Human Action Recognition from Video Sequences

Umakanthan Sabanadesan

B.Sc Eng (Hons, 1st Class)

PhD Thesis

Submitted in Fulfilment

of the Requirements

for the Degree of

Doctor of Philosophy

Queensland University of Technology

Speech, Audio, Image and Vision Research Laboratory

Science and Engineering Faculty

2016

Substitute the "official" thesis signature page here.

Keywords

Action Recognition, Human Motion Analysis, Video Surveillance, Bag-of-features, SVM Classification, Local Spatio-temporal Features, Sparse-representation, LDA representation, Multiple instance dictionary learning.

Abstract

Human motion analysis is currently receiving increasing attention from computer vision researchers. This interest is motivated by applications over a wide spectrum of topics. For example, segmenting the parts of the human body in an image, tracking the movement of joints over an image sequence, and recovering the underlying 3D body structure are particularly useful for analysis of athletic performance, as well as medical diagnostics. The capability to automatically monitor human activities using computers in security-sensitive areas such as airports, border crossings, and building lobbies is of great interest to the police and military. With the development of digital libraries, the ability to automatically interpret video sequences will save tremendous human effort in sorting and retrieving images or video sequences using content-based queries. Other applications include building man-machine user interfaces and video conferencing.

The research trend in the field of action recognition has recently led to more robust techniques, which to some extent are applicable for action recognition in complex scenes. Action recognition in complex scenes is an extremely difficult task due to challenges such as background clutter, camera motion, occlusions and illumination variations. To address these challenges, several methods, like tree-based template matching, tensor canonical correlation, prototype based action matching, incremental discriminant analysis of canonical correlation, latent pose estimation and a generalised Hough transform were proposed. Most of these

methods are very complex and require preprocessing, like segmentation, tree data structure building, target tracking, background subtraction or the fitting of a human body model. On the other hand, recently, spatio-temporal features have gained popularity because of their state-of-the-art performance with reduced or even no preprocessing. These methods apply interest point detectors and local descriptors to characterize and encode the video data, and thereby perform action classification. In this PhD program, local feature based action representation, recognition and classification algorithms are explored due to their superior state-of-the art performance under complex environmental settings with lower preprocessing, compared to other approaches.

Even though local feature-based methods have been researched by several researchers for more than a decade, these systems still have several limitations and far-from-real time implementations. The performance of local-feature based systems depends on three major areas: (1) Accurate Representation of video sequences as a set of feature vectors (Feature Extraction), (2) Reducing the dimensionality of the feature points to create compact representation of the video (Feature Representation), (3) Train the classifier to classify new video sequences (Classification). This thesis has investigated the above three major areas of a local feature-based action-recognition pipeline and has proposed several improvements to the overall system accuracy.

In order to address the shortcomings of the action recognition pipeline, first baseline system using the bag of visual words with SVM framework has been implemented. Several state-of-the-art spatio-temporal features, such as HOG, HOF and HOG3D features, have been extracted and tested against popular benchmark datasets. A comprehensive evaluation of state-of-the-art descriptors has been undertaken with a wide range of code book sizes.

In order to address the video representation problem, an efficient feature represen-

tation method, semi-binary features based on BRISK (Binary Robust Invariant Scalable Keypoints) descriptor, has been proposed. Because of the binary nature of this feature it provides compact representation while maximizing the overall classification performance on several benchmark datasets.

In order to provide efficient and compact feature representation, several popular machine learning techniques have been explored and three new representation techniques have been incorporated based on class-specific dictionaries. It has been found that class-specific dictionaries consistently perform well and three new machine learning techniques, such as Multiple instance dictionary learning, Class-specific simplex LDA (css-LDA) and class-specific sparse codes, have been incorporated to the action recognition domain. These representation methods have improved the overall performance of popular local feature descriptors.

Finally, to address the classification phase of the action recognition pipeline, a binary-tree SVM has been proposed. The proposed binary-tree SVM achieves comparable state-of-the-art performance with a significantly reduced computational complexity and can be easily scalable to large datasets.

Though the techniques proposed in this thesis achieve promising results compared to the state-of-the-art, further research effort is required to achieve comparable performance in more challenging environments that are encountered in practice. The limitations of the proposed techniques are discussed, together with possible future extensions.

Contents

List of Tables

List of Figures

List of Acronyms

Average Precision	AP
Bag Of Features	BoF
Bag-of-Visual-Words	BOVW
Binary Robust Invariant Scalable Keypoints	BRISK
Binary Robust Independent Elementary Features	BRIEF
Class Specific Simplex LDA	CSS-LDA
Expectation Maximization	EM
Extended SURF	ESURF
Features from Accelerated Segment Test	FAST
Fisher Discriminant Analysis	FDA
Fast Retina Keypoint	FREAK
Gaussian Mixture Models	GMM
Hidden Markov Models	HMM
Histograms of Optical Flow	HOF
Histograms of Oriented Gradients	HOG
K-Nearest Neighbour	KNN
Latent Dirichlet Allocation	LDA
Locality Constrained Linear Coding	LLC
Markov Chain Monte Carlo	MCMC
Max Margin Dictionary Learning	MMDL
Max Margin Multiple Instance Clustering	M^3IC
Max-margin Multiple instance clustering	M^3IC
Max-margin Multiple Instance Dictionary Learning	MMDL
Maximum Entropy Discrimination LDA	MEDLDA
Mean Average Precision	mAP
Motion Boundary Histogram	MBH
Motion Energy Images	MEI
Motion History Images	MHI
Multiple Instance Clustering	MIC
Multiple Instance Learning	MIL
Multiple Instance Learning	MIL
Multiple instance learning	MIL
Multiple Instance SVM	mi-SVM
Multiple Instance Multi Label Learning	MIML
Multiple Instance Single Label	MISL

Oriented fast and Rotated BRIEF	ORB
Principal component analysis	PCA
Probabilistic Latent Semantic Analysis	PLSA
Probabilistic Latent Semantic Indexing	PLSI
Radial Basis Function	RBF
Scale Invariant Feature Transformation	SIFT
Single Instance Multi Label	SIML
Single Instance Single Label	SISL
Single Instance Single Label Learning	SISL
Sparse Coding	SC
Sparse coding Spatial Pyramid Matching	ScSPM
Sparse Representation-based Classification	SRC
Spatial Pyramid Matching	SPM
Spatio Temporal Interest Points	STIP
Supervised Latent Dirichlet Allocation	LDA
Supervised LDA	sLDA
Support Vector Machine	SVM
Vector Quantization	VQ

Certification of Thesis

The work contained in this thesis has not been previously submitted for a degree or diploma at any other higher educational institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signed:

Date: 13/02/2016

Acknowledgments

I would firstly like to thank my principal supervisor, Professor Clinton Fookes, for his consistent research guidelines, discussions and support throughout this program. I would also like to express my deep gratitude to my associate supervisor, Professor Sridha Sridharan, for assisting me throughout the PhD application process, giving me the PhD opportunity, financial support for the past several years and guidance throughout this program.

I also extend my gratitude to associate supervisor, Dr Simon Denman. Dr Simon has provided a lot of suggestions, discussions, guidelines and technical support. He also made a lot of contributions to improving the conference papers and thesis contents, as well as setting the HPC and software development platform. This PhD program is partly supported by CSIRO and I want to extent my gratitude to Dr Tim Wark, for his supervision, guidance and support.

I would also like to thank everyone in the Speech, Audio, Image and Video Technology (SAIVT) laboratory for their friendship and assistance. I would also like to acknowledge the assistance given by QUT administrative staff in the progress of my thesis. Professional Editor, Ms Diane Kolomeitz (Editorial Services), provided copy editing and proofreading services, according to the guidelines laid out in the University-endorsed national policy guidelines, for the editing of research theses.

Finally, I would also like to thank my family for their support throughout my PhD journey.

Umakanthan Sabanadesan

Queensland University of Technology

2016

Chapter 1

Introduction

Human activity recognition is an important area of computer vision research today. The goal of human activity recognition is to automatically analyse ongoing activities from an unknown video (i.e. a sequence of image frames). In a simple case where a video is segmented to contain only one execution of a human activity, the objective of the system is to correctly classify the video into its activity category. In a more general case, the continuous recognition of human activities must be performed by detecting starting and ending times of all occurring activities from an input video.

The ability to recognise complex human activities from videos enables the construction of several important applications. Automated surveillance systems in public places like airports and subway stations require detection of abnormal and suspicious activities, as opposed to normal activities. For instance, an airport surveillance system must be able to automatically recognize suspicious activities like "a person leaving a bag" or "a person placing his/her bag in a trash bin". Recognition of human activities also enables the real-time monitoring of patients, children, and elderly persons. The construction of gesture-based human

computer interfaces and vision-based intelligent environments becomes possible with an activity recognition system as well.

There are various types of human activities. Depending on their complexity, they can be conceptually categorized into four different levels: gestures, actions, interactions, and group activities [3]. Gestures are elementary movements of a persons body part, and are the atomic components describing the meaningful motion of a person. "Stretching an arm" and "raising a leg" are good examples of gestures. Actions are single-person activities that may be composed of multiple gestures organized temporally, such as "walking", "waving", and "punching". Interactions are human activities that involve two or more persons and/or objects. For example, "two persons fighting" is an interaction between two humans and "a person stealing a suitcase from another" is a human-object interaction involving two humans and one object. Finally, group activities are the activities performed by conceptual groups composed of multiple persons and/or objects: "A group of persons marching," "a group having a meeting," and "two groups fighting" are typical examples. In this research, the main focus is given to improve the recognition accuracy of single human activities from real-time video sequences.

Nowadays, more and more people record their daily activities using digital cameras, and this brings the enrichment of video content on the internet, and also causes the problems of categorizing the existing video, and classifying new videos according to the action classes present. Categorizing these videos is a time-consuming task if it is done manually, and recognizing certain actions from scenes of interest in real movies is impossible to accomplish through manual effort. For these reasons, the area of human action recognition has attracted considerable attention. Existing approaches aimed at solving this problem have focused on a pattern recognition system, which is trained using feature descriptors extracted from the training videos, and enables the computer to identify the actions in

new videos automatically. The objective of this thesis is to present several novel approaches in feature extraction, representation and classification to improve the popular, widely used, local feature-based action-recognition system (Bag-of-visual words with SVM Framework).

1.1 Research Motivation

The development of computer vision has encouraged the occurrence of different novel recognition methods in both 2D images and 3D video sequences. Although it is still challenging to recognize a specific object from a dataset of images due to viewpoint change, illumination, partial occlusions, and intra-class difference and so forth, many successful methods have been proposed. But for the video recognition problem, the current methods still need improvement, especially for realistic movies which have wide variations in people's posture and clothes, dynamic background, and partial occlusions. Intuitively, a straightforward way is comparing an unknown video with the training samples by computing correlation between the whole videos. This approach makes good use of geometrical consistency, but it is not feasible when dealing with camera motions, zooming, intra-class differences and non-stationary backgrounds.

In fact, action recognition has become one of the hottest research areas in computer vision and impressive progress has been made in this direction. However, progress is primarily limited to a controlled experimental environment, which may lead to difficulties when we move to recognising and analysing actions in more realistic scenarios. To understand the possible difficulties, let us first examine some assumptions which have been made in traditional action recognition (i.e. controlled environment):

1. **Preprocessing Assumption:** For a computer vision problem, choosing appropriate visual features and representation is the first step to solving the problem. In most cases, the feature extraction requires some preprocessing steps. In action recognition, this preprocessing step can be the detection and tracking of body parts or a moving person, or the segmentation of the region of interest. However, if these preprocessing steps fail, the methods based on them will breakdown.

2. **Data Assumption:** Most action recognition systems are based on statistical machine learning methods, which learn a classifier from a set of training data. In the usual case, sufficient labelled training data is assumed to be available. However, when the labelled training data is insufficient or unavailable or the data can only be obtained from more complex settings, say, from an ambiguously annotated dataset, the system structure of the training process will need to be changed accordingly.

3. **Model Assumption:** To mathematically model an action, we often make the assumption that an action can be viewed as an equivalent simplified vision/machine learning problem. For example, if an action is represented by a set of silhouettes, an underlying assumption is that an action can be characterized by the temporal evolution of 2D shapes. If we model an action by a bag of local features, we assume that an action can be characterized by the orderless local spatial temporal patterns. Of course, the assumption does not always hold in many applications.

In practical action recognition problems, one or more aforementioned assumptions will not hold. Let us consider the following examples: when we try to classify action in the presence of a dynamic background, the foreground segmentation or reliable bounding box detection and tracking is often not available; when we try to retrieve an action in video, to detect an unusual action, or to discover action

categories from a set of videos, the assumption that sufficient labelled training data is available does not hold. Most daily actions are more complex than simple body movements e.g. boxing, hand waving. The interaction between object, environment and many other cues is often as important as the motion patterns. So the common assumption that action is equivalent to body movement is not enough for modelling more complex actions.

To conquer these deficiencies, a lot of researchers focus on part-based approaches for which only the 'interesting' parts of the video are analyzed, rather than the whole video. These 'parts' can be trajectories or flow vectors of corners, profiles generated from silhouettes and spatial temporal interest points. Although part based approaches are promising they are still suffered due to background clutter and motion which prevents from accurate detection and tracking of interesting parts. Meanwhile recently proposed local feature based approaches extract interesting points based on the motion information present in the videos, which makes them more robust to background motion, clutter, viewpoint changes compared to other approaches. Moreover, this research is particularly interested in the case when an action is represented by a set of local spatial-temporal features due to following reasons:

1. As will be seen in the literature review, this representation is more robust to pose and view variance.
2. This representation can impose a relaxed requirement on the bounding box detection and tracking (and can even work without it).
3. It is more flexible to model the local interactions between multiple features by using a local spatial-temporal feature-based representation.

Although local feature based approaches are promising due to many advantages,

still the recognition rate is constrained due to the inefficient and unreliable description and classification methods. The aim of this PhD research program is to address three major shortfalls such as lack of spatio-temporal relationship, scalability and computational complexity by proposing several novel and effective description and classification methods. The contributions made in this thesis will make the application of local feature based methods to be more scalable and computationally effective.

1.2 Research Objective and Scope

This thesis considers recognizing simple human activities from video sequences recorded under different environmental conditions varying from a fixed, clean background to complex, cluttered and moving backgrounds. A wide range of human activities have been investigated in this research from single person activities such as running, walking, jogging *etc.* to complex activities such as fight with person, get out of car, hugging *etc.*

A number of methods have been proposed over the past 30 years in action recognition research. Earlier approaches were focused on the appearance, and heavily related to the entire silhouette extraction and modelling the action as a sequence of changes over time using Hidden Markov Models (HMM). More recent research has focused primarily on model-free approaches such as bag-of-words. The details of these methods are described in Chapter 2. A local spatio-temporal based action recognition system typically consists of the fundamental tasks as shown in Figure 1.1.

In this thesis, due to its simplicity and superior performance, a local feature based action recognition system is incorporated as a baseline and several novel feature

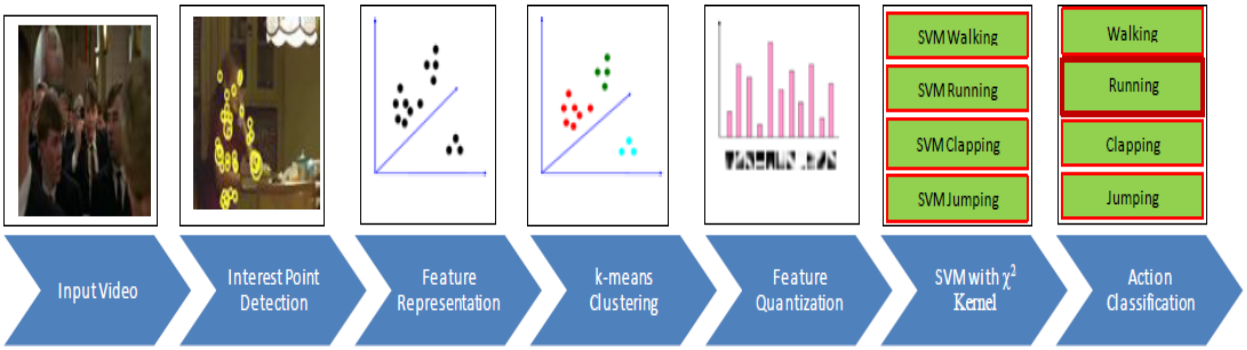


Figure 1.1: The Spatio-temporal Action recognition framework.

extraction, representation and classification techniques have been proposed to improve the overall performance. The proposed approaches are primarily evaluated with datasets specifically designed for human action recognition. In order to provide a fair comparison, the proposed methods have been investigated with popular features and datasets, in order to enable the easy benchmarking of the proposed techniques with past and future developments. Even though these techniques have been developed primarily for human action recognition, they are not limited to this domain and can be extended to other video- based computer vision applications as well. This thesis has used several challenging, publicly available datasets designed for human action recognition, which are still very challenging in the field and highlight the ample ongoing room for improvement.

1.3 Thesis structure

The remaining chapters of the thesis are organized as follows:

- **Chapter 2** provides an overall review of the literature. In this chapter, 30 years of evolution of human action recognition is briefly presented. This section provides an introduction to different approaches, different features

extraction, representation and classification techniques used by researchers over the last three decades. In addition, a comprehensive review of popular, challenging datasets and their evaluation metrics is also presented. A detailed review of a popular, local feature-based action recognition system is also presented, which is the main focus of this thesis and justification for the selection also presented.

- **Chapter 3** presents a detailed overview of Bag-of-feature based action recognition systems and their development over time with a comprehensive evaluation of how different stages in the pipeline affect performance. Popular local feature detectors and descriptors are presented with different classification schemes. In addition, parameters are optimized for different datasets in such a way as to improve the performance significantly with the existing features. This chapter will provide guidance to researchers to make decisions regarding different encoding approaches, codebook sizes, kernel matrices and spatio-temporal pyramids.

- **Chapter 4** introduces a new binary detector/descriptor, BRISK, to efficiently represent the video. In this chapter, the binary BRISK detector is extended into video domain to detect interest points followed by a new algorithm to select potential spatio-temporal points based on their significance. Then BRISK + MBH (Motion Boundary Histogram) descriptor is used to encode the detected key points. This proposed feature detector and descriptor combination is not only efficient but also demonstrates comparative performance in benchmark datasets.

- **Chapter 5** presents another spatio-temporal feature representation based on Multiple Instance Learning (MIL) techniques. MIL has gained popularity amongst machine learning researchers and in this chapter several MIL techniques such as 'miSVM + kmeans', Max-margin Multiple instance Dictionary learning (MMDL) and Max-margin Multiple instance cluster-

ing (M^3IC) are introduced to create effective feature representation. Experimental results are presented to demonstrate the effectiveness of this representation.

- **Chapter 6** presents a new feature representation technique based on Supervised Latent Dirichlet Allocation (LDA) techniques such as S-LDA and MedLDA. Also this chapter presents another efficient LDA technique, css-LDA, where topics are discovered class-by-class basic rather than a single topic simplex for the entire dataset. It is shown from the experiments that this representation is far more efficient than original unsupervised LDA and Bag-of-feature representation. A detailed evaluation is also presented with different LDA approaches in this chapter.

