. We can observe from the Table 4.1 that creating prototypes on event basis or on all data basis does not give us a much better performance on high number of clusters. The pooling type does not change the MAP values much. There is almost no difference between the average and maximum pooling approaches experimented on three

methods, except on the SVM Histograms methods, we can observe that average pooling approach works much better. Since the SVM Histograms method is costly in the prediction phase for creating shot feature vectors, it would be interesting to see how results would change by using different number of prototypes.

We compare the MAP values of the methods. Cluster Similarity Histograms method appears to be on top of the list. We believe that using the similarity information directly makes us gain more information about the data. Another method, Cluster Id Histograms, that use the similarity information in a different perspective is comparable with the Cluster Similarity Histograms method. Indeed, the usage of soft assignment approach while creating histograms makes Cluster Id Histograms method comparable with the first method. Surprisingly, Exemplar SVM Direct and SVM Histograms methods seem to give us the lowest results. We believe the confidence values of the prototype SVM models would give us better results. However, with the observation of some SVM models are not well trained, since we use very few instances to train the models. We sampled 200 videos from the research set to learn the initial prototypes for SVM Histograms method and the 200 videos provides us with 14852 shots. Since we learn 1000 prototypes on 14852 shots, this results us clusters created by very few number of instances. We believe the instance number that falls to each cluster is the main reason that our SVM Histograms methods does not give comparable results with the rest.

Chapter 5

Semantic Indexing (SIN)

5.1 Methods

Instead of learning the concepts with complex methods, we prefer to use web images to learn simple SVM models for indexing and classification. It is easier to index if we manage to learn discriminative models with the data in hand, instead using the complicated learning methods. For this purpose, we collected a set from the Bing Image Search Engine and the use for learning. Since the web data is noisy, we only need to use the relevant images. Therefore, we use a subset of the collected set based on the ranking of the search engine, since the less relevant images are ranked low on the search engine.

Another option is without trusting the ranking of the search engine and producing our own re-ranked image list for each concept. We use a MIL based approach proposed by Sener and Ikizler in [18] to re-rank of images in the set.

For this task, we gather images for our queries using text-based image search engines to train classifiers. However text based search engines may return irrelevant images due to the reasons such as wrong, irrelevant tags, polysemy and synonymy for queries. Since we use a supervised learning method SVM in our evaluation to compute classifiers we aim to retrieve the purest set of images for each query. There are some

methods which use the visual content of the images to improve the ranking order of images returned by text based search engines [84, 85, 86, 18].

In our study, we use the work proposed by Sener and Ikizler-Cinbis [18] to re-rank the images returned by text based image search engine. They automatically construct multiple bags from the returned list of images by text based search engines then utilize these bags by ensembles of Multiple Instance Learning classifiers. Finally, they re-rank the images based on multiple classifier scores. As they suggested, we use sliding window approach for bag sizes k = 1,2,3,4 and 5 to construct positive bags for MIL framework. Then, we use MILES [17] algorithm as proposed, which works by embedding the original feature space x, to the instance domain $\mathbf{m}(B)$, where each bag is represented by its similarity to each of the instances in the dataset. The similarity between a bag $\mathbf{B}_i$ and a concept c^k is determined by

$$s(c^k, \mathbf{B}_i) = \max_j \exp\left(-\frac{D(x_{ij}, c^k)}{\sigma}\right), \tag{5.1}$$

where $D(x_{ij}, c^k)$ measures the distance between a concept instance c^k and a bag instance x_{ij}.

$$\mathbf{m}(B_i) = \left[s(c^1, B_i), s(c^2, B_i), \ldots, s(c^n, B_i)\right]^T. \tag{5.2}$$

We then use an SVM classifier over this embedded representation. Then we apply late fusion to the classifier scores for each bags size and re-rank the images.

Illustration of the top ranked 20 images belonging to the Baby concept obtained from the original search engine results is given in the Figure 5.1 and the lowest ranked 20 images are given in Figure 5.2. After applying MIL approach to the images of the Baby concept we re-ranked the image list. Image re-ranking results with the MIL based approach is shown in the Figure 5.3 and Figure 5.4.

Figure 5.1: Highest ranked images of the Bing Image Search Engine for the Baby concept.

Figure 5.2: Lowest ranked images of the Bing Image Search Engine for the Baby concept.

Figure 5.3: Highest ranked 20 images of the image list obtained from the MIL based approach for the Baby concept. The scores of the images are given at the top of each image.

Figure 5.4: Lowest ranked 20 images of the image list obtained from the MIL based approach for the Baby concept. The scores of the images are given at the top of each image.

5.2 Evaluation

5.2.1 Datasets of SIN

In this chapter, datasets used in the SIN task are introduced. The dataset is collected by the TRECVID organizers [45].

5.2.1.1 IACC.2.A Dataset

IACC.2.A set is collected by the TRECVID organizers for the SIN task of 2013. For the task 2014, the IACC.2.A set is given to the participants to conduct experiments on it and evaluate their methods. The set consists of 200h of videos drawn from the general IACC.2 collection using videos with durations between 10 second and 6 minutes. The frames of this set is used for indexing and we give results on this set in Section 5.2.4.

5.2.1.2 IACC.2.B Dataset

IACC.2.B set is provided by the organizers for the 2014 task, collected by the TRECVID organizers for the SIN task of 2013. It is as large as the IACC.2.A set, 200 hours of videos drawn from the general IACC.2. Since this set is used for this years evaluation, we are not able to see the evaluated results yet.

There are a total of 500 concepts for each IACC.2.A and IACC.2.B sets but the evaluation is done by using 60 concepts selected by the organizers. Some of these are anchorperson, demonstration or protest, quadruped.

5.2.2 Data Collection from Web

The proposed method for semantic indexing problem is based on the web images crawled from Bing Image Search Engine.[1] The queries for crawling are the 60 concept names that are selected by the organizers. We used the concept names as it is since it is not allowed to extend or change the concept names. We tried to collect 1000 images for each concept, but the number images provided by the search engine differs for each. Therefore, if the number of images provided is less than 1000, we were able to collect the maximum number of images that is provided by the engine.

5.2.3 Feature Extraction

To describe the details of the descriptors where we use SIFT [19], Opponent SIFT [21] and HOG [22]. The details of the used descriptors are explained in the Section 2.1.

Before finding the interest points and describing them with SIFT and Opponent SIFT, all images are downsampled to have 15000 pixels and the height to width ratio is kept the same. Then, BoW model is applied to SIFT and Opponent SIFT descriptors. A codebook with 1000 words is generated for BoW model and applied to the frames using spatial five tiling. An illustration can be found in Figure 5.5. We sample 4000 frames from IACC.2.A set and created 1000 words on this subset to create our codebook. The resulted dimension of a feature vector for an image is 5000 for both descriptors since we apply five tiling with 1000 words on SIFT and Opponent SIFT descriptors.

HOG features are extracted using eight as the bin size with four orientations. HOG is extracted on the images that are down sampled where the width and height is 200. Therefore, the dimension of the resulted feature vectors is 10000.

[1] www.bing.com\?scope=images

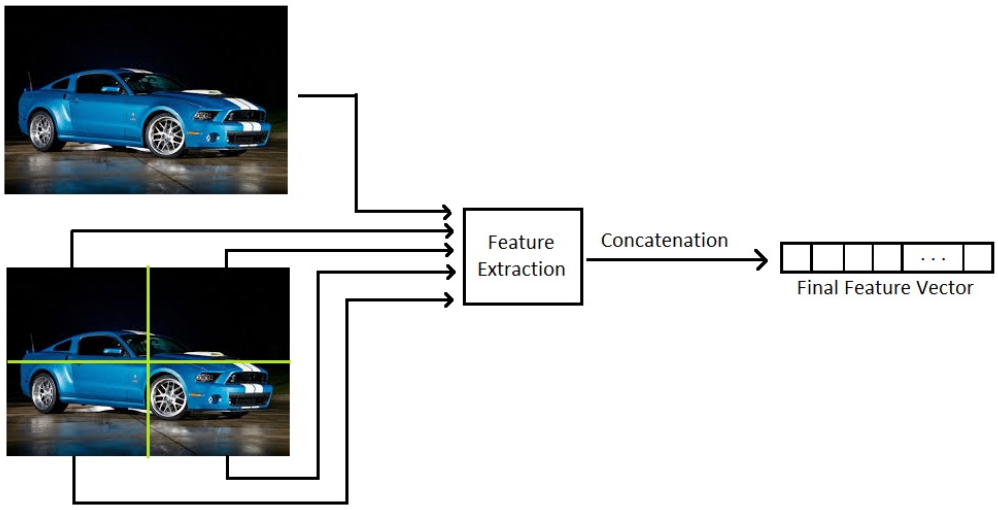


Figure 5.5: Feature extraction process for SIN methods is illustrated with spatial five tiling. Features are extracted for each tile and then by concatenation of the extracted features the final feature vector is created.