- **Chapter 7** investigates several sparse representation techniques and proposes a novel appearance and motion specific dictionary to encode features as a sparse coefficient vector. This separate motion and appearance dictionary significantly improves the performance compared to a single sparse-dictionary build for the entire dataset.

- **Chapter 8** addresses the classification problem by proposing a binary tree SVM to address the shortcomings of multi-class SVMs in activity recognition. This chapter also presents a new method of constructing a binary tree using Gaussian Mixture Models (GMM), where activities are repeatedly allocated to sub-nodes until every newly created node contains only one activity. Then, for each internal node a separate SVM is learned to classify activities. This approach reduces the training time and increases the speed of testing compared to popular the 'one-against-the-rest' multi-class SVM classifier.

- **Chapter 9** summarizes and concludes the thesis, highlights the achievements, addresses the limitations, and points to future research directions.

1.4 Original Contributions

This thesis has contributed several advances to the field of local feature-based activity recognition, by addressing several challenges. The popular, state-of-the art local feature based activity recognition system was built and the following novel techniques have been proposed to improve the overall performance of the system. The framework of the Bag-of-feature based SVM classification system is detailed in Chapter 3.

1. A comprehensive evaluation on several popular local feature detectors and descriptors is carried out with three challenging datasets. In this evaluation, several encoding techniques, codebook sizes and different kernel learning techniques have been investigated and optimized techniques have been proposed. This provides a guide for researchers to choose appropriate techniques based on the complexity of the dataset.

2. A novel semi-supervised binary feature is introduced to efficiently represent videos for the purpose of activity classification. In this proposed framework, first, the BRISK feature detector is applied on a frame-by-frame basis to detect interest points, then the detected key points are compared against consecutive frames for significant motion. Amongst the detected points, only the points with significant motion are retained. Then the retained key points are encoded with the BRISK descriptor in the spatial domain and Motion Boundary Histogram in the temporal domain. This descriptor is not only lightweight but also has lower memory requirements because of the binary nature of the BRISK descriptor, allowing the possibility of applications using hand-held devices or for other resource-constrained and real-time applications.

3. Two new, supervised LDA variants, MedLDA and css-LDA are introduced

in a local feature-based action recognition system to efficiently represent videos. MedLDA extends LDA to learn discriminative topics by employing a max-margin technique within the probabilistic framework. On the other hand, css-LDA introduces the supervision at the feature level and enables class specific topic simplexes and class-specific topic distributions to capture much richer intra-class information, which provides more discrimination to the representation compared to a single set of topics for the entire data set.

4. A novel feature representation technique based on Multiple Instance Learning (MIL) is proposed for the local feature-based action recognition framework. In this proposed approach, the k-means clustering is replaced with three MIL based feature representation techniques such as 'mi-SVM + k-means', M^3IC and MMDL. The proposed three representations provide highly discriminative feature representation compared to bag-of-features and significantly improve the classification accuracy. Unlike the k-means approach where k-means is applied in the entire feature set, in 'mi-SVM + k-means' approach the k-means is applied only on the positive features identified by SVM. In addition, dictionaries are built on a class-by-class basis in 'mi-SVM + k-means' and MMDL approaches as opposed to a single shared dictionary across the dataset. In the M^3IC approach, the MIL technique is used during code-book generation.

5. A new sparse representation based on class-specific appearance and motion over-complete dictionary is proposed to encode video features for discriminative classification. In this approach, separate dictionaries are built for appearance and motion vectors and then a block-structured dictionary is constructed to encode features as a sparse linear combination of a block-structured dictionary. This approach is shown to be effective, compared to shared and class-specific dictionaries. In addition, separate appearance and motion dictionaries explore different statistical characteristics captured by

appearance and motion features. It is also shown that as we go further into detail designing the sparse dictionary, the discriminative ability increases.

6. A Binary-Tree SVM is proposed to boost the speed of the classification stage in the local feature-based action recognition pipeline mentioned earlier. In this approach, training samples are assigned to the root node of the tree and a GMM is used to separate the training samples into two clusters, and the activities belonging to each cluster are assigned to the left and right sub-nodes respectively. In the training phase, it requires only $N-1$ SVMs to be trained for an N class problem; the amount of time required for training also reduces as the tree is traversed downwards as the number of classes (and amount of data) at each node is reduced. When performing classification, the proposed approach requires only $log_2 N$ SVMs to predict the sample due to the binary nature of the decision tree.

1.5 Publications

Listed below are the peer-reviewed publications resulted from this research programme.

Peer-reviewed international conferences

1. **Umakanthan Sabanadesan**, Denman Simon, Fookes Clinton B., & Sridharan Sridha. **Class specific sparse codes for representing activities** In Proceedings of the 2015 International Conference on Image Processing (ICIP), IEEE, Quebec, Canada.

2. **Umakanthan Sabanadesan**, Denman Simon, Fookes Clinton B., & Sridharan Sridha **Supervised Latent Dirichlet Allocation models for Efficient Activity Representation**. In Proceedings of the 2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA), IEEE, Wollongong, Australia.

3. **Umakanthan Sabanadesan**, Denman Simon, Fookes Clinton B., & Sridharan Sridha **Multiple instance dictionary learning for activity representation**. In Nilsson, Mikael (Ed.) Proceedings of the 22nd International Conference on Pattern Recognition (ICPR 2014), IEEE, Stockholm, Sweden.

4. **Umakanthan Sabanadesan**, Denman Simon, Fookes Clinton B., & Sridharan Sridha **Activity recognition using binary tree SVM**. In Proceedings of the 2014 IEEE Workshop on Statistical Signal Processing (SSP), IEEE, Gold Coast, Australia, pp. 248-251.

5. **Umakanthan Sabanadesan**, Denman Simon, Fookes Clinton B., & Sridharan Sridha **Semi-binary based video features for activity repre-**

sentation. In Proceedings of the 2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA), IEEE, Wrest Point, Hobart, TAS, pp. 178-184.

6. **Umakanthan Sabanadesan**, Denman Simon, Sridharan Sridha, Fookes Clinton B., & Wark Tim **Spatio temporal feature evaluation for action recognition**. In Proceedings of The 2012 International Conference on Digital Image Computing Techniques and Applications (DICTA 12), IEEE, Fremantle, Western Australia, pp. 1-8.

Chapter 2

Literature Review

2.1 Introduction

This section reviews the state-of-the-art methods for action recognition in realistic, uncontrolled video data. To this end, we structure the existing works into three categories:

- **Human model-based methods** (Section 2.2) employ a full 3D (or 2D) model of human body parts, and action recognition is done using information on body part positioning as well as movements.
- **Holistic methods** (Section 2.3) use knowledge about the localization of humans in video and consequently learn an action model that captures characteristic, global body movements without any notion of body parts.
- **Local feature methods** (Section 2.4.1) are entirely based on descriptors of local regions in a video; no prior knowledge about human positioning nor any of its limbs/body parts is given.

Surveys on generic action and activity recognition, as well as motion analysis and body tracking, include Aggarwal *et al.* [3], Weinland *et al.* [101], Poppe *et al.* [74], Moeslund *et al.* [68], Moeslund and Granum [67], Gavrila [27] and Aggarwal and Cai [2]. Furthermore, Hu *et al.* [35] present a survey for video surveillance, and Turaga *et al.* [90] review the state-of-the-art for high level activity analysis. Most relevant in our context are the surveys by Aggarwal *et al.* [3], Weinland *et al.* [101] and Poppe *et al.* [74], which focus on the recognition of actions and action primitives, which are closely related to this research in human action recognition.

2.2 Human model-based methods

Human model-based methods recognize actions by employing information such as body part positions and movements. A significant amount of research is devoted to action recognition using trajectories of joint positions, body parts, or landmark points on the human body, with or without a prior model of human kinematics, e.g., [Ali *et al.* [5], Parameswaran and Chellappa [71], Yilmaz and Shah [109]].

The localization of body parts in movies has been investigated by Ramanan *et al.* [76] and Ferrari *et al.* [24]. However, the detection of body parts is a difficult problem in itself, and existing approaches, especially for the case of realistic and less constrained video data, remain limited in their applicability. Some recent approaches that are able to provide more robust results [1], use strong prior knowledge by assuming particular motion patterns in order to improve tracking of body parts. However, this also limits their application to action recognition.

2.3 Holistic methods

Holistic methods do not require the localization of body parts. Instead, global body structure and dynamics are used to represent human actions. Polana and Nelson [73] referred to this approach as "getting your man without finding his body part". The key idea is that, given a region of interest centred on the human body, global dynamics are discriminative enough to characterize human actions. Compared to approaches that explicitly use a kinematic model or information about body parts, holistic representations are much simpler, since they only model global motion and appearance information. Therefore their computation is, in general, more efficient as well as robust. This aspect is especially important for realistic videos in which background clutter, camera ego-motion and occlusion render the localization of body parts particularly difficult.

In general, holistic approaches can be divided into two categories.

- The first category employs shape masks or silhouette information, stemming from background subtraction or difference images, to represent actions.
- The second category is mainly based on shape and optical flow information.

2.3.1 Shape mask and silhouette based methods

Several approaches for action recognition use human shape masks and silhouette information to represent the human body and its dynamics.

Bobick and Davis [12] use shape masks from difference images to detect human actions. As action representation, the authors employ so-called motion energy images (MEI) and motion history images (MHI), as illustrated in Figure 2.1.

Figure 2.1: motion history images (MHI) and motion energy images (MEI) [12]. This can be viewed as a weighted projection of a 3-D XYT volume into 2-D XY Dimension

Figure 2.2: Space-time volumes for action recognition based on silhouette information [9]

More precisely, MEIs are binary masks that indicate regions of motion, and MHIs weight these regions according to the point in time when they occurred (the more recent, the higher the weight). This approach is the first to introduce the idea of temporal templates for action recognition.

Sullivan and Carlsson [89] detect tennis forehand strokes by matching a set of hand-drawn key postures, together with annotated body joint positions, to edge information in a video sequence. Positions of joints are then tracked between the key frames using silhouette information of the tennis player. This approach allows the positions of body parts to be inferred, which can be applied to animation.

An action model, based on space-time shapes from silhouette information, is introduced by Blank *et al.* [9]and Gorelick *et al.* [28]. Silhouette information is computed using background subtraction. Figure 2.2 illustrates some examples of

space-time shapes. The authors use the Poisson equation to extract features such as local saliency, action dynamics, shape structure and orientation. Sequences of 10 frames length are then described by a high-dimensional feature vector. During classification, these sequences are matched in a sliding window fashion to space-time shapes in test sequences.

Another work that uses space-time shapes of humans, is proposed by Yilmaz and Shah [110]. Spatio-temporal shapes are obtained from contour information using background subtraction, similar to Blank *et al.* [9]. For a robust representation, actions are then represented by sets of characteristic points (such as saddle, valley, ridge, peak, pit points) on the surface of the shape. In order to recognize actions, the authors propose to match spatio-temporal shapes by computing a homography using point-to-point correspondences.

Weinland and Boyer [100] introduce an orderless representation for action recognition using a set of silhouette exemplars. Action sequences are represented as vectors of minimum distance between silhouettes in the set of exemplars and in the sequence. Final classification is done using Bayes classifier with Gaussians to model action classes. In addition to silhouette information, the authors also employ the Chamfer distance measure to match silhouette exemplars directly to edge information in test sequences.

Foreground shape masks based on motion information in chunks of video data are employed by Zhang *et al.* [115], as shown in Figure 2.3 . A Motion Context descriptor is computed over consistent regions of motion by using a polar grid. Each cell in the grid is described with a histogram over quantized SIFT [53] features. The final descriptor for a sequence is a sum over all chunk descriptors. For classification, support vector machines (SVM) and different models for probabilistic latent semantic analysis (PLSA) are employed.

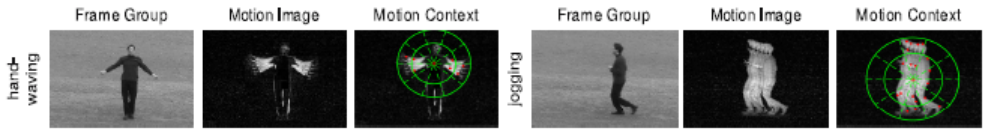


Figure 2.3: Motion Context descriptor for the actions hand waving and jogging: motion images are computed over groups of images; the Motion Context descriptor is computed over consistent regions of motion [115]

Silhouettes are also a popular representation for surveillance applications [35]. Since cameras are in general static, background subtraction techniques can be employed to compute silhouette information. In order to cope with more challenging video data and camera motion, Ramasso *et al.* [77] employ a human tracker and camera motion estimation to compute shape information.

Another way to match space-time shape models to cluttered image data with heterogeneous background is demonstrated by Ke *et al.* [38]. The authors over segment video sequences using colour information. Volumetric and optical flow features are then matched to action templates in the form of space-time shapes.

Silhouettes provide strong cues for action recognition. Nevertheless, they are difficult to compute in the presence of clutter and camera motion. Furthermore, they only describe the outer contours of a person and thus lack discriminative power for actions that include self-occlusions.

2.3.2 Optical flow and shape-based methods

Human-centric approaches based on optical flow and generic shape information form another sub-class of holistic methods. As one of the first works in this direction, Polana and Nelson [73] proposed a human tracking framework along with an action representation using spatio-temporal grids of optical flow magnitudes. The action descriptor is computed for periodic motion patterns. By matching against

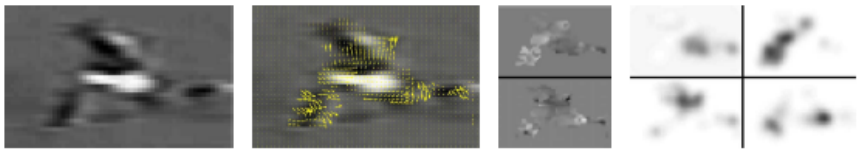


Figure 2.4: Motion descriptor using optical flow: (a) Original image, (b) Optical flow, (c)Separating the x and y components of optical flow vectors, (d) Half-wave rectification and smoothing of each component [21]

reference motion templates of known periodic actions (e.g., walking, running, swimming) the final action can be determined.

In another approach purely based on optical flow, Efros *et al.* [21] track soccer players in videos and compute a descriptor on the stabilized tracks using blurred optical flow. Their descriptor separates x and y flow as well as positive and negative components into four different channels, as shown in Figure 2.4. For classification, a test sequence is frame-wise aligned to a database of stored, annotated actions. Further experiments include tennis and ballet sequences as well as synthetic experiments.

The same human-centric representation based on optical flow and human tracks for action recognition is employed by Fathi and Mori [22]. As a classification framework, the authors use a two-layered AdaBoost variant. In the first step, intermediate features are learned by selecting discriminative pixel flow values in small spatio-temporal blocks. The final classifier is then learned from all previously aggregated intermediate features.

Rodriguez *et al.* [81] propose an approach using flow features in a template matching framework. Spatio-temporal regularity flow information is used as the feature. Regularity flow shows improvement over optical flow since it globally minimizes the overall sum of gradients in the sequence. Rodriguez *et al.* [81] learns cuboid templates by aligning training samples via correlation. For classifi-

cation, test sequences are correlated with the learned template via a generalized Fourier transform that allows for vectorial values. Results are demonstrated on the KTH dataset, for facial expressions, as well as on custom movie and sports actions.

To localize humans performing actions such as sit down, stand up, grab cup and close laptop, Ke *et al.* [37] use a forward feature selection framework and learn a classifier based on optical flow features. Spatio-temporal Haar features on optical flow components are efficiently computed using an integral video structure. During learning, a discriminative set of features are greedily chosen to optimally classify actions which are represented as spatio-temporal cuboidal regions. For classification, the authors perform a sliding window approach and classify each position as containing a particular action or not.

A method purely based on shape information is presented by Lu and Little [54]. In their experiments, Lu and Little track soccer or ice-hockey players and represent each frame by a descriptor using histograms of oriented gradients. They then employ principal component analysis (PCA) to reduce dimensionality. An HMM with a few states models actions such as running/skating left, right etc.

Hybrid representations combine optical flow with appearance information. Schindler and van Gool [85] use optical flow information and Gabor filter responses in a human-centric framework. For each frame, both types of information are weighted and concatenated. PCA over all pixel values is applied to learn the most discriminative feature information. Majority voting yields a final class label for a full sequence in multi-class experiments. Evaluations are carried out on the KTH and Weizmann dataset.

Human centric approaches require a method for localizing humans, therefore they rely intrinsically on the quality of human detections. To cope with imperfect

localizations from weakly labelled training data and an automatic human tracker, Hu *et al.* [34] introduce an approach based on multiple instance learning. In the neighbourhood around an annotated action or a human detection, a bag of possible action localization hypotheses is generated. An initial classifier is learned on all positive and negative instances. Iteratively, instances in bags are relabelled using the previously learned classifier and the classifier is retrained on the new data. Hu *et al.* [34] apply a simulated annealing strategy to ensure convergence. Feature types that are used are histograms of oriented gradients, foreground segmentation, and motion history images [12]. Results are presented on simple actions in crowded sequences as well as in more challenging data recorded in a shopping mall.

Even though holistic approaches have been shown to be suitable for action recognition in more realistic video data, certain points are important to note. Holistic representations are in general not invariant to camera view direction. This needs to be accounted for, either by learning different models for particular views (frontal, lateral, rear), or by providing a sufficiently large amount of training data. Additionally, humans can appear at different scales (distant view, close-up view) such that certain parts of the body might not be visible in the image. However, human localizations reduce the computational complexity of detecting actions in time substantially.

2.4 Local feature methods

Local image and video features have been successfully used in many action recognition applications such as object recognition, scene recognition and activity recognition. Local space-time features capture characteristic shape and motion information for a local region in video. They provide a relatively independent

representation of events with respect to their spatio-temporal shifts and scales as well as background clutter and multiple motions in the scene. These features are usually extracted directly from video and therefore avoid possible dependencies on other tasks such as motion segmentation and human detection.

In the following, we first discuss existing space-time feature detectors and feature descriptors. Methods based on feature trajectories are presented separately, since their conception differs from space-time point detectors. Finally, methods for localizing actions in videos are discussed.

2.4.1 Feature detectors

Feature detectors usually select characteristic spatio-temporal locations and scales in videos by maximizing specific saliency functions. Laptev [43] proposed a feature detector based on a spatio-temporal extension of the Harris cornerness criterion [31]. The cornerness criterion is based on the eigenvalues of a spatio-temporal second-moment matrix at each video point. Local maxima indicate points of interest. The authors note the importance of using separate spatial and temporal scale values since spatial and temporal extent of events are, in general, independent. Results of detecting Harris interest points in an outdoor image sequence of a person walking is illustrated in Figure 2.5.

Dollar *et al.* [19] argue that in certain cases, true spatio-temporal corner points (according to the Harris criterion) are relatively rare, while enough characteristic motion is still present in other regions. Therefore, they design their interest point detector to yield denser coverage in videos. Their method employs spatial Gaussian kernels and temporal Gabor filters. As with Harris 3D, local maxima give final interesting positions.

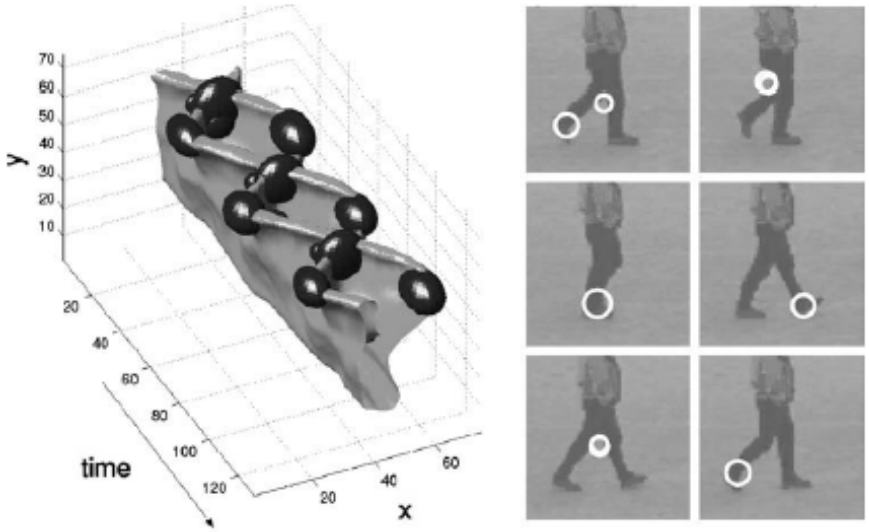


Figure 2.5: Spatio-temporal interest points from the motion of the legs of a walking person; (left) 3D plot of a leg pattern and the detected local interest points; (right) interest points overlaid on single frames in the original sequence [43]

A space-time extension of a salient region detector using entropy, is introduced by Oikonomopoulos *et al.* [70]. Entropy is computed in a cylindrical neighbourhood around a given space-time position for the temporal derivative of a video sequence. To obtain a sparse representation and more stable interest points, local maxima are thresholded and clustered.

The Hessian3D detector is proposed by Willems *et al.* [103] as a spatio-temporal extension of the Hessian saliency measure applied for blob detection in images [8]. The authors aim at a rather dense, scale-invariant, and computationally efficient interest point detector. Their detector measures saliency using the determinant of the 3D Hessian matrix. An integral video structure allows a speed up of computations by approximating derivatives with box-filter operations. A non-maximum suppression algorithm selects joint extrema over space, time and different scales.

Most feature detectors determine the saliency of a point with respect to its local neighbourhood. Wong and Cipolla [104] suggest determining salient features by considering global information. For this, video sequences are represented as a dynamic texture with a latent representation and a dynamic generation model. This not only allows motion to be synthesised, but also allows the identification of important regions in motion. The dynamic model is approximated as a linear transformation. A sub-space representation is computed via non-negative matrix factorization.

2.4.2 Feature descriptors

Feature descriptors capture shape and motion information in a local neighbourhood surrounding interest points. Among the first works on local descriptors for videos, Laptev and Lindeberg [44] develop and compare different descriptor types: single- and multi-scale higher-order derivatives (local jets), histograms of

optical flow, and histograms of spatio-temporal gradients. Histograms for optical flow and gradient components are computed for each cell of a M × M × M grid layout describing the local neighbourhood of an interest point. A different variant describes the surrounding of a given position by applying PCA to concatenated optical flow or gradient components of each pixel. The resulting descriptor uses the dimensions with the most significant eigenvalues. In their experiments, Laptev and Lindeberg [44] report best results for descriptors based on histograms of optical flow and spatio-temporal gradients.

In a similar work, Dollar *et al.* [19] evaluates different local space-time descriptors based on brightness, gradient, and optical flow information. They investigate different descriptor variants: simple concatenation of pixel values, a grid of local histograms, and a single global histogram. Finally, PCA reduces the dimensionality of each descriptor variant. Overall, concatenated gradient information yields the best performance.

Histograms of oriented spatial gradients (HOG) and Histograms of optical flow (HOF) descriptors are introduced by Laptev *et al.* [45]. To characterize local motion and appearance, the authors combine HOG and HOF in a late fusion approach. The histograms are accumulated in the space-time neighbourhood of detected interest points. Each local region is subdivided into a N × N × N grid of cells; for each cell, 4-bin HOG histogram and a 5-bin HOF histogram are computed. The normalized cell histograms are concatenated into the final HOG and HOF descriptors.

An extension of the SIFT descriptor [53] to 3D was proposed by Scovanner *et al.* [87]. For a set of randomly sampled positions, spatio-temporal gradients are computed in the local neighbourhood of each position. Each pixel in the neighbourhood is weighted by a Gaussian centred on the given position and votes into an M × M × M grid of histograms of oriented gradients. For orientation quan-

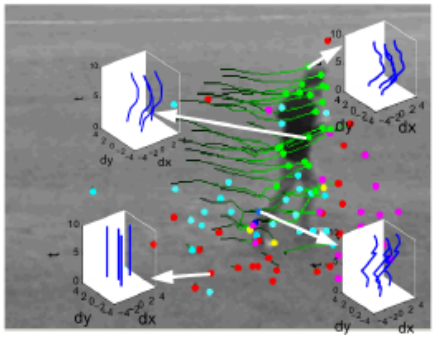


Figure 2.6: Feature trajectories by detecting and tracking spatial interest points. Trajectories are quantized to a library of trajections which are used for action classification [61]

tization, the authors represent gradients in spherical coordinates ϕ, φ; that are divided into an 8×4 histogram. To be rotation-invariant, the axis corresponding to $\phi = \varphi = 0$ is aligned with the dominant orientation of the local neighbourhood.

Willems *et al.* [103] propose the extended SURF (ESURF) descriptor, which extends the image SURF descriptor [7] to videos. Like in previous approaches, the authors divide 3D patches into a grid of local M $\times$ M $\times$ M histograms. Each cell is represented by a vector of weighted sums of uniformly sampled responses of Haar-wavelets along the three axes.

2.4.3 Feature Trajectories

Feature trajectories are based on spatial interest points tracked in time-as opposed to spatio-temporal interest points. Trajectory shapes encode information about local motion patterns and can thus be directly used as a local feature. Messing *et al.* [64] represent feature trajectories of varying length as sequences of log-polar quantized velocities. Activities are modelled using a generative mixture of Markov chain models.

In a different approach, Matikainen *et al.* [61] employ feature trajectories of a fixed length in a bag-of-features framework for action classification, as shown in Figure 2.6. Trajectories of a video are clustered together, and for each cluster centre, an affine transformation matrix is computed. In addition to displacement vectors, the final trajectory descriptor contains elements of the affine transformation matrix for its assigned cluster centre.

2.4.4 Voting based action localization

Combined with a voting scheme, local features can also be employed to spatially and temporally localize actions in videos. For instance, Niebles *et al.* [69] perform a latent topic discovery and model the posterior probability of each quantized feature for a given action class. In order to localize actions, features are spatially clustered in each frame using k-means.

Mikolajczyk and Hirofumi [66] propose a voting approach to localize objects that perform a particular action. The authors use a forest of tree classifiers for fast feature quantization. The GLOH image descriptor [65], together with its dominant motion orientation, is used as local descriptor type. Features in motion cast initial hypotheses for position and scale of objects performing an action. Maxima in the voting space indicate detections, and static features refine their initial localization. For the final pose estimation, the object's global orientation is computed from the orientation of voting features.

In order to localize actions in YouTube video sequences, Liu *et al.* [51] propose an approach based on pruning local features. First, spatio-temporal features are detected and their mean position over a range of neighbouring frames is computed. Features that are too far away from the center position are pruned. Second, static features are computed over all frames. By applying the Page Rank

algorithm over a graph for feature matches in a video sequence, the authors are able to identify discriminative features. For this, similar background features are assumed to be less frequently visible than foreground features. Finally, static and motion features are combined with an AdaBoost classifier. Action localization is carried out with a temporal sliding window over spatio-temporal candidate regions, defined by the centre and the second moments of motion as well as static features.

Willems *et al.* [102] model actions as space-time cubes. They localize drinking actions in movies by casting localization hypotheses for the strongest visual codebook entries of an action. Weak hypotheses are pruned, and a non-linear χ^2 SVM evaluates the BoF representations of remaining ones. Local maxima in the voting space indicated the final action positions.

A related approach by Yuan *et al.* [112] employs the branch-and-bounds algorithm to localize actions in video sequences. Actions are, again, represented as cuboid volumes. The volumes themselves are scored based on mutual information and a Gaussian kernel for density estimation. For a more efficient density estimation, the authors introduce an approximated nearest neighbour search based on local sensitive hashing. Experimental results are shown for the KTH and the CMU actions dataset.

2.4.5 Summary

A key advantage of local features-based approaches is their flexibility with respect to the type of video data. They can be applied to videos for which the localization of humans or their body parts is not feasible. More recent works demonstrate their successful application to real world video data, such as Hollywood movies and YouTube video sequences (Laptev *et al.* [45], Mikolajczyk and Hirofumi [66],

Marszalek *et al.* [60], Liu *et al.* [51], Kovashka *et al.* [41], Le *et al.* [48]).

Even though local feature-based methods are promising, they are still far behind for real world application. In this thesis several features, representations and classification methods are investigated and several techniques to improve the overall classification accuracy are proposed.

2.5 Datasets

This section presents most popular action recognition datasets that are being used to benchmark state-of-the art action recognition algorithms. Subsections 2.5.1 and 2.5.2 describe the KTH and Weizmann actions dataset, respectively. Both datasets have been used extensively in research, however both represent only a set of rather artificial actions with a homogeneous background. Additionally, the Weizmann dataset is about one order of magnitude smaller than KTH. The UCF sports dataset (Subsection 2.5.4) is a collection of TV sport events. It offers a large variety of action classes while being limited in its size. The most challenging and extensive datasets that have been published in the literature are the YouTube and Hollywood2 datasets, which are presented in Subsections 2.5.5 and 2.5.3. They offer an extensive amount of video sequences in realistic setups: YouTube videos and Hollywood movies, respectively.

2.5.1 KTH Actions Dataset

The KTH actions dataset has been introduced by Schuldt *et al.* [86][1].The KTH Human actions dataset contains six action classes: jogging, running, walking,

[1]Available at http://www.nada.kth.se/cvap/actions/

Figure 2.7: Sample Frames from the **KTH** Human actions dataset [86]. Boxing (first column), handclapping (second column), handwaving (third column), jogging (fourth column), running (fifth column), walking (sixth column)

boxing, waving and clapping (see Figure 2.7). These actions are performed by 25 different actors under four different scenarios: outdoors, outdoors with zooming, outdoors with different clothing and indoors. There is considerable variation in the performance, duration and view point. The background is almost static with only slight camera movement. KTH contains 600 action videos and we divide the samples into a test set containing nine subjects (2, 3, 5, 6, 7, 8, 9, 10 and 22), with the remaining 16 subjects assigned for training as proposed in [86]. Evaluation on this dataset is done via multi-class classification. Classification performance is evaluated as average accuracy over all classes.

Figure 2.8: Sample frames from the Weizmann actions dataset [9]

2.5.2 Weizmann Actions Dataset

The Weizmann dataset introduced by Blank *et al.* [9][2] contains 90 videos separated into 10 actions: walk, gallop-sideways, run, jump, bend, one-hand-wave, two-hands-wave, jumping-jack, skip and jump-in-place (see Figure 2.8); each performed by nine different persons. The videos were taken with a static background and fixed viewpoint. This dataset is relatively small compared to KTH and Hollywood2. Blank *et al.* [9] advocate to test using leave-one-out cross-fold validation, i.e., testing is performed for one sequence at a time while training is executed on all remaining sequences. Performance is given in terms of average accuracy (error rate).

2.5.3 Hollywood Actions Dataset

There are two versions of the Hollywood actions dataset: Hollywood1 [45] and Hollywood2 [60]. To avoid exhaustive manual annotation of several hundreds of hours of movie data, the authors use, in both cases, movie scripts that provide textual description of the movie content, such as scenes, characters, transcribed dialogues, and human actions. In a first step, scripts are aligned to movie subtitles, since they usually come without time information. In a second step, classifiers are trained on a bag-of-words representation of the scene description for different action classes. Several features are used: bag-of-words over single words, over adjacent pairs of words, as well as over pairs of words in a small neighbourhood. This allows the trained system to cope with significant variations in the text and to retrieve action samples. The authors manually ensure the visual integrity of annotations in the train and test set and additionally provide a noisy training set.

[2]Available at http://www.wisdom.weizmann.ac.il/ vision/SpaceTimeActions.html

Figure 2.9: Sample frames from the Hollywood2 action dataset [60]

The first version, Hollywood[3] , has been published by Laptev *et al.* [45]. It contains eight different action classes: answering the phone, getting out of the car, hand shaking, hugging, kissing, sitting down, sitting up, and standing up. Action samples have been collected from, in total, 32 different Hollywood movies. The full dataset contains 663 video samples, divided into a clean training set (219 sequences) and a clean test set (211 sequences), where training and test sequences were obtained from different movies. The additional noisy training set consists of 233 sequences.

Hollywood2 is the extended version introduced by Marszalek *et al.* [60][4]. In total, it consists of samples from 69 different Hollywood movies. The initial eight action classes were extended by adding four additional ones: driving car, eating, fighting, and running. Action samples for all classes are illustrated in Figure 2.9. In total, there are 2517 action samples split into a manually cleaned training set (823 sequences) and a test set (884 sequences). The noisy training set contains 810 sequences. Train and test sequences are obtained from different movies.

2.5.4 UCF sports actions Dataset

The UCF sport actions dataset [81][5] contains ten different types of human actions: swinging, diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking (see Figure 2.10). The dataset consists of 150 video samples, which show a large intra-class variability. The performance criterion for the multi-class task is the average accuracy over all classes. The original setup employs leave-one-out for testing.

[3]Available at http://www.irisa.fr/vista/actions/
[4]Available at http://www.irisa.fr/vista/actions/hollywood2
[5]Available at http://www.cs.ucf.edu/vision/public_html/

Figure 2.10: Sample frames from UCF sports action datasets [81]

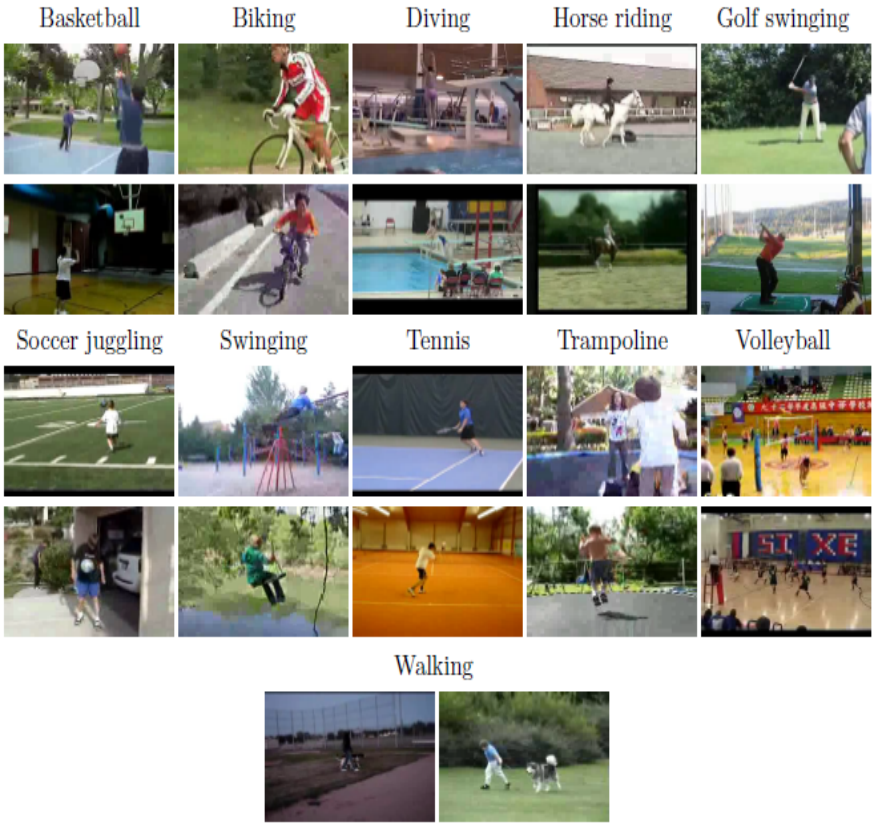


Figure 2.11: Sample frames from the YouTube action dataset [51]

2.5.5 Youtube Actions Dataset

The YouTube dataset has been introduced by Liu *et al.* [51][6] and contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog (see Figure 2.11). This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions etc. The dataset contains a total of 1600 sequences. In the original setting, the evaluation is carried out using cross validation for a set of 25 folds that is defined by the authors. Average accuracy over all classes is used as the performance measure.

[6]Available at http://www.cs.ucf.edu/l̃iujg/YouTube_Action_dataset.html

2.6 Chapter summary

This chapter presented different techniques that have been used by researchers to tackle the action recognition problem over the past three decades. Initially, algorithms were developed with the focus of recognizing single human activities in a clean background. Success in recognizing single human activities led them to explore single human activities with complex and cluttered backgrounds and furthermore, the possibility of recognizing group activities in a cluttered and complex background. To provide a common benchmark to evaluate proposed algorithms, different datasets have been proposed with varying complexity with different evaluation criteria based on the size of the datasets.

Although different techniques have been used by researchers, the local features based action recognition methods are shown to be not only efficient but also provides state-of-the art results compared to other complex approaches, and requires lower computational resources. In this research project local feature based methods have been chosen because of their attractiveness to potential real world applications in resource constrained environments. Even though Bag-of-feature based approach to human action recognition is attractive, there are still several drawbacks such as they failed to capture spatio-temporal relationships which provides major glue about the activities that are closely related spatially and temporally. This problem is tackled in several chapters by incorporating class-specific information into discovered local features. Chapter 5 presents mi-SVM approach, chapter 6 presents css-LDA approach and Chapter 7 presents class-specific sparse codes to capture class-specific information into learned features to boost the performance. Computational complexity of local features is further improved by proposing BRISK+MBH approach in Chapter 4, which significantly reduces computational and storage requirements. A Binary-tree SVM approach has been proposed to scale local feature based approach to hundreds of activities

in Chapter 8. In summary, this thesis addresses three major shortfalls such as scalability, computational efficiency and lack of spatio-temporal feature relationships in the following chapters. Next chapter presents performance evaluation of several state-of-the art descriptors and optimizes the codebook size and kernel matrices for different datasets.

Chapter 3

Comprehensive Evaluation of Local Feature Descriptors

This chapter presents the baseline local spatio-temporal-based action recognition framework followed by comprehensive evaluation of several state-of-the-art local feature descriptors. Bag-of-features based action representation followed by SVM classification is the popular method used in low level action recognition literature to compare and benchmark several feature detectors, descriptors, representation and classification algorithms. In this thesis, this framework has been adapted as a baseline to evaluate proposed feature description and representation algorithms.

The following chapter is organised as follows: The first part of this chapter will focus on the baseline action recognition framework and processing steps involved. Next, the popular feature detectors and descriptors are described and the effect of different pre-processing steps with three different datasets recorded in different environmental settings are evaluated. Finally this chapter presents a differential optical flow descriptor, which improves the performance under moving backgrounds.