5.2.4 Experiments

We conduct experiments with the SVM models that we learn from our collected concept dataset with the ranking of the search engine and with re-ranking based on MIL based approach. The results are based on interpolated average precision (iAP) metric. Interpolated precision is where you pick a recall level r and for all recall levels $r' >= r$; it is the best precision you can achieve. The aim of the SIN task is to provide an image list with 2000 images for each concept, where we rank the images according to how relevant the image is.

SVM models for m concepts can be trained as a multi-class basis SVM or binary-class SVM basis, where each model is learned with the n images selected for a concept and $2n$ images sampled from other concepts are used as negatives. In the binary-class SVM approach, we have the same number of models with the m number of concepts and the prediction is done by finding the maximum confidence value that we get from all learned models. However, for multi-class SVM approach we only have a model used for learning of m concepts and the predicted concept is the maximum confidence value got from the model. For both cases we have compared the linear and RBF kernels

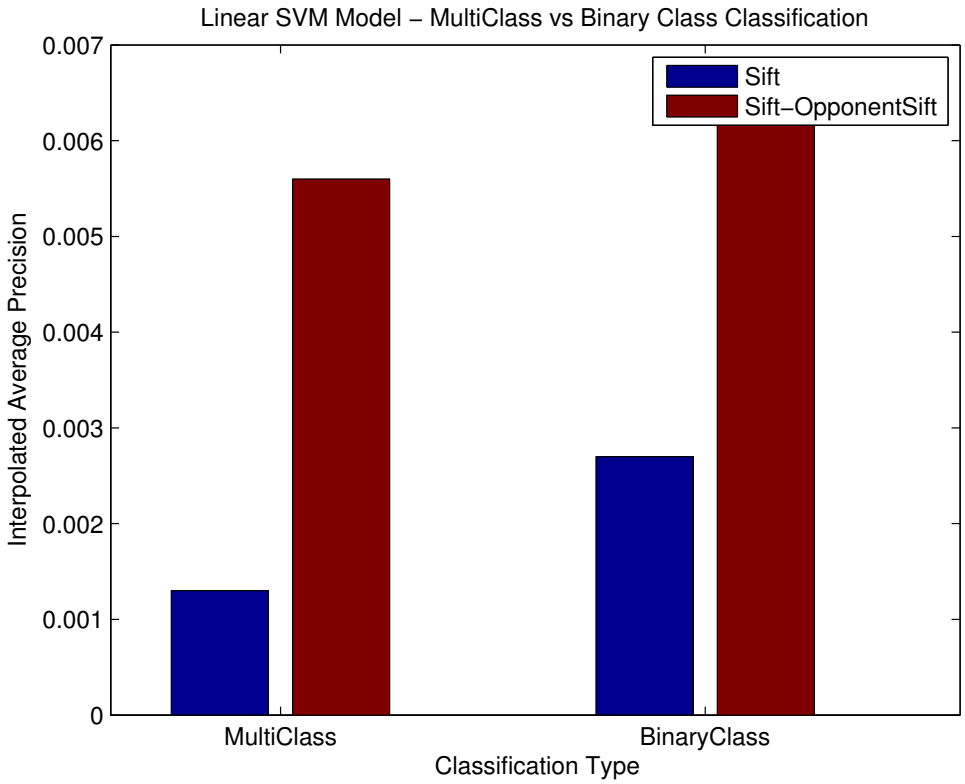


Figure 5.6: Interpolated Average Precision Results with the comparison of multi-class SVM model learning and binary-class SVM model learning approaches with linear kernel.

with different parameters.

We first try to see the difference between the binary-class SVM learning and multi-class SVM learning approaches using SIFT and the feature obtained by the concatenation of SIFT and Opponent SIFT features. The dimension of the resulted features are 5000 and 10000, respectively. The first experiment compares these two approaches with a linear kernel SVM model learned by using all the images in our collection. The results can be seen in Figure 5.6. The binary-class SVM approach seems to work much better on our collection since the number of concepts is high and they are not easily separable from each other. Even if it is more computationally expensive compared to the multi-class SVM approach, we will use the binary-class SVMs in the rest of the experiments.

Table 5.1: iAP results obtained by using all web images for binary-class SVM model creation with Linear and RBF kernels where we used the concatenation of SIFT - Opponent SIFT features.