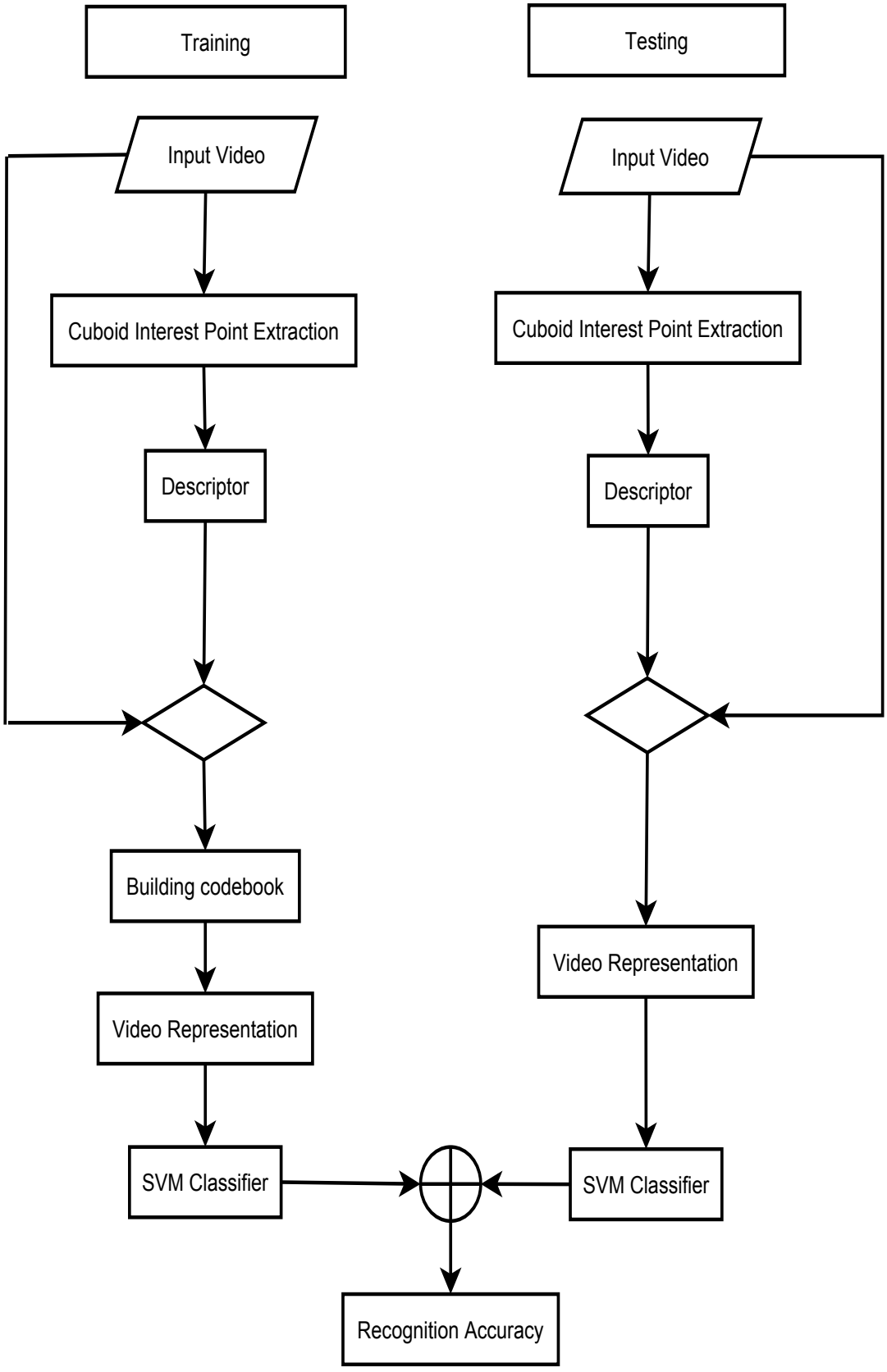


Figure 3.1: Flowchart of the Bag-of-feature based algorithm.

3.1 Human Action Recognition Framework

The whole recognition process can be divided into two phases: the training phase and the testing phase. During the training phase, as per the flow chart shown in Figure 3.1, the interest points as well as the cuboids surrounding them are extracted by some interest point detector from the training sequences, and then the descriptors of each sequence are generated by the structural distribution of interest points or the appearance information embedded in each cuboid. Descriptors from all training sequences are gathered together for further clustering by K-means, which uses Euclidean distance as the clustering metric. The cluster centres are represented as the video words and they constitute the codebook. Each feature descriptor is assigned to a unique video word based on the distance between the descriptor and cluster centres. The codebook membership of each feature descriptor is utilized to create a model representing the characteristics of each class of the training sequences.

During the testing phase, the same steps are followed to extract interest points, build descriptors and assign codebook membership as those done during the training phase. Then, Support Vector Machine (SVM) and K-Nearest Neighbor (KNN) classifiers are adopted to classify each testing sequence to the most probable action type according to the model built in the training phase, and the correctly classified sequences against all sequences give the final recognition accuracy.

3.1.1 Feature detectors

Interest points from a video sequence are localized not only along the spatial dimensions x and y but also the temporal dimension t. Currently there are three

types of detection approaches: static features based on edges and limb shapes, dynamic features based on optical flow measurements and spatio-temporal features obtained from local video patches. In the experiments of this research, the third type interest point detectors were used. Even though several spatio-temporal descriptors have been proposed and used in the literature, Harris3D detector consistently generates robust, view invariant and salient interest points under challenging environmental settings. Hence in this comprehensive study, the Harris3D detector was adopted to study the performance of different local descriptors under different experimental settings.

Harris3D

The Harris3D detector is an extension of the Harris corner detector [31] proposed by Laptev *et al.* [43]. A spatio-temporal second-moment matrix at each video point is computed,

$$\mu(\cdot;\sigma;\tau) = G(\cdot;s\sigma;s\tau) \star \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix}, \tag{3.1}$$

using independent spatial and temporal scale values, σ and τ; a separable Gaussian smoothing function, G; and space-time gradients, ∇L. The Harris corner function for the spatio-temporal domain is defined by combining the determinant and the trace of μ as follows,

$$H = det(\mu) - ktrace^3(\mu),\ H > 0. \tag{3.2}$$

The final locations of interest points are given by the local maxima of Equation (3.2). Following [46], the points are extracted at multiple scales based on a regular sampling of the scale parameters, σ and τ. The original implementation available on-line [1] and standard parameter settings of $k = 0.0005, \sigma^2 = 4, 8, 16, 32, 64, 128, \tau^2 = 24$ with a detection threshold of 10^{-9} has been used to extract spatio-temporal interest points.

3.1.2 Feature Descriptors

A cuboid (spatial temporal video patch) is extracted around each interest point and it contains spatio-temporally windowed pixel values. The size of the cuboid is determined in such a way to provide good performance for a given database. The information contained in each cuboid is utilized to form a representative descriptor and moreover to build the action training model. The locality of cuboids facilitates the feature extraction, which means preprocessing steps are not needed, such as foreground subtraction and figure tracking and alignment etc. Relying on each individual cuboid, we can obtain the appearance information from the cuboid itself as well as structural information from the distribution of all cuboids (interest points).

Feature descriptors are calculated for video patches centred at (x, y, t) for each interest point, (x, y, t, σ, τ). Spatial size, $\Delta_x(\sigma), \Delta_y(\sigma)$, is a function of σ and the temporal length, $\Delta_t(\tau)$, is a function of τ (see Figure 3.2).

[1] http://www.irisa.fr/vista/Equipe/People/Laptev/download.html#stip

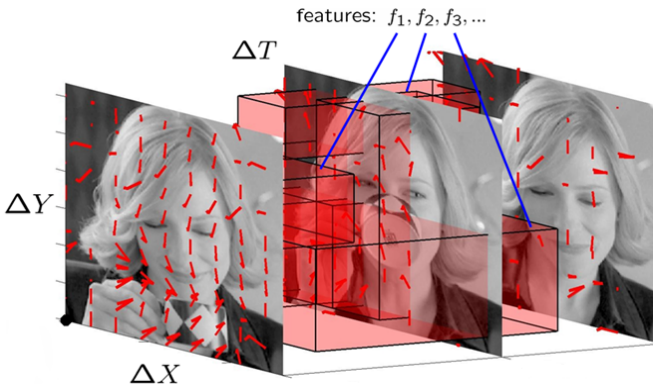


Figure 3.2: Spatio-Temporal Feature Descriptor

HOG/HOF

HOG/HOF descriptors were proposed by Laptev *et al.* [46]. The Histogram of Oriented Gradient (HOG) descriptors are used to describe appearance information and Histogram of Optical Flow (HOF) descriptors are used to describe local motion information present in the detected patches. The histograms are created by accumulating space-time neighborhoods of detected interest points, where the region is given by a cuboid of the size $\Delta_x(\sigma) = \Delta_y(\sigma) = 18\sigma$ and $\Delta_t(\tau) = 8\tau$. Each cuboid region is subdivided into an $n_x \times n_y \times n_t$ grid of cells. For each cell, a 4-bin HOG histogram (4 directions) and a 5-bin HOF histogram (4 directions and an additional bin for no motion) are calculated. Cell histograms are normalised and combined into a HOG/HOF descriptor. This section presents detailed experiments carried out on the HOG, HOF and combined HOG/HOF descriptors with different preprocessing techniques . Experiments use the default grid parameters $n_x = 3, n_y = 3, n_t = 2$ suggested by the authors [45] to provide fair and comprehensive evaluation and the impact of different preprocessing steps involved in the local feature-based action recognition framework. This results in a 72-element HOG descriptor ($3 \times 3 \times 2 \times 4$), a 90-element HOF descriptor ($3 \times 3 \times 2 \times 5$) and a 162-element HOG/HOF descriptor.

HOG3D

The HOG3D descriptor was developed by Klaser *et al.* [39]. It is an extension of the SIFT [53] descriptor to videos. Gradients are computed using an integral video representation. Regular polyhedrons are used to uniformly quantize the orientation of spatio-temporal gradients. Therefore it encapsulates both shape and motion information in a single descriptor. The 3-D patch detected by the Harris3D detector is divided into $n_x \times n_y \times n_t$ cells. Histograms are calculated and normalised for each cell separately and concatenated into a single descriptor. The recommended parameter settings [39] were used to compute the features to provide comparative performance evaluation.

For the KTH and Weizmaan datasets, the optimized parameter settings for controlled datasets with static background were used. The descriptor size is set to $\Delta_x(\sigma) = 16\sigma, \Delta_y(\sigma) = 16\sigma, \Delta_t(\tau) = 4\tau$. Spatial and temporal cells are set to $n_x = 4, n_y = 4$ and $n_t = 4$, and an icosahedron with half orientation is used for quantizing orientations, which results in a dimensionality of 1000.

For the Hollywood2 dataset, the parameters recommended for videos with cluttered backgrounds, camera motion and complex motion patterns have been used to encode the video into a feature vector. The descriptor size is set to $\Delta_x(\sigma) = 14\sigma, \Delta_y(\sigma) = 24\sigma, \Delta_t(\tau) = 12\tau$. Spatial and temporal cells are set to $n_x = 2, n_y = 2$ and $n_t = 5$, and spherical coordinates for half orientation with five spatial and three temporal bins are used for orientation quantization, which results in a descriptor with a dimensionality of 300.

3.1.3 Bag of features representation

A popular representation, based on local features, is the bag-of-features (BoF) model. It originates from document retrieval applications where orderless methods are a popular choice for representing textual data. The bag-of-words model describes text documents as frequency distributions over words and has been applied extensively in this domain.

Schuldt *et al.* [86], Dollar *et al.* [19], Niebles *et al.* [69] proposed the first extensions to action recognition. For the BoF representation in videos, feature detectors determine a set of salient positions present in the video sequences. Feature descriptors compute a vector representation for the local neighbourhood of a given position. The visual vocabulary (or codebook) is then computed by applying a clustering algorithm (e.g., k-means) on feature descriptors obtained from training sequences; each cluster is referred to as a visual word. Descriptors are quantized by assignment to their closest visual word, and video sequences are represented as a histogram of visual word occurrences. Finally a non-linear SVM with χ^2 kernel is a popular classifier that is used throughout different works, e.g., Schuldt *et al.* [86], Dollar *et al.* [19], Laptev *et al.* [45], Willems *et al.* [103], Le *et al.* [48] to benchmark different feature descriptors. Such histogram representations have the ability to capture global statistics about the type of descriptors that are present in the video sequence.

3.1.4 Classification Techniques

Histograms of extracted features are classified for action recognition using the discriminative SVM classifier with different kernel matrices presented in the following section:

Support Vector Machines (SVM) based classification

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. The idea behind SVMs is to make use of a mapping function ϕ that transforms data in input space to data in feature space in such a way as to render a problem linearly separable. The SVM then automatically discovers the optimal separating hyper plane (which, when mapped back into input space via ϕ, can be a complex decision surface) as shown in Figure 3.3.

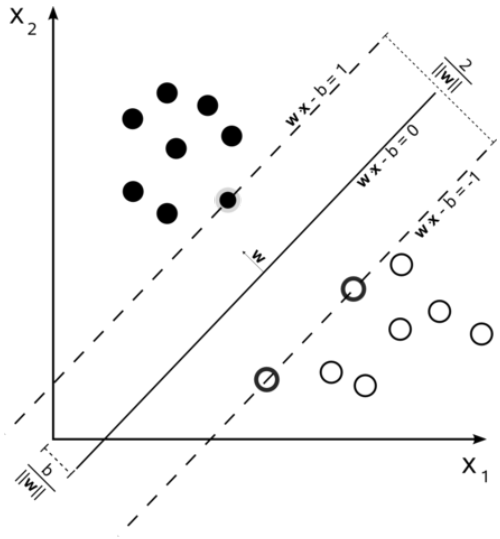


Figure 3.3: Max-margin hyperplane derived from the training of two class SVM [88]

3.2 Experimental results

Related Work

Most of the aforementioned features are evaluated with different experimental settings using a single data set. From the reported results it is difficult to predict which descriptor performs best for a given dataset. However, evaluations such as [88, 97] have sought to overcome this problem and provide a fair evaluation of various detectors and descriptors. Wang *et al.* [97] comprehensively evaluate the performance of different feature detectors and descriptors under a common framework in a wide range of datasets with varying complexity. Experiments were carried out with a bag-of-features representation and an SVM recognition framework. However [97] imposes restrictions on the evaluation, such as using a maximum of 100,000 features to learn cluster centroids, and setting the codebook size to 4,000 for all features and databases. Julian *et al.* [88] proposed a way of evaluating the repeatability of detectors and robustness of descriptors. They evaluated detector performance using repeatability measurements in 3D similar to [103]. For the descriptors they proposed a principled classification pipeline, where every video undergoes eight types of transformations known as challenges, and original video is used as ground truth to observe the extent to which the features change under the challenges. While this evaluation provides valuable insights into the feature detectors and the classification accuracy of individual descriptors under different image alterations, it does not address how well a set of given feature descriptors performs for action recognition under different environmental settings.

In order to address the above mentioned gaps in the action recognition literature, this chapter presents a comprehensive study of popular local feature descriptors under different experimental settings with different datasets recorded under

complex environments. This chapter investigates the effect of the recognition performance with the various pre-processing steps listed below:

- Section 3.2.1 presents an evaluation of a wide range of code book sizes and their influence in the performance of a local feature-based action recognition system in KTH, Hollywood2, and Youtube datasets.
- Section 3.2.2 presents and evaluates different state-of-the-art encoding methods and proposes alternative methods that outperform the popular, baseline k-means clustering methods in KTH, Hollywood2, and Youtube datses.
- Section 3.2.3 presents an evaluation of popular kernel methods in conjunction with the SVM classifier

In the local feature-based action recognition framework, first interest points are located using a popular Haris3D detector. Then local spatio-temporal features are calculated around the neighbourhoods of detected interest points using the feature descriptors followed by the popular bag-of-visual words to represent each video as a histogram of visual words. Finally, a non-linear Support Vector Machine with different kernels is used for classification.

3.2.1 Evaluation of the impact of different code book sizes

Bag-of-video words Representation

In this section, a popular Bag-of-feature representation has been used to evaluate how various codebook sizes influence the action recognition performance. First,

the K-means clustering algorithm is used to generate the vocabulary/ Bag-of-Visual-Words (BOVW). Then, all the descriptors calculated from the training examples are used to generate different sets of vocabularies with different sizes such as 1000, 1500, 2000, 2500, 3000, 3500 and 4000 followed by each video, which is represented by a histogram of visual word occurrences. In these experiments, each video is represented by seven different histograms with different vocabulary sizes. To improve the results further, k-means has been initialized 4-times to obtain the best results.

Evaluation Framework

The baseline non-linear support vector machine (SVM) with a χ^2 kernel [97] has been used as a classification framework to compare the effect of different codebook sizes. First the χ^2 kernel matrix is calculated for each generated histogram of features,

$$K(H_i, H_j) = exp\big(-\frac{1}{A}D(H_i, H_j)\big), \tag{3.3}$$

where H_i and H_j are the histograms of word occurrences and $D(\cdot)$ is the χ^2 distance defined by,

$$D(H_i, H_j) = \frac{1}{2}\sum_k \frac{(H_i(k) - H_j(k))^2}{H_i(k) + H_j(k)}, \tag{3.4}$$

and A is the average distance between all training examples,

$$A = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} D(H_i, H_j). \tag{3.5}$$

A 'one against the rest' approach for multi-class classification is used and the class with highest score is chosen.

The experimental results for various datasets (Section 2.5) with different descriptor/vocabulary combinations are presented in the following subsections. The results reported in various works with different experimental settings can't be directly compared with the results obtained by this framework, however results reported in [97] are comparable with this work.

KTH Dataset

The KTH dataset is one of the most popular benchmark datasets for evaluating action recognition algorithms and is described in Section 2.5.1. The results are presented in Table 7.1. Experimental results demonstrate that the HOF descriptor performed well compared with other descriptors. This can be explained by the fact that as KTH contains actions with a static background, HOF is able to reliably and accurately capture the motion information, while the appearance information captured by HOG is of little use in describing actions. The popular baseline system, where the codebook size is set to 4000, was found to perform well across a wide range of datasets. In contrast to that, in this section the experiments have been carried out with wide range of vocabulary sizes and achieved a 3% improvement for HOF with codebook size of 1500. It is noted that the more compact HOF features achieve optimal performance with a smaller codebook compared to the HOG/HOF and HOG. It is also worth noting that, the HOG/HOF features cannot match HOF alone, suggesting that for situations where the subject is well isolated from the background with sufficient training samples, HOF is a better descriptor than the HOG/HOF combination, which adds more noise to the feature space and reduces the overall performance.

Vocabulary Size	HOG	HOF	HOG/HOF	HOG3D
1000	82.6%	94.6%	92.7%	90.5%
1500	82.6%	**95.0%**	92.8%	90.8%
2000	**82.7%**	94.9%	92.9%	92.3%
2500	82.6%	93.6%	**93.6%**	92.7%
3000	82.6%	93.4%	93.3%	**92.6%**
3500	81.9%	93.4%	93.1%	92.4%
4000	81.9%	93.5%	92.9%	92.1%

Table 3.1: Average Accuracy for different descriptor/codebook combination on **KTH** Dataset

Weizmann Dataset

The results for the Weizmann dataset is presented in Table 3.2. Based on the experimental results, the HOG/HOF descriptor provides highest accuracy of 91.75% with codebook size of 2500, which is nearly 6% higher compared to the baseline system. Next to HOG/HOF, HOF and HOG3D achieve good performance with 90.25% and 90.15% accuracy respectively. Except for HOG3D, all the other descriptors achieve best results with a code book size of 2500. This dataset contains small duration clips with a single action sequence, therefore HOF alone is unable to capture the complete representation of the action, even though the database is recorded with a static background. When the HOF descriptor is augmented with the HOG descriptor, the richer representation is able to improve performance. This suggests that when training data is limited, textural information is of value.

Hollywood2 Dataset

Evaluation results for the Hollywood2 actions dataset is presented in Table 3.3. Features have been extracted from the full spatial videos to maximize the effectiveness of the descriptor. As expected, the combined HOG/HOF descriptor produces the best results. The improvement of 1.2% was obtained with the

Vocabulary Size	HOG	HOF	HOG/HOF	HOG3D
1000	82.6%	87.5%	90.8%	88.2%
1500	83.1%	87.9%	90.4%	88.0%
2000	85.2%	90.1%	91.4%	89.2%
2500	**85.6%**	**90.2%**	**91.7%**	89.3%
3000	85.6%	89.5%	91.2%	89.3%
3500	85.5%	89.2%	91.5%	**90.1%**
4000	85.5%	89.1%	91.6%	89.7%

Table 3.2: Average Accuracy for different descriptor/codebook combination on **Weizmann** Dataset

HOG/HOF descriptor for codebook size of 4000. HOG3D and HOF descriptors have acheived 1.1% and 1.3% improvement respectively over the baseline system. Since Hollywood2 movie clips are rich in context information, the HOG/HOF descriptor performed well as it captures the complete spatio-temporal information present in the videos. In addition to that, Table 3.3 presents the average precision (AP) of every action class in Hollywood2. It is clear from the experimental results that 10 out of 12 action classes show improved performance compared to the baseline [97]. Next to the HOG/HOF descriptor, HOG3D and HOF descriptors perform well in this dataset.

Precision-recall plots for different HOG/HOF codebook sizes for a subset of actions are presented in Figure 3.4. It can be seen that for the four selected actions, consistent and significant performance trends are observed, suggesting that codebook size has a consistent impact across all activities within the database. Further, the extent to which the size of a codebook can be reduced without significant drop in performance was also observed. From the experimental results presented in Table 3.3, it was noted that the codebook size of HOF, HOG/HOF and HOG3D features can be reduced up to 10 fold times with only 6-8% performance degradation, while performance with HOG features degrades by 10-12%. This suggests that both HOF and HOG/HOF can be used in situations where

Action class	Wang *et al.* [97]	Our method
AnswerPhone	20.1%	20.0%
DriveCar	85.4%	**86.9%**
FightPerson	68.9%	**70.7%**
GetOutCar	32.4%	**34.2%**
Kiss	48.6%	**49.9%**
Run	68.6%	**70.2%**
Eat	61.1%	**63.5%**
SitDown	56.3%	**58.1%**
SitUp	19.5%	**22.3%**
StandUp	52.9%	51.1%
HandShake	18.5%	**20.5%**
HugPerson	35.3%	**38.1%**
mAP	47.4%	48.8%

Table 3.3: Average Precision(AP) per action class for the **Hollywood2** dataset compared against the baseline [97]

system speed is crucial, without significantly compromising system performance.

Vocabulary Size	HOG	HOF	HOG/HOF	HOG3D
150	27.1%	35.2%	37.2%	38.1%
250	27.3%	37.6%	40.2%	39.2%
500	31.2%	38.8%	41.4%	40.8%
750	32.5%	40.1%	43.5%	41.5%
1000	34.1%	42.9%	44.7%	42.7%
1500	34.1%	43.4%	45.2%	44.2%
2000	36.2%	43.7%	46.5%	44.9%
2500	38.5%	44.6%	47.2%	45.8%
3000	**40.5%**	**45.2%**	47.8%	46.1%
3500	40.3%	45.2%	48.4%	**46.9%**
4000	40.1%	44.6%	**48.8%**	46.9%

Table 3.4: Mean Average Precision(mAP) for different descriptor/codebook combination on **Hollywood2** Dataset

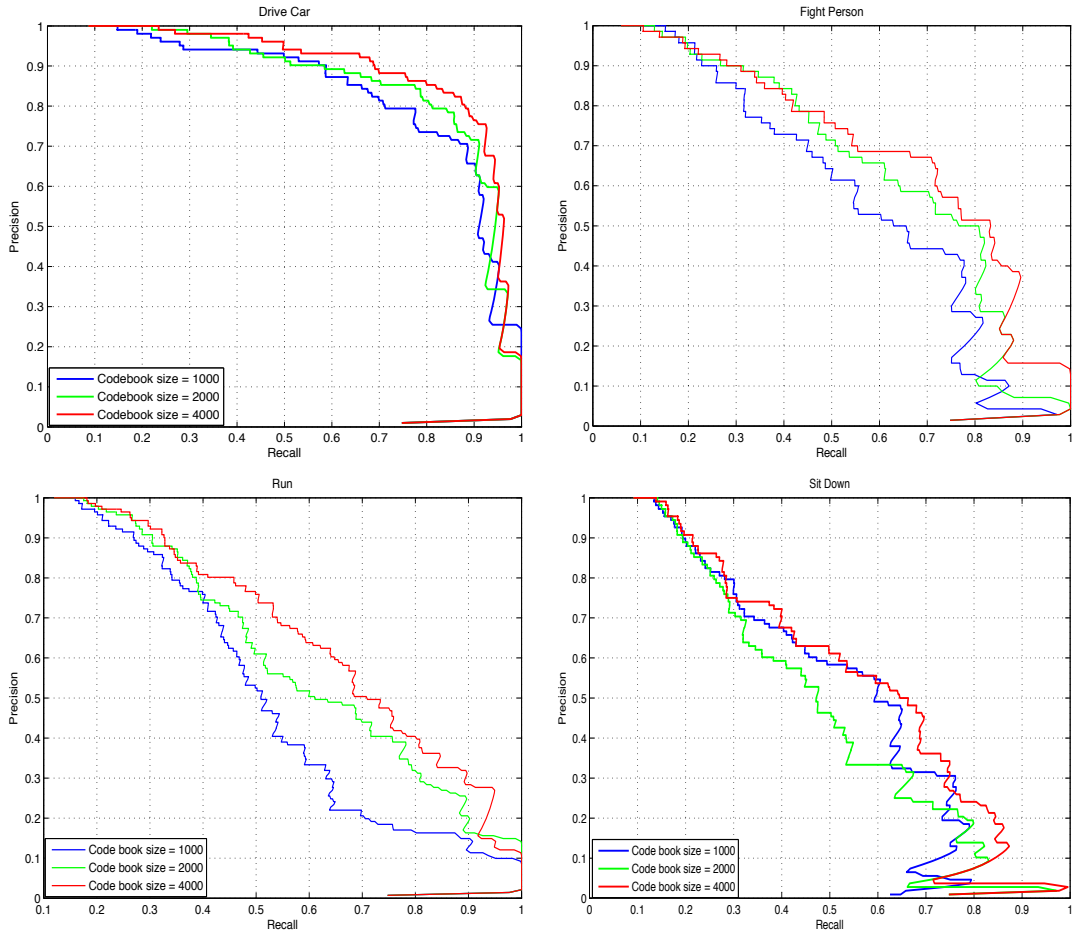


Figure 3.4: Precision-Recall plots for different HOG/HOF codebook sizes on the Hollywood2 actions dataset

Summary

In this section, the effect of codebook sizes has been thoroughly explored with three popular benchmark datasets used for the task of human action recognition. Also, it was found that performance improvement of up to 6% can be achieved by carefully tuning the codebook sizes for different datasets. Also, experimental results suggest that there can be up to $3-4\%$ deviation in the performance depending on the selected codebook size. Therefore, careful consideration of the codebook size is critical in achieving optimal performance. Smaller codebooks can

still achieve good recognition performance with slight performance degradation, which is greatly useful in real time recognition systems where the smaller code book sizes are able to increase the recognition speed without severely compromising the actual performance. It was also found that the HOF descriptor augmented with HOG, performs well in a wide range of environmental settings and provides best results with standard bag of feature representation and SVM classification. HOG features consistently perform poorly, suggesting that motion information is vital and that while appearance can aid classification (i.e. HOG/HOF and HOG3D), appearance alone is insufficient.

3.2.2 Evaluation of feature encoding methods

In this section, three popular encoding schemes such as Vector Quantization (VQ), Sparse Coding (SC) and Locality Constrained Linear Coding (LLC) have been investigated. These encoding schemes have been extensively evaluated over different codebook sizes with different descriptors.

The K-means clustering algorithm is used to generate the vocabulary from a set of local feature descriptors *i.e.* $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{D \times N}$. The K-means algorithm tries to allocate the features to a set of k clusters based on their euclidean distances. The features are partitioned into k clusters with the centres of $B = [b_1, b_2, ..., b_n] \in \mathbb{R}^{D \times K}$. The feature vector x_m is assigned to the cluster k, then $o_{mk} = 1$ and $o_{mj} = 0$ for $j \neq k$. The k-means algorithm uses the following objective function:

$$\arg\min_{o_{mk}, \mu_k} \sum_{m=1}^{N} \sum_{k=1}^{K} o_{mk} \| \, x_m - b_k \, \|^2, \tag{3.6}$$

Vector Quantization (VQ)

In the feature encoding phase, D-dimensional feature descriptors, *i.e.* $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{D \times N}$, extracted from videos are mapped to a codebook $B = [b_1, b_2, ...b_M] \in \mathbb{R}^{D \times M}$, of length M. Though several coding methods exist in literature, vector quantization (VQ) is the most popular method used in action recognition. VQ solves the following least square fitting problem:

$$\begin{aligned} \arg\min_{c} \sum_{i=1}^{N} \parallel x_i - Bc_i \parallel^2, \\ s.t. \parallel c_i \parallel_{l^0} = 1, \parallel c_i \parallel_{l^1} = 1, c_i \succeq 0, \forall i, \end{aligned} \tag{3.7}$$

where $C = [c_1, c_2.., c_N]$ is the set of codes for a video. Since this method only finds a single nearest neighbour, it generates large quantization errors. In addition, VQ ignores the relationship between different bases and needs expensive non-linear kernel projections to improve the recognition accuracy.

Sparse Coding (SC)

To improve the quantization error and obtain a non-linear representation, sparse coding [108] was proposed for object recognition. Unlike vector quantization, the sparse coding represents a feature vector x_n as a sparse linear combination of basis vector. The sparse representation is obtained by solving the following the l_1-norm optimization problem.

$$\arg\min_{c} \sum_{i=1}^{N} \parallel x_i - Bc_i \parallel^2 + \lambda \parallel c_i \parallel_{l^1}. \tag{3.8}$$

In the SC approach, the sparsity regularization term allows the learned representation to capture salient patterns of local descriptors and achieve much lower quantization error compared to VQ.

Locality-constrained Linear Coding (LLC)

LLC [99] was initially introduced for image classification. In the LLC coding, the sparsity term has been replaced with the locality term that captures the proximity of the features with respect to the cluster centres more accurately compared to SC. *i.e.* the cluster centres far away from the local feature x_n is assigned with lower weights, while more weight is given to the closest codebook elements. The coefficients are obtained by solving the following optimization problem,

$$\arg\min_{c} \sum_{i=1}^{N} \| x_i - Bc_i \|^2 + \lambda \| d_i \odot c_i \|^2. \tag{3.9}$$

The $\odot$ represents element wise multiplication, and d_i is the locality adaptor that gives different freedom for each basis vector proportional to its similarity to the input vector, x_i. Compared to VQ, SC and LLC minimize the quantization error by representing an input with multiple elements from the codebook. Furthermore, LLC captures locality information and correlation between similar descriptors.

Experiments & Discussion

The k-means clustering algorithm has been used to generate the codebook, and different encoding methods were used to assign each feature to the codebook elements. In the experiments, HOG/HOF features have been used to investigate

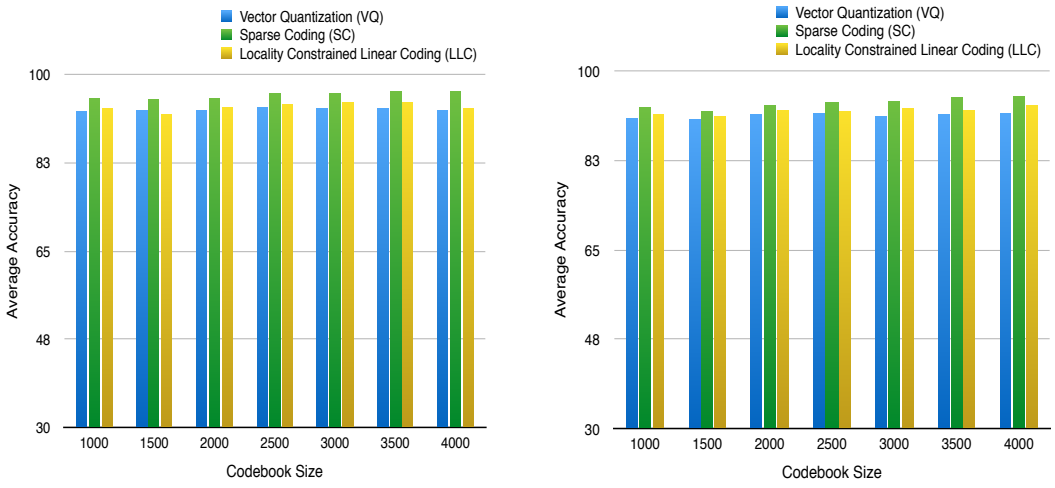
the impact of different encoding methods as they perform well across all datasets with different environmental settings.

- **Vector Quantization (VQ):** This is the baseline method popularly used among vision researchers to tackle several recognition problems such as object recognition, scene recognition, activity recognition *etc*. The final representation has been obtained by sum-pooling followed by $l1$-normalization.
- **Sparse Coding (SC):** In sparse coding the default parameter $\lambda = 0.15$ was chosen to maintain modest sparsity while minimizing the loss. The final representation was obtained using max-pooling followed by $l2$-normalization.
- **Locality constrained Linear Coding (LLC):** In the LLC coding the max-pooling followed by $l2$- normalization was used to create the final representation.

The experimental framework detailed in Section 3.2.1 has been used to study different encoding methods. The experiments have been carried across different codebook sizes such as 1000, 1500, 2000, 2500, 3000, 3500 and 4000.

Figure 3.5 shows how classification accuracy is varying with different encoding schemes in KTH and Weizmann datasets. Experimental results demonstrate that compared to VQ, the LLC and SC performs well in both datasets and the best results were obtained using SC across all the codebook sizes. SC and LLC outperform the VQ by up to 4% with HOG/HOF descriptors.

The experimental results obtained in the Hollywood2 dataset is presented in Figure 3.6. Similar to KTH and Weizmann datasets, the VQ performs poorly compared to SC and LLC encoding methods. The SC and LLC outperform the baseline VQ by up to 6%. Unlike the KTH and Weizmann datasets, which



(a) KTH Dataset (b) Weizmann Dataset

Figure 3.5: *Average classification accuracy of different encoding methods applied on KTH and Weizmann Datasets with HOG/HOF descriptor.* (b) *KTH Dataset,* (b) *Weizmann Dataset.*

were recorded in static environments, the best performance in the Hollywood2 dataset was obtained with LLC encoding. This demonstrates the fact that the locality information helps more compared to sparsity in complex environments to represent videos, and the performance improves with the size of the codebook.

From the experiments carried out on three different datasets it was observed that VQ consistently yields poor performance. This is due to hard vector assignment, where a single feature is assigned to a single codebook element and ignores any relationships between other codebook elements and fails to capture the relationships. On the other hand, in sparse coding the sparsity has produced more discriminative representation, hence the improved performance. Locality constrained linear coding explores the underlying spatio-temporal structure and assigns the features to multiple local codebook elements. It was also noted from the experiments that the sparsity helps to boost the performance in static environments (*i.e.* KTH & Weizmann) while the locality plays an important role in improving the recognition performance of complex datasets such as Hollywood2.

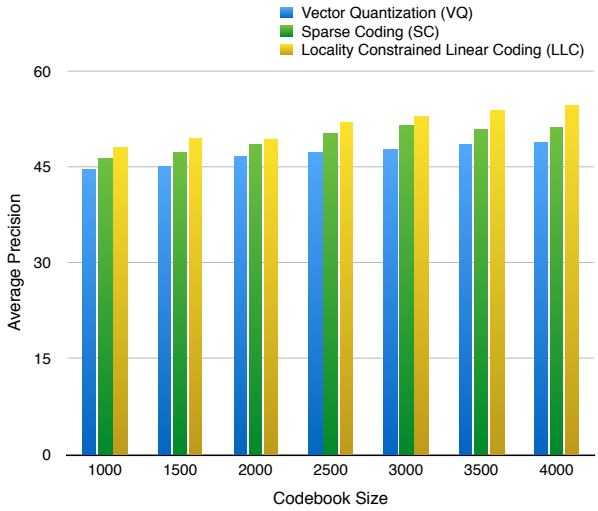


Figure 3.6: Mean Average Precision (mAP) obtained on Hollywood2 dataset with HOG/HOF descriptor and different encoding methods.

3.2.3 Evaluation of different kernel methods

Kernel Methods are a new class of method used in pattern analysis, which can be operated on very general types of features and can detect very general types of relationships. The basic idea behind the kernel method is to transform the low dimensional feature space into higher dimensional space to explore more hidden statistical characteristics. This method also provides a natural way to merge and integrate different types of features.

Kernel methods have a modular framework in which the features are processed into a kernel matrix where the features can be of the same type or various types. In the next step, a variety of kernel algorithms such as Support Vector Machines (SVM), Principal Component Analysis (PCA), Spectral Clustering and Fisher Discriminant Analysis (FDA) can be used to analyse the transformed feature space, using the information contained in the kernel matrix.

For an input feature vector x kernel methods are used to transform the features

into a higher-dimensional vector space in such a way as to find linear relationships in that space, which are not clear in the original low dimensional feature space. Based on the appropriate selection of the feature space, the relationships can be simplified and easily observed, as shown in Figure 3.7.

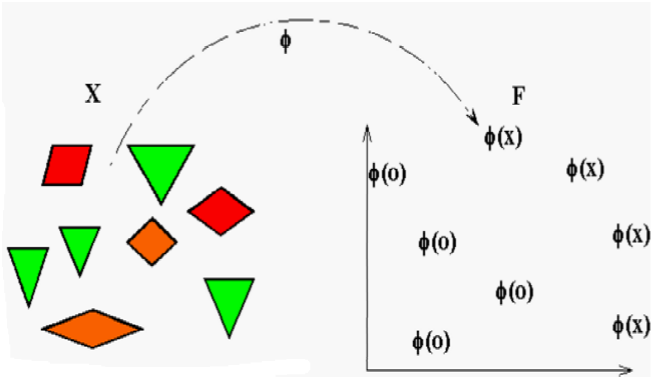


Figure 3.7: Demonstration of a kernel mapping from a input to an non-linear feature space [99].