Result Metric/Kernel Type	Linear Kernel SVM	RBF Kernel SVM
iAP	0.0063	0.015

The other experiment aims to show the difference between RBF and Linear kernel SVM models for prediction. For both type of models, all of the web images are used for learning and the number of negative images sampled is two times the number of positive images. Even if RBF kernel is slower than the Linear kernel SVM models, since the iAP results are better than the Linear kernel SVM models as shown in Table 5.1, unless it is stated we use RBF kernel for next experiments.

Next experiment aims to show the effectiveness image ranking of the search engine. For this purpose, we vary the number of images that the RBF kernel SVM models are learned from for each concept. We use the top p number of images from the ranked image list of the search engine, where p is 100, 200, 400 and all of the images for a concept. If the total number of images that a concept contains is less than p, we use the highest number of images that is available for that concept.

In this experiment, besides the usage of SIFT - Opponent SIFT features as they are proven to work better in previous experiment, we also combined the SIFT - Opponent SIFT features with the HOG features and compare the results. The iAP results obtained with using the image ranking of search engine is given in Figure 5.7 One of our observations is that our models are not good enough to discriminate some of the dark or light colored images from the rest. Therefore, we apply a very simple color selection procedure that effects the final ranking of retrieval results in a positive way. We simply take the average of the intensity values of an image and if the average value is between an interval, the image is placed where the algorithm is already predicted. However, if the value falls outside of the defined interval the image is placed to the last rows of the retrieval image list.

We also provide results that we obtained using the images from the ranked list

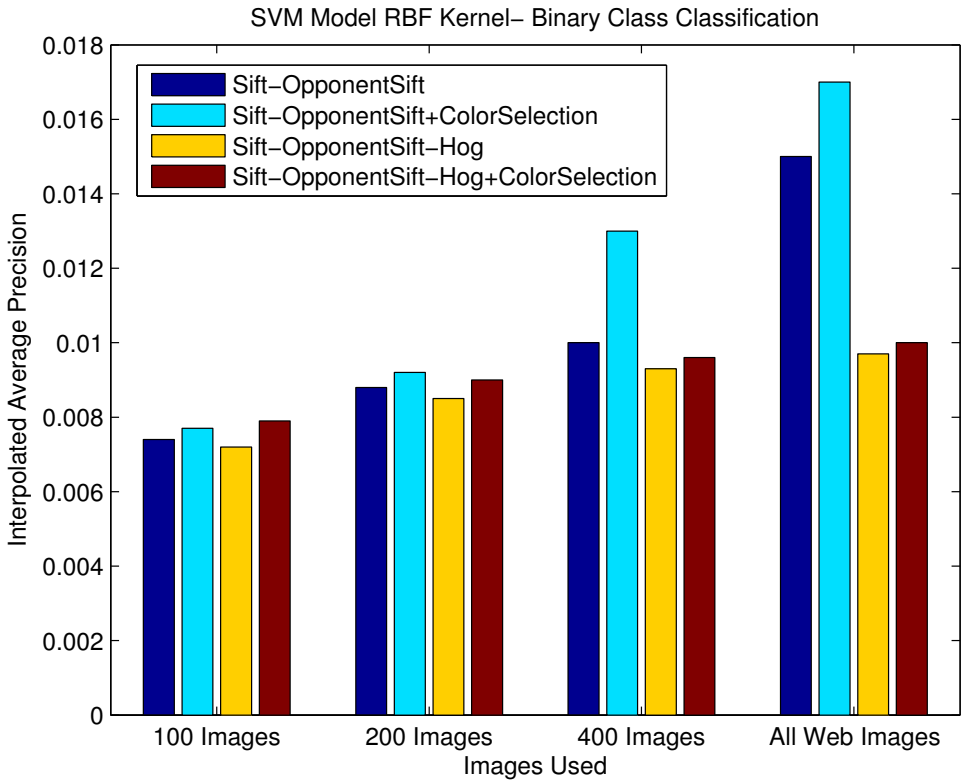


Figure 5.7: Interpolated Average Precision Results obtained with using the image ranking list of the search engine. 100, 200, 400 and all the images are used and trained binary-class SVM models with RBF kernel where for each model number of negative images are the two times the number of positive images used. For the color selection method we used the interval [20,230], meaning that if the average intensity value of the image is in the interval we consider the image, if it is not we put the image at the end of the ranked list.