For models that are based on a fixed non-linear feature space mapping $\phi(x)$, the kernel function is given by the following equation,

$$k(x, x') = \phi(x^T)\phi(x'), \tag{3.10}$$

A kernel is a symmetric function of its arguments and $k(x, x') = k(x', x)$. The simplest kernel function is obtained by the identity mapping of the feature space $\phi(x) = x$ in which case $k(x, x') = x^T x'$, which is a linear kernel. In the SVM classification algorithm, where the input vector enters as a scalar/inner product of a feature space, it allows us to replace the scalar product with the choice of a kernel function. This replacement improves the linear separation compared to the original feature space. This technique is known as 'kernel substitution' or the 'kernel trick'.

Given a training set of instance-label pairs (x_i, y_i), $i = 1, ..., T$ where $x_i \in R^n$ and

$y_i \in \{1, -1\}$, different types of kernels, as listed below, can be used to map the data for SVM classification:

- Linear: $K(x_i, x_j) = x_i^T x_j$.
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T + r)^d, \gamma > 0$.
- Radial Basis Function (RBF) : $K(x_i, x_j) = \exp(-\gamma \left\| x_i - x_j \right\|^2), \gamma > 0$.
- Sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$.
- χ^2: $K(H_i, H_j) = exp\big(-\frac{1}{A} D(H_i, H_j)\big)$.

where γ, r, and d are kernel parameters. The RBF and χ^2 is by far the most popular choice of kernel type used in Support Vector Machines. This is mainly because of their localized and finite responses across the entire range of the real x-axis.

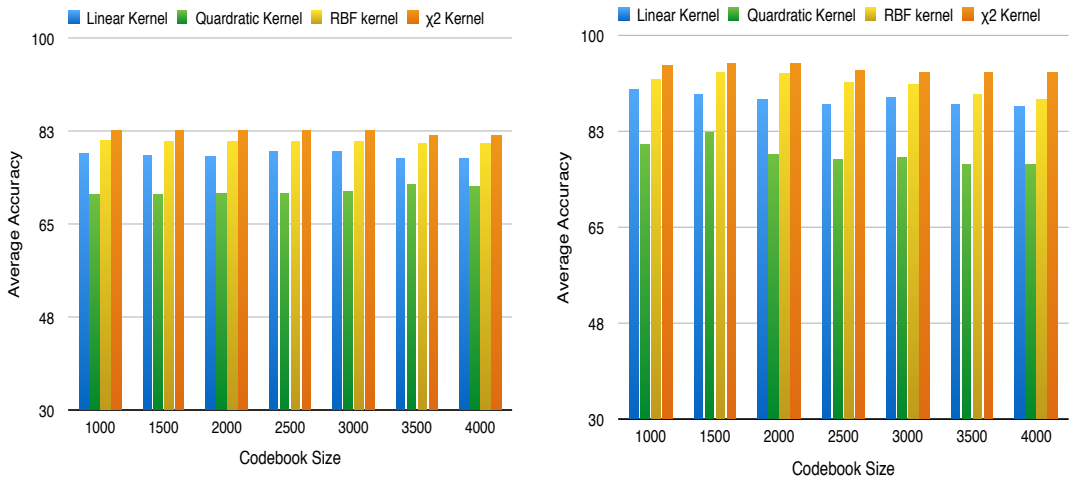
The performance of the SVM [2] classifier heavily depends on the ability of the kernel method used. In this section, four different kernel classifiers, *i.e.*, linear, Polinomial, RBF and χ^2 have been studied with the different codebook sizes over four descriptor combinations.

Experimental Results

The similar framework mentioned in Section 3.2.1 with different kernel matrices has been used to investigate the impact of different kernel matrices. Figure 3.8 shows the experimental results on the KTH dataset. From the results, it can be clearly observed that the χ^2 kernel consistently performs well in all descriptors

[2]Available at https://www.csie.ntu.edu.tw/ cjlin/libsvm/

across all the codebook sizes followed by RBF kernel. At the same time, the quadratic kernel degrades the overall accuracy by 10% to 15%.



(a) HOG Descriptor

(b) HOF Descriptor

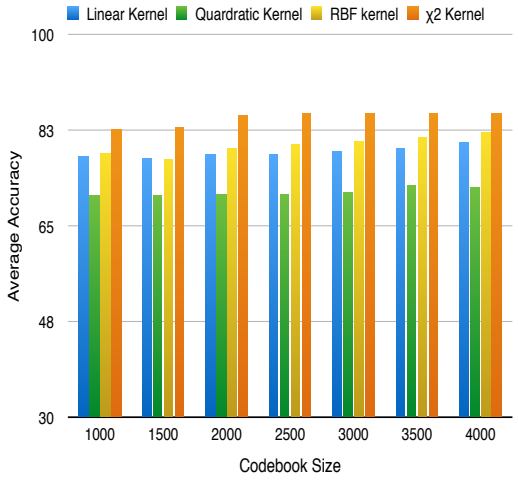
(c) HOG/HOF Descriptor

(d) HOG3D Descriptor

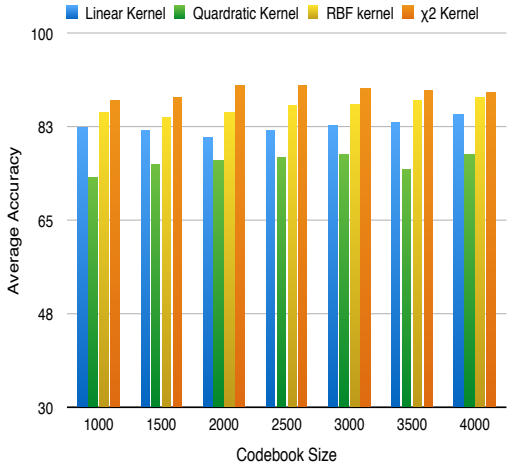
Figure 3.8: *Classification Accuracy of different kernels for different descriptors with KTH Dataset* (a) *HOG Descriptor,* (b) *HOF Descriptor,* (c) *HOG/HOF Descriptor and* (d) *HOG3D Descriptor.*

Figure 3.9 presents average accuracy on Weizmann dataset and Figure 3.10 presents the average precision on Hollywood2 dataset with various kernel matrices. Similar to the KTH dataset χ^2 kernel produces better performance across

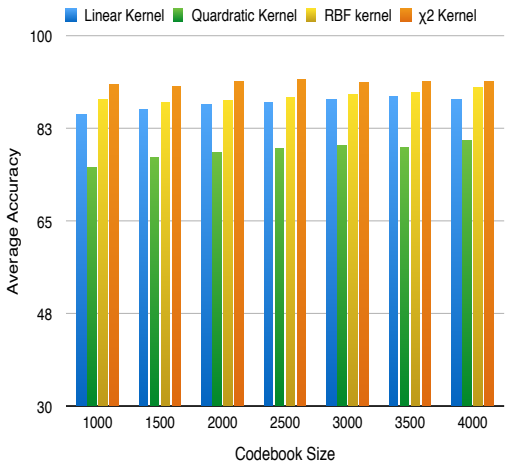
all the codebook sizes followed by RBF kernel. In the mean-time, the choice of the kernel method can vary the overall performance by 2% to 10%. Results for the χ^2 and RBF kernel are comparable and they are good choice for different codebook sizes with different datasets recorded under different environmental conditions while quadratic kernel reduces the overall performance by 10%.



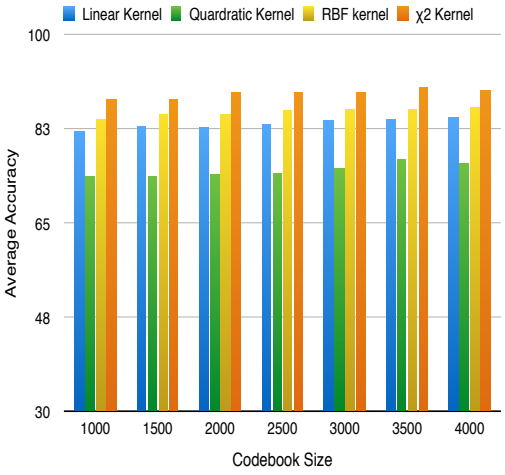
(a) HOG Descriptor



(b) HOF Descriptor

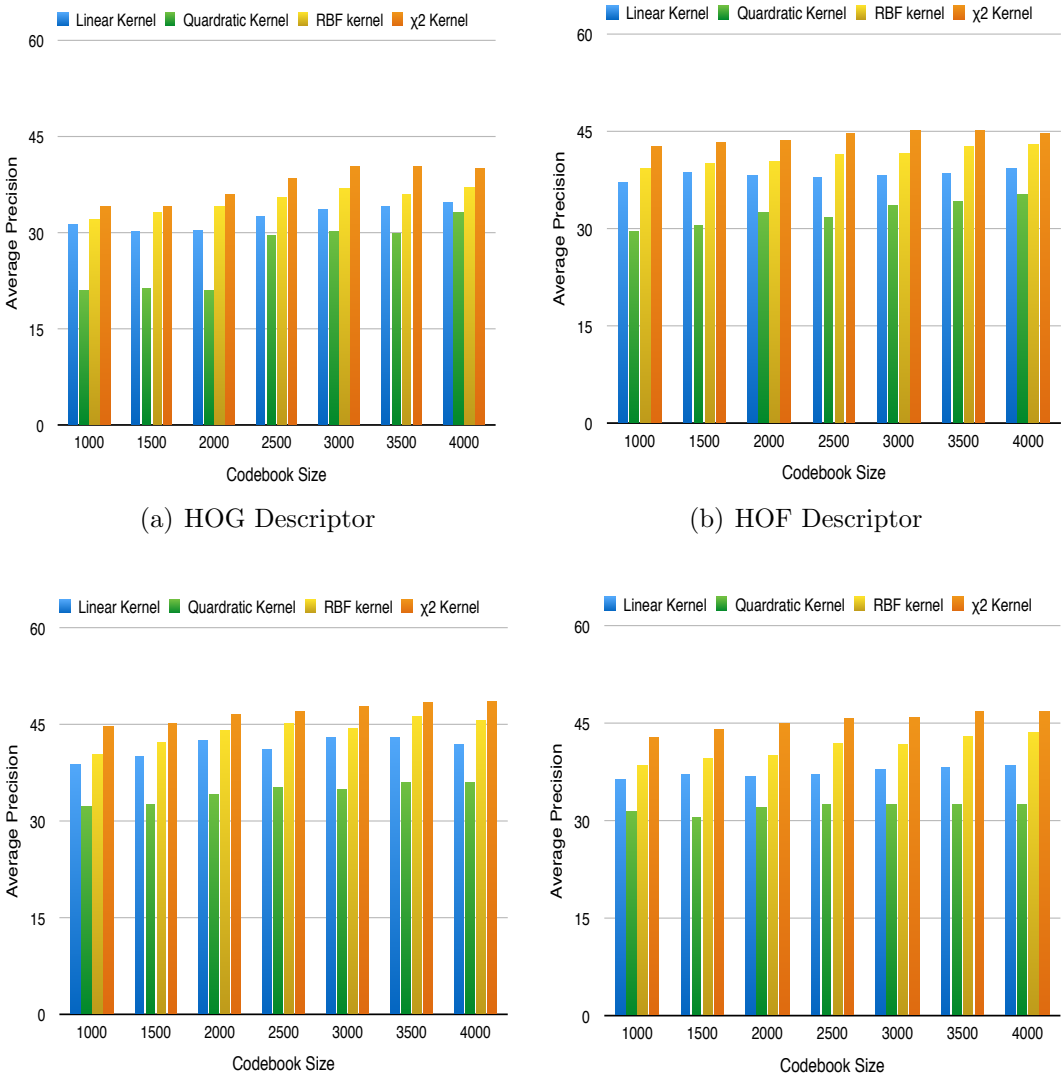


(c) HOG/HOF Descriptor



(d) HOG3D Descriptor

Figure 3.9: *Classification Accuracy of different kernels for different descriptors with Weizmann Dataset* (a) *HOG Descriptor,* (b) *HOF Descriptor,* (c) *HOG/HOF Descriptor and* (d) *HOG3D Descriptor.*



(a) HOG Descriptor

(b) HOF Descriptor

(c) HOG/HOF Descriptor

(d) HOG3D Descriptor

Figure 3.10: *Classification Accuracy of different kernels for different descriptors with Hollywood2 Dataset* (a) *HOG Descriptor,* (b) *HOF Descriptor,* (c) *HOG/HOF Descriptor and* (d) *HOG3D Descriptor.*

3.3 Chapter summary

This chapter presents a comprehensive study of popular, state-of-the art local feature descriptors under different experimental settings with three different datasets. The evaluation presents how different codebook sizes, encoding meth-

ods and various kernel matrices influence the overall classification accuracy, and several conclusions have been drawn based on the results. This study also provides a detailed understanding of how different pre-processing stages can influence the local feature-based action recognition system. This chapter also presents the most effective way to choose different codebooks, encoding and kernel matrices to achieve the best performance for real-world application. KTH, Weizmann and Hollywood2 datasets, the VQ performs poorly compared to SC and LLC encoding methods. The SC and LLC outperform the baseline VQ by up to 6%. In the Kernal evaluation χ^2 kernel consistently performs well in all descriptors across all the codebook sizes followed by RBF kernel. At the same time, the quadratic kernel degrades the overall accuracy by 10% to 15%.

In this study it was also noted that, proper kernel and encoding methods can significantly contribute to the overall performance improvement of 5-10%. The following chapter presents an efficient semi-binary feature descriptor that can be used as an alternative to the local features with significantly reduced computational requirements.

Chapter 4

Semi-Binary Based Video Features for Activity Representation

4.1 Introduction

This chapter addresses the problem of efficient and compact representation of videos by proposing a semi binary-based feature detector-descriptor based on the BRISK detector, which can detect and represent videos with significantly reduced computational requirements, while achieving comparable performance to the state-of-the-art spatio-temporal feature descriptors. This proposed feature detector/descriptor can be used not only in action recognition but also in different video-based applications such as motion analysis, anomalous event analysis, video retrieval *etc.*

4.1.1 The Problem & Motivation

Efficient and effective feature detection and representation plays a crucial role in local feature-based action recognition systems. Although local features have became increasingly popular for representing videos because of their simplicity, efficiency and their state-of-the-art performance with low computational complexity, still they are not applicable for real-time applications due to significant computational requirements. Furthermore, rapid increases in the uptake of mobile devices has increased the demand for algorithms that can run with reduced memory and computational requirements.

Due to the increasing power in consumer electronic devices such as phones and tablets, as well these devices being equipped with cameras, there is a growing interest in being able to process videos on the devices themselves. However, the high computational and memory requirements of such approaches mean they are poorly suited to mobile applications. To address these issues, recently several binary string-based descriptors have been proposed in the context of object recognition [4, 49, 83]. Inspired by their performance with significantly reduced computational requirements, a semi-binary-based feature detector/descriptor for the local feature-based action recognition system is proposed.

4.1.2 Overview of proposed approach

The proposed framework is shown in Figure 4.1. First, the Binary Robust Invariant Scalable Keypoints (BRISK) feature detector is applied on a frame-by-frame basis to detect interest points, then the detected key points are compared against consecutive frames for significant motion. Amongst the detected points only the points with significant motion are retained. Then the retained key

points are encoded with the BRISK descriptor in the spatial domain and Motion Boundary Histogram (MBH) in the temporal domain. This descriptor is not only lightweight but also has lower memory requirements because of the binary nature of the BRISK descriptor, allowing the possibility of applications using hand held devices.

The proposed detector-descriptor performance has been comprehensively evaluated in the context of action classification with a standard, popular bag-of-features with SVM framework. Experiments have been carried out on two popular datasets with varying complexity and yield comparable performance with other descriptors with reduced computational complexity. The proposed descriptor has the potential for real time recognition in resource constrained environments.

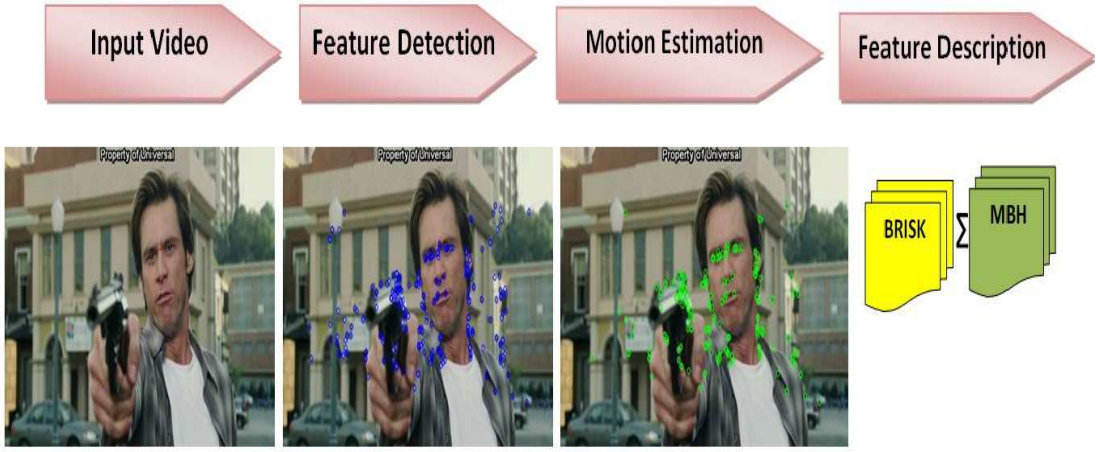


Figure 4.1: Proposed Framework for local feature extraction, which consists of key point detection, motion estimation followed by appearance and motion description.

The reminder of this chapter is organized as follows: Section 4.2 reviews related works that have been performed in the feature detection and descriptor space. Section 4.3 provides details of the proposed descriptor. Experimental results for various datasets are presented in Section 4.4. Finally, Section 4.5 concludes the chapter.

4.2 Related work

There are numerous feature detectors that have been proposed in the literature [20, 43, 70, 103, 104] to extract regions of interest, and the detail description is presented in Section 2.4.1.

Descriptors are used to code appearance and motion information from the region of selected interest points using image gradients and optical flow. Several descriptors have been proposed in the past [40, 43, 87, 103]. Detail description of these approaches is presented in Section 2.4.2. Williams *et al.* [103] extends the SURF descriptor to video, by representing each cell as a vector of weighted sums of uniformly sampled responses to Haar-wavelets along the three axes. However the descriptors proposed in [87, 103] have been directly extended from 2D to the temporal domain (*i.e.* 3D), and they treat both spatial and temporal domains in a similar manner. Therefore, these representations fail to accurately capture temporal information, which has dissimilar characteristics to 2D spatial information.