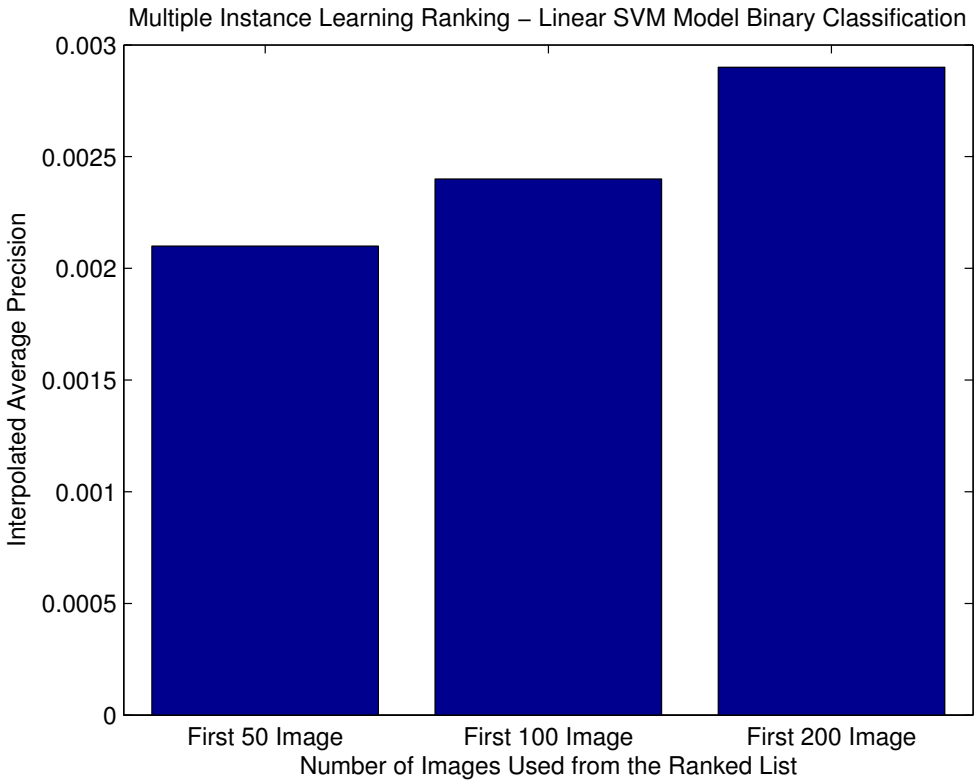


Figure 5.8: Interpolated Average Precision results obtained with using the image ranking list of MIL approach. The top ranked 50, 100 and 200 images are used and trained binary-class SVM models with Linear kernel where for each model number of negative images are the two times the number of positive images used. SIFT - Opponent SIFT features are used in this experiment.

that we generated using MIL based approach. We used the top 50, 100 and 200 images from the re-ranked list of MIL approach and the results are provided in Figure 5.8. The iAP results of the MIL based approach seems much lower than the approach that we used search engine's ranked list. The comparison of the iAP values for search engine based method and the MIL based method is given in Table 5.2 where the same number of images used for model learning.

Table 5.2: The comparison of Interpolated Average Precision results of Search Engine based and MIL based approaches for the same number of images used from the ranked lists where the number of images are 100 and 200.

Method/Image Count	100	200
Search Engine	0.0088	0.01
MIL	0.0024	0.0029

5.2.5 Discussion

We have experimented with different number of settings for SVM model learning, including using ranked images of Bing Image Search Engine and re-ranking the search engine's results with a MIL based approach. Surprisingly, MIL based approach works much worse than the original ranked list provided by search engine for each concept. This may be due to the number of dominant groups for a concept. If a concept refers to a number of different concepts at same time, [18] ranks the dominant sub-concept first and then ranks the rest of the images in a concept. Another reason that is probable is that the method of [18] is shown to work well on image ranking where each image is labeled with text based tags but image search engines like Bing also compare the images visually and produces the ranked list accordingly. The MIL based method seems to produce results a bit lower than the results produced based on the ranked list of search engines'.

Still the results obtained with the binary-class SVM models by using the image ranked list of Bing produces considerable results obtained compared with the computational methods. Also we may observe that the difference of the iAP values where we use top ranked 100 images and all the collected web images is less than the expected. Therefore, it can be said that the images that appear at the top of the image list define the concept better than the rest.

Chapter 6

Conclusion

In this thesis, we presented the methods for leveraging large scale video data for video retrieval. We developed methods on three domains, Unusual Video Detection, multimedia event detection (MED), and Semantic Indexing (SIN).

The problem of detecting unusuality or anomality has been handled in a very constraint setting up to now. Usually, the video from only one camera is used, so all the actions are seen from one angle only. Most of the works in the literature solve this challenge by detecting irregular events by finding regular events. However, this limits the problem.

Our main goal in this part of the thesis is to generalize the solution for the problem described above. We would like to find unusualities in videos, regardless of the scene, actions, or from what angle the video was taken from. This is not an easy task, as we have an infinite number of possible actions, and it would be impossible to learn them all. Furthermore, same action can be seen completely different in two different perspectives. We propose a simple but efficient method to capture the unusualness in videos, and our experiments give us promising results. As far as we know, this is the first work that attack the problem of discovering unusualness in videos shared in social media regardless of the ongoing events.

The growing number of available video data allows us to gain more information

to learn high number of possible complex events that occur in daily life. TRECVID's MED task has been a competitive challenge for recent years.

For the task, we make use the idea of *prototypes*. The *prototypes* are high level models that we use to learn sub-concepts of an event to model the events. Initial prototypes are extracted based on clustering. We also make use of the segments of a video in two ways. The first one is the *snippet* idea where each segment has the same length and the second idea is the *shot* idea where the segments are consisted of scenes that differ from each other. The difference of scenes are computed based on color histograms. We present four methods that use the prototypes on *snippets* or *shots*. The first two method, *Cluster Similarity Histograms* and *Cluster Id Histograms*, directly use the similarity information of segments to initial *prototypes*. The other two methods make use of the SVM models for feature creation and uses the initial *prototypes* to create improved *prototypes*. The proposed methods are still comparable with the highly computational methods proposed in MED task.

One of the other challenges in TRECVID is the SIN task where we make use of images from an image search engine to model the concepts. For each 60 concept in the dataset, we create binary-SVM models from the collected web concept image set and compared the quality of models for prediction with using different number of images from the ranked image list.

To increase the quality of our models we re-rank the images based on a *Multiple Instance Learning* algorithm and experiments are done with different number of images from the re-ranked image list. iAP results are decreased comparing to the results we have by using the original image ranking list of the search engine.

Bibliography

[1] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *ECCV*, 2012.

[2] A. Jain, A. Gupta, M. Rodriguez, and L. Davis, "Representing videos using mid-level discriminative patches," in *CVPR*, 2013.

[3] Y. Liu, D. Xu, I. W. Tsang, and J. Luo, "Using large-scale web data to facilitate textual query based retrieval of consumer photos," in *ACM MM*, 2009.

[4] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *Int. J. Comput. Vision*, vol. 106, pp. 210–233, 2014.

[5] J. Tang, Q. Chen, S. Yan, T.-S. Chua, and R. Jain, "One person labels one million images," in *ACM MM*, 2010.

[6] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *CVPR*, 2012.

[7] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IROS*, 2009.

[8] V. Ramanathan, P. Liang, and L. Fei-Fei, "Video event understanding using natural language descriptions," in *ICCV*, pp. 905–912, 2013.

[9] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot, "Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *TRECVID 2013*, 2013.

[10] R. Aly, R. Arandjelovic, K. Chatfield, M. Douze, B. Fernando, Z. Harchaoui, K. McGuinness, N. E. OConnor, D. Oneata, O. M. Parkhi, D. Potapov, J. Revaud, C. Schmid, J. Schwenninger, D. Scott, T. Tuytelaars, J. Verbeek, H. Wang, and A. Zisserman, "Axes at trecvid 2013," in *TRECVID*, vol. 2013, (Gaithersburg, MD, USA), 2013.

[11] P. Natarajan, P. Natarajan, S. Wu, X. Zhuang, A. Vazquez-Reina, S. Vitaladevuni, K. Tsourides, C. Andersen, R. Prasad, G. Ye, D. Liu, S.-F. Chang, I. Saleemi, M. Shah, Y. Ng, B. White, L. Davis, A. Gupta, and I. Haritaoglu, "Bbn viser trecvid 2012 multimedia event detection and multimedia event recounting systems," in *TRECVID*, 2011.

[12] U. Niaz, M. Redi, C. Tanase, B. Merialdo, G. Farinella, and Q. Li, "EURECOM at TRECVID 2011: The light semantic indexing task," in *TRECVID*, (Gaithersburg, UNITED STATES), 2011.

[13] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, 2010.

[14] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *CVPR*, 2004.

[15] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *IJCV*, 2007.

[16] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *ICCV*, 2011.

[17] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *IEEE Transactions on PAMI*, vol. 28, pp. 1931–1947, 2006.