Recently in object recognition the focus has been given to detecting and representing key points quickly with low computational and memory requirements, more suitable for real-time applications. Several feature detectors [49, 56, 82] have been proposed to process the images almost in real time. FAST [82] imposes hard real-time constraints to achieve state-of-the art results while AGAST [56] improves the performance by extending the FAST detector. The recently proposed BRISK [49] is a multi-scale AGAST, where the Features from Accelerated Segment Test (FAST) score is used as a saliency measure to search maxima in scale space. The increasing focus on high quality, computationally efficient performance has also yielded several binary string features for image encoding [4, 14, 49, 83]. BRIEF [14], created using simple image intensity comparisons

at random pixel locations, that yields a description consisting of binary strings. Rublee *et al.* proposed Oriented fast and Rotated BRIEF (ORB) [83], by making BRIEF more invariant to scale and rotation changes as well as robust to noise. BRISK [49] uses a specific sampling pattern to build a descriptor invariant to rotation and scale. Alexandre *et al.* proposed Fast Retina Keypoint (FREAK) [4], where a binary string is computed efficiently by comparing image intensities over a retinal sampling pattern.

In this chapter, inspired by the above fast and efficient detectors and descriptors, an extension of the BRISK descriptor to videos has been proposed in the context of activity recognition.

The BRISK detector has been chosen as it is a high-quality, fast-key point detector. Similar to the above mentioned descriptors for videos, which detect key points at multi-scale and with invariance to transformation, BRISK achieves these with dramatically reduced computational cost. In this proposed method, more emphasis has been given to make the algorithm as simple as possible to minimize computational complexity while retaining the classification performance. In order to handle the spatial and temporal domain separately, the BRISK descriptor is used to encode the appearance information while the Motion Boundary Histogram (MBH) [17] is used to encode the motion information. Unlike other optical flow based methods, MBH features remove the camera motion and represent only the actual motion present. Experimental results are presented using the standardized evaluation framework (bag-of-words with SVM), and performance on benchmark datasets KTH [86] and Hollywood2 [60] demonstrate comparable performance with other state-of-the-art approaches with much greater efficiency. The details of the proposed method are presented in the following section.

4.3 Proposed method

As shown in Figure 4.1, the proposed method consists of four steps. In the first step, BRISK is employed on a frame-by-frame basis to detect interest points. Figure 4.1 shows the extracted interest points in a frame. Then a sparse optical flow algorithm [55] is applied to detect the motion of all detected key points at the current frame, t, w.r.t. frame $t + W$. Points which exhibit motion are considered as candidate spatio-temporal points for video description. In the third step, appearance information of the points is extracted using the BRISK descriptor and the motion component is extracted using the MBH descriptor. Lastly, the final spatio-temporal feature is created by combining the appearance and motion features.

4.3.1 Interest Point Detection

In the proposed framework, interest points are detected based only on the appearance information in each frame of a video sequence. Interest points are extracted from each frame by applying the BRISK detector, which is an order of magnitude faster than other algorithms. The BRISK detector detects the location and scale of each key point in the continuous domain via quadratic function fitting. Furthermore, it detects the actual scale of a key point in a continuous scale-space. The scale-space pyramid consists of n octaves c_i; and n intra-octaves d_i; these are formed by down sampling the original frame (*i.e.* c_0). Intra octaves are positioned between two adjacent octaves (c_i and c_{i+1}) as shown in the Figure 4.2. Potential interest points are detected by applying a FAST 9-16 detector in each octave and intra-octave separately. Points having the highest score amongst eight neighbouring FAST scores in the same layer and a lower score in the layer above and below are considered as potential interest points. Each interest point

$P_t = (x_t, y_t)$ detected in the t^{th} frame has its own spatial size, Δ_x and Δ_y. The detected keypoints in KTH and Hollywood2 datasets are shown in the Figures 4.3 and 4.4 respectively. These datasets have been chosen to demonstrate the effectiveness of the descriptor in both static and complex environments.

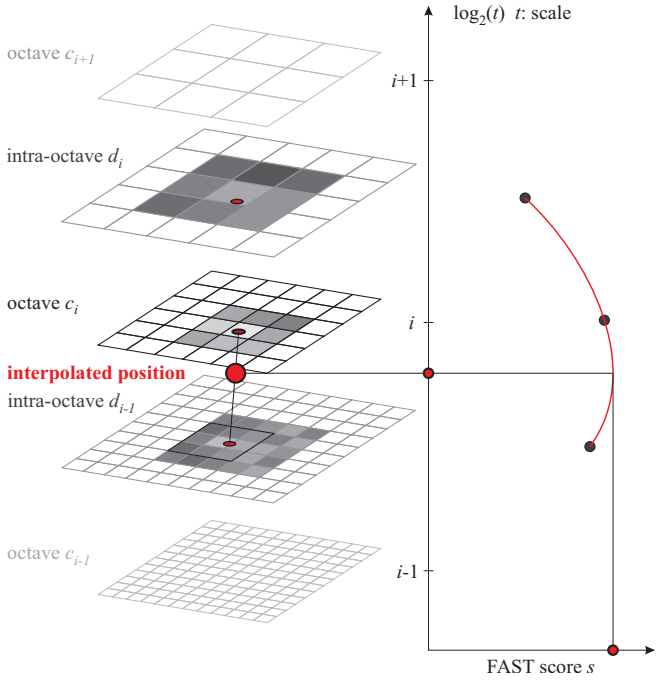


Figure 4.2: Brisk Interest point detector [49]; a keypoint is detected by analyzing the saliency scores in c_i and the layers above and below.

4.3.2 Motion Estimation

The points that were detected by considering spatial domain characteristics may contain points that do not possess significant motion. These points are not required to represent videos effectively and hence reduce the discriminatory power of the descriptor. To choose the best candidate points for describing video from amongst those detected by BRISK, and to improve efficiency, a sparse optical flow algorithm [55] has been applied.

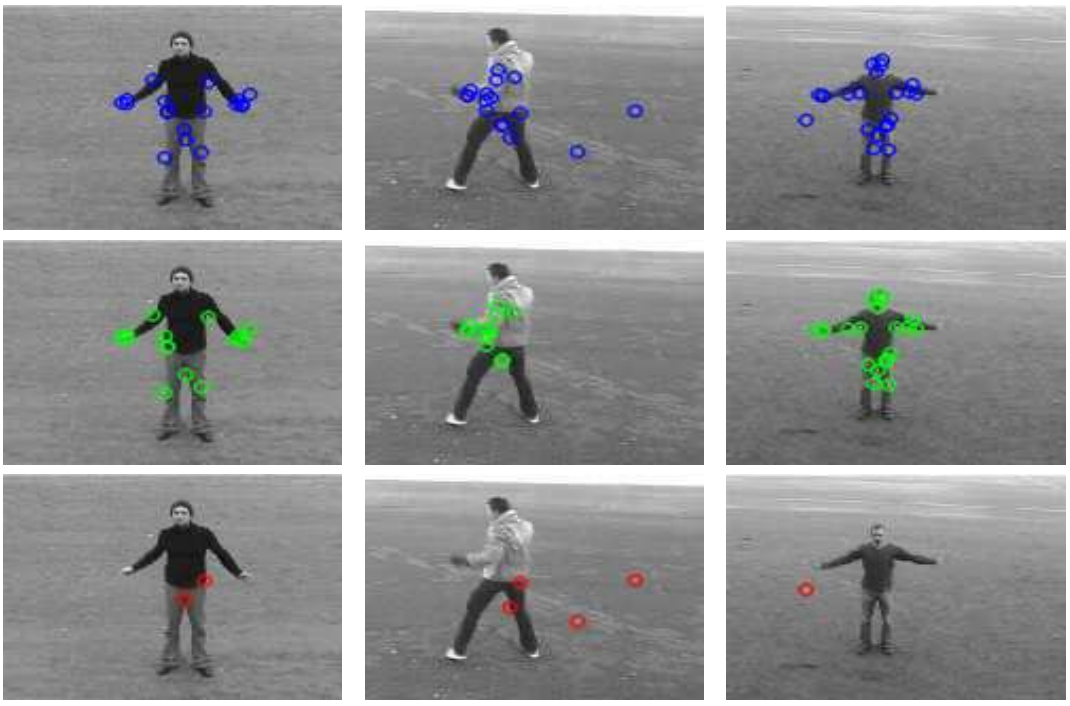


Figure 4.3: Key points detected by **BRISK** detector on sample frames from **KTH** dataset are shown in the first row. The second row shows the candidate key points for description and the last row shows the eliminated points due to insignificant motion. Sample actions are Hand clapping (first column), Boxing (second column), Waiving (third column)

.

Optical flow has been calculated between the points detected in the current frame F_t and the next frame F_{t+W}, where W is the temporal window size between the current frame and next frame against which optical flow is compared. *i.e.* the next frame is W frames away from the current frame. The selection of W is paramount to detect and describe the cuboid. Values of W that are too small will result in very little motion being detected and most key points being removed. On the other hand, too large a W will cause the frames to be too far apart and will fail to capture actions with small temporal duration when the frame rate is too low. Though this parameter has to be tuned for various databases based on the type of actions present, in Section 4.4, a detailed analysis on how the window size is affecting the overall performance is presented with KTH and Hollywood2

datasets.

Local coherent motion around the key point (described by the spatial size, $\Delta_x \times \Delta_y$) detected by BRISK is analysed to determine if it is in motion. Key points which are in motion (as determined by the optical flow) are used as feature points about which to extract appearance and motion information. In order to account for temporal information in the key points and to extract motion information, a cuboid is formed by setting temporal size, $\Delta_t = W$.

Figure 4.4: Sample Frames from **Hollywood2** human actions dataset are shown in the first row, key points detected by **BRISK** are shown in the second row. The third row shows the candidate key points for description and the final row shows the eliminated points due to insignificant motion. Sample actions are Eat (first column), Run (second column), Kiss (third column), Getoutcar(fourth column) and Answerphone (fifth column)

.

4.3.3 Appearance Modelling

For each interest point $P_t = (x_t, y_t, \Delta_x, \Delta_y, \Delta_t)$, the BRISK descriptor is applied to efficiently capture the appearance information. BRISK is calculated as a binary string and consists of the results of a binary comparison. When computing the descriptor, the neighbourhood of the key point is sampled in a pattern similar to the DAISY descriptor to achieve restricted memory and processing requirements while focusing on maximizing descriptiveness. Then the sampled pattern is rotated around the key point and an intensity comparison is done between point pairs to form a bit string descriptor. The BRISK descriptor yields a 512 length bit vector around each detected key point to represent the video efficiently.

4.3.4 Motion Modelling

To capture motion information surrounding the interest point, the current frame and the subsequent W frames are considered. Optical flow-based methods are widely used to encode the motion information in a spatio-temporal video feature representation. While optical flow is a popular method used to represent motion, the calculated motion between two adjacent frames includes constant camera and background motion in addition to the actual motion relating to the action being performed. To alleviate this problem, the Motion Boundary Histogram (MBH) has been used, where the optical flow is resolved into horizontal and vertical components, *i.e.* $I^w = (I^x, I^y)$; then gradient magnitude and direction is calculated on the two flow components separately. In this way, constant camera and background motion is removed (the gradient of a constant is 0) and only the foreground motion is retained.

Each cuboid is subdivided into a $(n_x = 3, n_y = 3, n_t = 2)$ grid of cuboids. A

normalized 4 bin histogram is calculated for each cell resulting in a 72 dimension ($3 \times 3 \times 2 \times 4$) feature for each component (*i.e.* x and y).

The final spatio-temporal descriptor is formed by concatenating[1] the above appearance and motion descriptors. The appearance representation based on BRISK consists of a 512 dimensional binary string vector and the motion representation consists of a 144 dimensional fixed point vector; resulting in a 656 dimensional semi-binary feature vector for each key point.

4.4 Experimental Results and Discussion

Our proposed method has been extensively tested with a popular, widely used local feature-based action recognition system detailed in Section 3.2.1. This pipeline consists of feature detection and extraction, then vector quantization with K-means followed by classification with an SVM using a χ^2 kernel as shown in Figure 3.1. In this framework the first step is replaced with the proposed key point detection and description, keeping the remaining parts the same. The experiments have been carried out with two popular benchmark datasets with varying complexity.

4.4.1 KTH Dataset

The Figure 4.6 shows how the classification performance of the KTH dataset varies with different temporal window sizes (W) used to detect the relative motion between two frame sequences. From the Figure 4.6 it can be observed that the

[1]First the interest points were detected followed by spatial encoding and motion encoding. These encoded features and joined together (simple matrix concatenation) to provide complete representation.

Approach	Average Accuracy
Schüldt *et al.* [86]	71.72%
Dollar *et al.* [20]	81.17%
Niebles *et al.* [69]	81.50%
Our Method	**91.2%**
Laptev *et al.* [45]	91.8%
Liu *et al.* [51]	93.8%
Wang *et al.* [94]	94.2%
Le *et al.* [48]	93.9%

Table 4.1: Comparison of recognition accuracy on the **KTH** Dataset using different approaches. Approaches used in [69], [94], [48] are not fallen into spatio-temporal descriptors.

maximum classification performance in KTH is obtained when the window size $W = 5$. Therefore, for the remaining experiments in KTH datasets, the window size is set to 5. The lower temporal window size eliminates most of the interest points due to lack of relative motion and reduces the overall performance. In the meantime the higher temporal window size ignores significant motion present between two consecutive frames and fails to capture fine temporal movements.

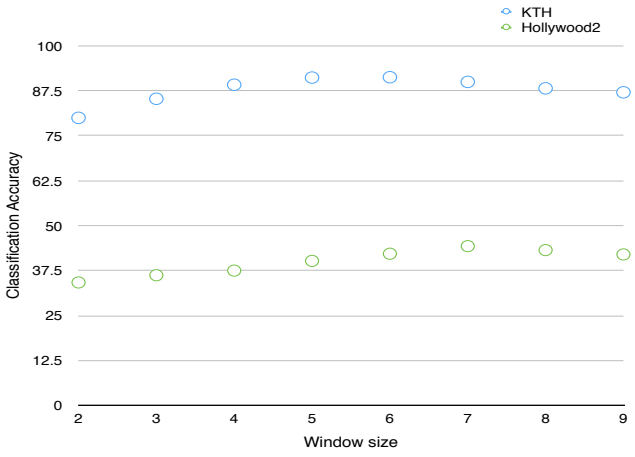


Figure 4.5: The comparison between the classification performance and different temporal window sizes (W) in KTH and Hollywood2 Datasets.

Comparisons with other state-of-the-art methods are presented in Table 4.1. The proposed method achieves almost comparable performance with other local

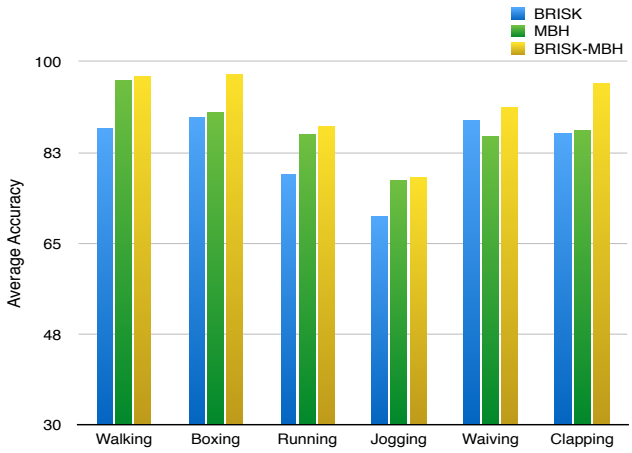


Figure 4.6: The recognition accuracy of three descriptors BRISK, MBH, BRISK+MBH with BRISK key point detector in KTH dataset.

feature-based methods with significantly reduced computational requirements. Figure 4.6 shows the classification accuracy of the six different action classes within KTH with different combinations of descriptors. The BRISK+MBH combination performs well across all classes compared to BRISK or MBH only. This demonstrates that both motion and appearance are important to distinguish between actions.

The motion descriptor (MBH) performs well in actions such as running, jogging and walking compared to BRISK. This can be explained by the fact that these three actions are almost same in the spatial domain, but occur at different speeds in the temporal domain, hence MBH captures the variation well while BRISK alone confuses these action categories. On the other hand, the appearance feature performs well in hand clapping, hand waving and boxing where a significant amount of contextual information is present and is well captured by the BRISK descriptor (see Figure 4.6).

Interestingly, it was noted that the BRISK only descriptor performs well in almost all action classes, which allows for a further reduction in computational and memory requirements. As mentioned earlier, BRISK is a binary string descriptor

with the dimension of 512, and only requires 64 Bytes of memory for each key point. Also it was noted that, because of the static background in KTH, almost all BRISK detected key points are placed around the person (see Figure 4.3) where significant motion is present, which eliminates the need for motion estimation. *i.e.* few points are eliminated due to the lack of motion. In addition, when building the codebook using k-means, the hamming distance can be used to obtain the histogram of visual words rather than costly Euclidean distance. This further reduces the computational complexity. The confusion matrix of the KTH dataset is presented in the Table 4.7.

	Walking	Boxing	Running	Jogging	Waiving	Clapping
Walking	0.96	0.00	0.01	0.03	0.00	0.00
Boxing	0.00	0.94	0.00	0.00	0.04	0.02
Running	0.00	0.00	0.85	0.08	0.07	0.00
Jogging	0.09	0.00	0.05	0.86	0.00	0.00
Waiving	0.00	0.04	0.00	0.00	0.92	0.04
Clapping	0.00	0.05	0.00	0.00	0.02	0.93

Figure 4.7: The confusion matrix of the KTH dataset with BRISK detector and BRISK+MBH descriptor, the temporal window size is set to $W = 5$.

4.4.2 Hollywood2

The effect of different temporal window sizes has been investigated with Hollywood2 dataset (see Figure 4.5) to optimize the window size for best classification performance. The experimental results show that the optimum results are obtained when the temporal window size is set to 7. This value is higher compared to the KTH dataset because of higher frame rate and lower relative motion between the adjacent frames.

As shown in Table 4.3, BRISK+MBH detector-descriptor combination performs well compared with the other local statio-temporal feature based approaches.

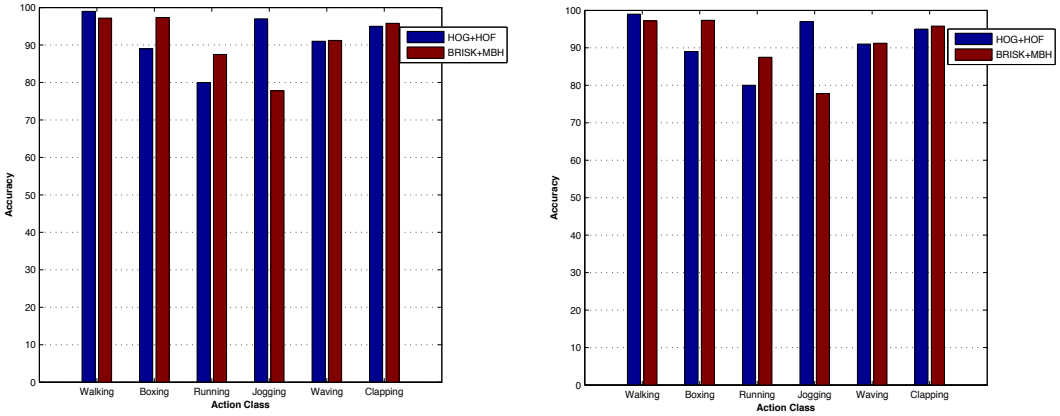


Figure 4.8: Recognition accuracy for different classes on the KTH dataset: Figure (**left**) shows the performance with three different descriptors BRISK, MBH and BRISK+MBH, Figure (**right**) shows the performance of the BRISK detector with BRISK+MBH descriptor against Harris3D detector with HOG+HOF descriptor

Approach	mean AP
Our Method	**44.3%**
Laptev *et al.* [45]	45.2%
Le *et al.* [48]	53.3%
Wang *et al.* [94]	58.2%

Table 4.2: Comparison of recognition accuracy on the **Hollywood2** Dataset using different approaches. Approaches used in [94], [48] are not fallen into spatio-temporal descriptors.