[18] F. Sener and N. Ikizler-Cinbis, "Ensemble of multiple instance classifiers for image re-ranking," *Image Vision Comput.*, vol. 32, no. 5, pp. 348–362, 2014.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, vol. 2, pp. 1470–1477, 2003.

[21] K. van de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, 2010.

[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[23] H. Li, L. Bao, Z. Gao, A. Overwijk, W. Liu, L. fei Zhang, S. i Yu, M. yu Chen, F. Metze, and E. Hauptmann, "Trecvid," 2010.

[24] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, (Washington, DC, USA), pp. 32–36, 2004.

[25] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *ICCV*, (Washington, DC, USA), pp. 166–173, 2005.

[26] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *ICCCN*, (Washington, DC, USA), pp. 65–72, 2005.

[27] J. Wang, Z. Chen, and Y. Wu, "Action recognition with multiscale spatio-temporal contexts," in *CVPR*, (Washington, DC, USA), pp. 3185–3192, 2011.

[28] J. Liu, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *CVPR*, (Washington, DC, USA), pp. 3681–3688, IEEE Computer Society, 2012.

[29] H. Li, L. Bao, Z. Gao, A. Overwijk, W. Liu, L. fei Zhang, S. i Yu, M. yu Chen, F. Metze, and E. Hauptmann, "Informedia @ trecvid 2010."

[30] J. Yuen and A. Torralba, "A data-driven approach for event prediction," in *ECCV*, 2010.

[31] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *IJCV*, 2013.

[32] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, (Berlin, Heidelberg), pp. 143–156, Springer-Verlag, 2010.

[33] G. Csurka and F. Perronnin, "Fisher vectors: Beyond bag-of-visual-words image representations," in *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, pp. 28–42, Springer, 2011.

[34] J. a. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling," in *ECCV*, (Berlin, Heidelberg), pp. 430–443, Springer-Verlag, 2012.

[35] M. Douze, A. Ramisa, and C. Schmid, "Combining attributes and fisher vectors for efficient image retrieval," in *CVPR*, (Washington, DC, USA), pp. 745–752, 2011.

[36] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013.

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

[38] M. J. Roshtkhari and M. D. Levine, "Online dominant and anomalous behavior detection in videos," in *CVPR*, 2013.

[39] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *CVPR*, 2011.

[40] X. Sun, H. Yao, R. Ji, X. Liu, and P. Xu, "Unsupervised fast anomaly detection in crowds," in *ACM MM*, 2011.

[41] K. Kitani, B. D. Ziebart, J. A. D. Bagnell, and M. Hebert, "Activity forecasting," in *ECCV*, 2012.

[42] J. Revaud, M. Douze, C. Schmid, and H. Jégou, "Event retrieval in large video collections with circulant temporal encoding," in *CVPR*, 2013.

[43] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?," *ACM SIGGRAPH*, vol. 31, no. 4, 2012.

[44] F. Z. Wen-Sheng Chu and F. de la Torre, "Unsupervised temporal commonality discovery," in *ECCV*, 2012.

[45] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR'06*, (New York, NY, USA), pp. 321–330, ACM Press, 2006.

[46] Z.-Z. Lan, L. Jiang, S.-I. Yu, S. Rawat, Y. Cai, C. Gao, S. Xu, H. Shen, X. Li, Y. Wang, *et al.*, "Cmu-informedia@ trecvid 2013 multimedia event detection," in *TRECVID*, 2013.

[47] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li, "Object bank: A high-level image representation for scene classification semantic feature sparsification," in *NIPS*, pp. 1378–1386, 2010.

[48] J. Hays and G. Patterson, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," *CVPR*, 2012.

[49] Y. Zhao, B. Gan, S. Tang, J. Liu, X. Li, Y. Li, Q. Qu, X. Yang, and L. Zhang, "Bit@ trecvid 2013: Surveillance event detection,"

[50] L. Brown, L. Cao, Y. Cheng, A. Choudhary, N. Codella, Q. Fan, R. Feris, L. Gong, M. Hill, G. Hua, M. Merler, S. Pankanti, and J. R. Smith, "Ibm research and columbia university trecvid-2013 multimedia event detection (med) system."

[51] S. Little, I. Jargalsaikhan, R. Albatal, C. Direkoglu, N. E. O?Connor, A. F. Smeaton, K. Clawson, M. Jing, B. Scotney, H. Wang, J. Li, M. Nieto, J. D. Ortega, A. Rodriguez, I. Aramburu, and E. Kafetzaki, "Savasa project @ trecvid 2013: Semantic indexing and interactive surveillance event detection," 2013.

[52] N. Ballas, B. Labbé, H. Le Borgne, P. Gosselin, M. Redi, B. Merialdo, R. Vieux, B. Mansencal, J. Benois-Pineau, S. Ayache, A. Hamadi, B. Safadi, T.-T.-T. Vuong, H. Dong, N. Derbas, G. Quénot, B. Gao, C. Zhu, Y. Tang, E. Dellandrea, C.-E. Bichot, L. Chen, A. Benoît, P. Lambert, and T. Strat, "IRIM at TRECVID 2013: Semantic Indexing and Instance Search," 2013.

[53] H. Bai, Y. Dong, S. Cen, L. Wang, L. Liu, W. Liu, Y. Bian, C. Huang, N. Zhao, B. Liu, *et al.*, "Orange labs beijing (ftrdbj) at trecvid 2013: Instance search,"

[54] U. Niaz, M. Redi, C. Tanase, and B. Merialdo, "EURECOM at TrecVid 2012: The light semantic indexing task," in *TRECVID*, 2012.

[55] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, 1997.

[56] O. Maron and T. Lozano-Prez, "A framework for multiple-instance learning," in *Advances In Neural Information Processing Systems*, pp. 570–576, 1998.

[57] Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique," in *In Advances in Neural Information Processing Systems*, pp. 1073–1080, 2001.

[58] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *NIPS*, pp. 561–568, 2003.

[59] J. Wang and J.-D. Zucker, "Solving the multiple-instance problem: A lazy learning approach," in *ICML*, pp. 1119–1125, 2000.

[60] J. Ramon and L. De Raedt, "Multi instance neural networks," in *ICML*, pp. 53–60, 2000.

[61] Z. H. Zhou and M. L. Zhang, "Neural networks for multi-instance learning," tech. rep., ICIIP, 2002.

[62] Y. Chevaleyre and J.-D. Zucker, "Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. application to the mutagenesis problem," in *Lecture Notes in Artificial Intelligence*, vol. 2056, pp. 204–214, 2001.

[63] H. Blockeel, D. Page, and A. Srinivasan, "Multi-instance tree learning," in *ICML*, (New York, NY, USA), pp. 57–64, 2005.

[64] T. Deselaers and V. Ferrari, "A conditional random field for multiple-instance learning," in *ICML*, 2010.

[65] Z. Fu, A. Robles-Kelly, and J. Zhou, "Milis: Multiple instance learning with instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 958–977, 2011.

[66] J. Amores, "Multiple instance classification: Review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, pp. 81 – 105, 2013.

[67] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, pp. 1–25, 2010.

[68] Z.-J. Zha, X.-S. Hua, T. Mei, J. Wang, G.-J. Qi, and Z. Wang, "Joint multi-label multi-instance learning for image classification," in *CVPR*, pp. 1–8, June 2008.

[69] O. Y. an Vasant Honavar, "Multi-instance multi-label learning for image classification with large vocabularies," in *BMVC*, 2011.

[70] Y. Chen and J. Z. Wang, "Image categorization by learning and reasoning with regions," *J. Mach. Learn. Res.*, vol. 5, pp. 913–939, Dec. 2004.

[71] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *NIPS*, pp. 1419–1426, MIT Press, 2006.

[72] P. Dollár, B. Babenko, S. Belongie, P. Perona, and Z. Tu, "Multiple component learning for object detection," in *ECCV*, 2008.

[73] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR*, 2009.

[74] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof, "On-line semi-supervised multiple-instance boosting," in *BMVC*, June 2010.

[75] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Text-based image retrieval using progressive multi-instance learning," in *ICCV*, 2011.

[76] L. Duan, W. Li, I. W. Tsang, and D. Xu, "Improving web image search by bag-based re-ranking," *IEEE Trans. on Image Processing*, pp. 3280–3290, 2011.

[77] Q. Zhang, S. A. Goldman, W. Yu, and J. E. Fritts, "Content-based image retrieval using multiple-instance learning," in *ICML*, pp. 682–689, 2002.

[78] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," SCIA, 2003.

[79] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *BMVC*, 2012.

[80] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM TIST*, 2011.

[81] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[82] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *ECCV*, 2006.

[83] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC*, pp. 995–1004, sep 2008.

[84] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *ICCV*, 2005.

[85] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang, "Noise resistant graph ranking for improved web image search," in *CVPR*, pp. 849–856, 2011.

[86] S. Li, "Image search results refinement via outlier detection using deep contexts," in *CVPR*, 2012.