Still, the Harris3D with HOG-HOF performs slightly better compared to BRISK detector with the BRISK-MBH descriptor. This is due to the nature of Harris3D, which explores both spatial and temporal content when detecting a point, while the BRISK detector detects the points based only on the spatial context. Though there is a slight performance compromise against Harris3D, the BRISK detector detects points far more efficiently than Harris3D.

Table 4.3 compares the performance of the proposed BRISK+MBH features to the popular HOG+HOF features. The BRISK+MBH descriptor outperforms the HOG-HOF combination on both Harris3D and BRISK keypoints. On the other

Dataset	Harris3D		BRISK	
	HOG+HOF	BRISK+MBH	HOG+HOF	BRISK+MBH
KTH	91.8	92.3	88.3	91.15
Hollywood2	45.2	47.2	42.1	44.3

Table 4.3: Performance comparison between the popular HOG+HOF and BRISK+MBH descriptor with the Harris3D and the BRISK keypoints. Average accuracy is reported on the **KTH** dataset and mean average precision is reported on the **Hollywood2** Dataset.

hand, Harris3D outperforms the BRISK detector at the expense of computational complexity.

The confusion matrix of the Hollywood2 dataset is presented in the Table 4.9. Significant confusion has been observed among two class subsets such as {Hand shake, Hug person, Kiss} and { Sit down, Sit up, Stand up}. In the first set of actions involved with two people, only a small spatial scale differentiates the activities and the descriptor fails to capture more discriminative information to improve the performance. The second set of actions can only be differentiated by the temporal order in which the action takes place. Most of the confusion occurs due to the Bag-of-feature representation where the temporal order in which the action takes place is usually ignored.

	Answer phone	Drive car	Eat	Fight person	Get out car	Hand shake	Hug person	Kiss	Run	Sit down	Sit up	Stand up
Answer phone	0.19	0.08	0.14	0.02	0.01	0.05	0.07	0.09	0.07	0.10	0.09	0.09
Drive car	0.00	0.82	0.00	0.03	0.10	0.00	0.05	0.00	0.00	0.00	0.00	0.00
Eat	0.10	0.00	0.64	0.07		0.00	0.02	0.06	0.02	0.03	0.04	0.02
Fight person	0.00	0.00	0.01	0.57	0.00	0.04	0.22	0.12	0.02	0.00	0.02	0.00
Get out car	0.07	0.31	0.01	0.05	0.42	0.03	0.01	0.02	0.04	0.00	0.02	0.02
Hand shake	0.00	0.03	0.02	0.13	0.03	0.12	0.14	0.21	0.15	0.07	0.08	0.02
Hug person	0.03	0.00	0.00	0.13	0.00	0.15	0.32	0.32	0.02	0.01	0.00	0.02
Kiss	0.00	0.00	0.00	0.12	0.00	0.14	0.23	0.48	0.02	0.00	0.01	0.00
Run	0.02	0.03	0.00	0.02	0.08	0.00	0.00	0.04	0.62	0.06	0.07	0.06
Sit down	0.00	0.00	0.02	0.00	0.00	0.04	0.01	0.02	0.01	0.48	0.23	0.19
Sit up	0.00	0.00	0.03	0.00	0.00	0.00	0.03	0.03	0.01	0.39	0.19	0.32
Stand up	0.00	0.00	0.00	0.02	0.00	0.03	0.01	0.02	0.00	0.23	0.21	0.48

Figure 4.9: The confusion matrix of the Hollywood2 dataset with BRISK detector and BRISK+MBH descriptor, the temporal window size is set to $W = 7$.

4.4.3 Computational complexity

The main aim of this descriptor is to achieve computational efficiency while retaining a reasonable level of performance. To experimentally evaluate the computational complexity of this feature detector and descriptor, 100 randomly selected video clips from the KTH and Hollywood2 dataset have been used. Each video in KTH has 100 frames on average with the frame size of 160×120, while Hollywood2 has a frame size of 528×224 pixels and consists of 350 frames on average. The average time to detect features within the frame, perform motion estimation and extract appearance and motion descriptions are presented in Table 4.4. Timings are reported on a PC with a core i7, 3.40 GHz processor running the Windows 7 operating system (32bit) with a single core for processing. Our proposed implementation uses an unoptimized C++ code, and authors' original STIP implementation [2] was used to calculate computational complexity for STIP method. One of the most compelling motivations for the use of binary descriptors is their efficiency and compactness. If they are stored as floating-point values, the storage savings of binary features are even more significant. Even if the real value parameterization descriptors are stored in a quantised form they still requires at least a byte per dimension to store without losing much precision. Overall, binary descriptors reduce storage requirements significantly. Appearance and motion features can be calculated simultaneously, once the motion estimation phase is done.

The computational complexity comparison between the proposed approach (BRISK+ MBH) with the popular spatio-temporal interest points (STIP) in Hollywood2 dataset is presented in Table 4.6.

The features are extracted at the speed of 6.96 frames/second using BRISK-MBH,

[2] http://www.irisa.fr/vista/Equipe/People/Laptev/download.html#stip

	Proposed Method		STIP Method	
	KTH	Hollywood2	KTH	Hollywood2
Feature detection (ms)	4.57	18.43	14.28	41.05
Motion estimation (ms)	2.31	25.60	-	-
Apperance description (ms)	62.30	74.60	140.32	220.40
Motion description (ms)	72.30	103.00	160.30	270.50

Table 4.4: Time spent on different stages of our proposed feature detection and description method against the STIP method. Processing time is calculated on randomly selected 100 samples from each datasets without parallel processing.

	Proposed Method	STIP
Feature detection	$\mathcal{O}(kn^2)$	$\mathcal{O}(kn^3)$
Motion estimation	$\mathcal{O}(kn)$	-
Apperance description	$\mathcal{O}(kn)$	$\mathcal{O}(kn^2)$
Motion description	$\mathcal{O}(kn^2)$	$\mathcal{O}(k^2n^2)$

Table 4.5: Algorithmic complexity of our proposed method against the STIP method during feature detection and description. Spatial size of the cuboid is assumed to be $n \times n$ and temporal size is k.

which is nearly four times higher compared to STIP with nearly seven times more features per frame compared to STIP.

Algorithmic complexity

Table 4.5 compares algorithmic complexity in different stages of feature detection and description for cuboids with spatial size of $n \times n$ and temporal size of k. Our proposed method yields overall computational complexity of $\mathcal{O}(kn^2) < \mathcal{O}(n^3)$, where $k < n \quad \text{and} \quad k, n > 0$. (i.e. temporal size of the cuboid is always smaller than the spatial size). In the mean time STIP method yields the overall complexity of $\mathcal{O}(kn^3) < \mathcal{O}(n^4)$, where $k < n \quad \text{and} \quad k, n > 0$. It is clear that the STIP is more computationally intensive compared to our method as the size of the cuboid increases.

	Frames/second	Features/frame
STIP [45]	1.7	24.3
BRISK+MBH	6.96	183.5

Table 4.6: Computational complexity comparison of STIP and BRISK+MBH descriptor on Hollywood2.

4.5 Summary

In this chapter, a semi-binary video descriptor in the context of activity recognition has been presented. The proposed approach consists of four phases: BRISK feature detection, motion estimation, appearance and motion modelling. The proposed efficient video representation demonstrates comparable performance in two popular, widely used datasets with significantly reduced computational requirements. While most of the video descriptors are restricted to academic research due to their complexity, this descriptor demonstrates a potential for real world applications due to the greatly reduced computational requirements compared to other popular spatio-temporal interest point techniques. This descriptor is not only limited to human activity recognition, but can also be used in applications such as on-demand video retrieval, where computational complexity is a main priority.

Chapter 5

Multiple Instance Dictionary Learning for Activity Representation

5.1 Introduction

This chapter investigates several multiple instance learning techniques and presents an effective feature representation method for a local feature-based action recognition framework. Efficient and effective feature representation plays a crucial role, not only in activity recognition, but also in a wide range of applications such as motion analysis, tracking, 3D scene understanding *etc*. While spatio-temporal features are popular for analysing videos and have achieved state-of-the-art performance with low computational requirements, their performance is still limited for real world applications due to a lack of contextual information and models not being tailored to specific activities.

Traditional classification tasks consider entire image/video as a single entity and completely ignore the important semantic meanings arising from its constituent regions. They can be considered as single-instance single-label (SISL) problems where each example is assumed to have a single instance associated with a single label. On the other hand, Multiple instance learning is a newly proposed framework, where each example is associated with multiple instances and one or more labels. Several applications, such as image classification, protein synthesis, text classification, *etc*. have already explored the applicability of multiple instance learning in different machine learning problems such as classification, clustering and regression and demonstrated better performance. Multiple instance learning provides flexibility to formulate complex, real-world problems with different techniques such as Multiple-instance multi-label learning (MIML), Multiple-instance single-label learning (MISL) and Single-instance single-label learning (SISL) approaches. Under this framework, an image can be partitioned into several patches and represented with separate instances and each entity can be associated with multiple class labels. For example an image containing 'car' and 'cloud' can be partitioned into two separate instances with two separate class labels such as car and cloud and in the same-way, a document may contain several sections and can be treated as separate instances and associated with different topics such as fiction, non-fiction, comedy, *etc*.

In this chapter, we focus on Multiple-instance learning or multiple-instance single-label (MISL) learning, because this technique learn a separate dictionary for each action class and improve the dis-criminality between classes. MISL was initially proposed by Dietterich *et al.* [18] to predict drug activities. Let $\chi = \mathbf{R}^d$ denote the instances space and learn the MISL function: $f_{MISL} : 2^{\chi} \rightarrow \{+1, -1\}$ from a set of training examples $\{(X_i, y_i) \leq i \leq N\}$, where $X_i \subseteq \chi$ is a set of instances $\{x_1^i, x_2^i, x_3^i, \ldots, x_{n_i}^i\}$ and $y_i \in \{+1, -1\}$ is the label of X_i. Several multiple-instance learning algorithms have been explored and successfully applied

in object recognition and image retrieval applications [15, 59, 63, 113, 114]. This chapter investigates several state-of-the-art, multiple instance learning algorithms (*i.e.* MISL learning techniques have been explored because this thesis focusses on improving the classification performance of single actions from complex videos, therefore each video is associated with a single label) to effectively cluster and encode the local features to boost the overall classification performance.

On the other hand MIML learns the function: $f_{MIML} = 2^{\chi} \rightarrow 2^{\mathcal{Y}}$ from a set of examples $\{(X_i, Y_i) \leq i \leq N\}$ where $X_i \subseteq \chi$ is a bag of instances $\{x_1^i, x_2^i, x_3^i, \ldots, x_n^i\}$ and $\mathcal{Y} = \{1, 2, 3, \ldots, P\}$, $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^i, y_2^i, y_3^i, \ldots, y_l^i\}$ corresponds to X_i. Number of instances in X_i is n_i and number of labels in Y_i is l_i. MIML framework has been applied in multi-label learning frameworks [63, 84]. Single instance multi-label learning (SIML) or multi-label learning learns a function $F_{SIML} : \chi \rightarrow 2^{\mathcal{Y}}$ from a set of training samples $\{(x_i, Y_i) \leq i \leq N\}$, where $x_i \in \chi$ is an instance and $Y_i \subseteq \mathcal{Y}$ is a set of labels $\{y_1^i, y_2^i, y_3^i, \ldots, y_l^i\}$ corresponds to x_i. SIML techniques have been widely used to categorize images and text documents [13, 26, 91].

5.2 Motivation and proposed Approach

Even though local feature based systems produce superior classification performance in the context of activity recognition, the underlying bag-of-features-based representation to consolidate the local features imposes several drawbacks. This framework fails to capture underlying spatial and temporal relationships. Furthermore, a simple bag-of-features fails to incorporate the relationship between action categories. This is due to the clustering phase, where the method considers the entire feature space as a whole to build the vocabulary: *i.e.* one dictionary is built for all activity classes, which leads to an inappropriate feature allocation.

Also, clustering approaches suffer from initialization and inappropriate allocation of clusters to action categories (*i.e.* some unique features corresponding to a given activity may not have their own cluster, and instead are allocated to a different cluster, which predominantly contains features from different activities). The use of Vector Quantization (VQ) and the Euclidean distance is used to assign each feature to one element in the codebook, leading to large quantization errors and ignoring the relationship between different bases.

In this chapter, to address the above mentioned challenges to some extent, we propose a new activity representation framework based on different multiple instance learning techniques. As shown in Figure 5.1, Multiple instance learning technique replaces the popular k-means clustering in local action recognition framework. The following three multiple instance techniques have been investigated and proposed for local feature-based action recognition systems.

- **mi-SVM + K-means Approach:** Similar to [98], instead of learning a single codebook for all action classes using K-means, we learn a separate codebook for each activity class using Multiple Instance SVM (mi-SVM) and k-means clustering. In this approach, we treat multiple instance learning and mixture modelling as two separate steps. Given a set of training videos, dense histogram of oriented gradients (HOG) and histogram of optical flow (HOF) features are extracted. Then, one activity class is treated as positive and all the features (instances) from the target activity are assigned to a set of positive bags, and the rest of the classes are treated as negative and their features (instances) are assigned to negative bags. Then SVM is computed on positive and negative bags to identify the positive features in the positive bags, as shown in Figure 5.2. Finally K-means is used to cluster the positive instances. This process is repeated for each action class and a unique codebook is generated for each activity class. In contrast to VQ

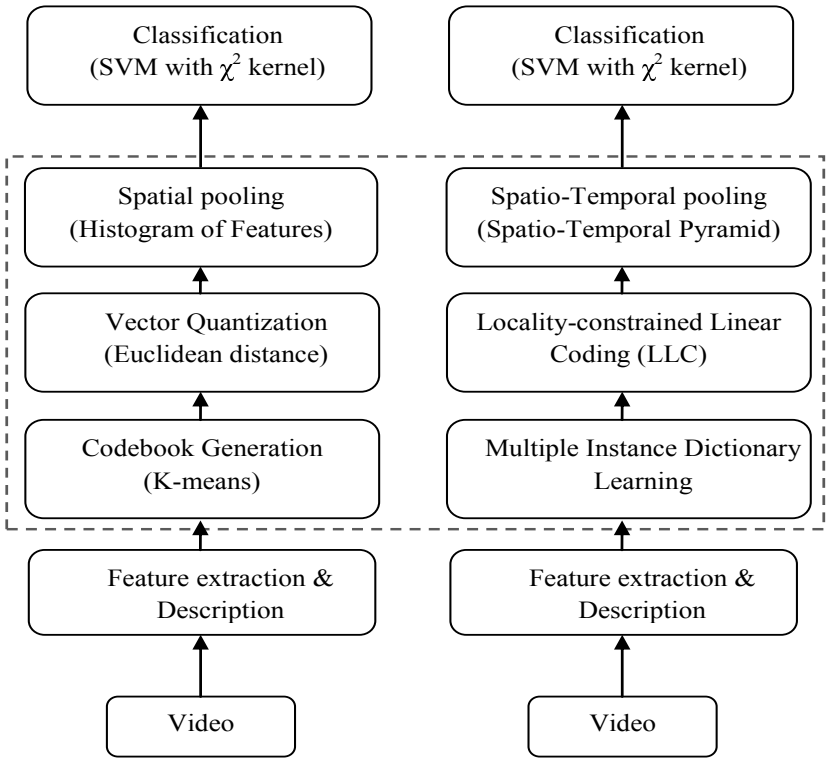


Figure 5.1: Schematic diagram of the popular bag-of-feature representation (Left) and our proposed feature representation (Right) in the context of activity recognition.

based feature encoding, locality constrained linear coding (LLC) is used to represent each input feature with multiple elements of the codebook. Finally, spatio-temporal pyramid pooling is used to capture the contextual information. This feature representation method demonstrates significant performance improvement over the popular bag-of-features method in two popular datasets.

- **Max Margin Dictionary Learning (MMDL) Approach**: This approach is similar to the above mentioned approach, but instead of separately performing multiple instance learning and mixture modelling, two steps are carried out simultaneously [98]. This representation produces best results compared to 'mi-SVM + K-means' and bag-of-features representation.

- **Max Margin Multiple Instance Clustering (M^3IC) Approach:**

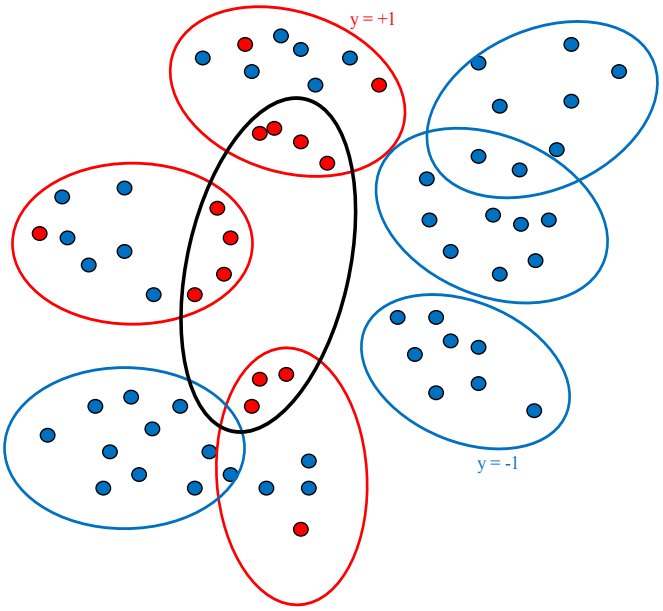


Figure 5.2: Illustration of mi-SVM to separate the instances in positive bags. A video (bag) is represented as a collection of features (instances), the bag is labelled positive if at least one of the instances (red) in the bag is positive and the bag is regarded negative if all instances (blue) are negative. mi-SVM aims to find the positive instances in the positive bags by maximizing the margin between positive and negative instances (the black ellipse denotes instances identified as positive by mi-SVM). Then, k-means is used to cluster the positive instances.

> M^3IC was proposed by Zhang *et al.* [117] to clustering images. The M^3IC approach has been incorporated to replace k-means clustering and to a create a new compact video representation suitable for classification. This representation demonstrates superior performance compared to k-means algorithms but lower performance compared to class-specific dictionaries built with 'miSVM + K-means' and 'MMDL' approaches.

The remainder of this chapter is organized as follows: Section 5.3 reviews related representation and encoding techniques. Section 5.4 provides details of the multiple instance learning techniques. Experimental framework and results for various datasets with different representation techniques are presented in Section 5.5. Finally, Section 5.6 summarises the chapter.

5.3 Related work

Several improved representation methods have been proposed in literature to improve local feature-based action recognition accuracy. Kovashka *et al.* [41] learnt class-specific distance functions that form the most informative configurations rather than dictate a particular scaling of the spatial and temporal dimensions. Zhang *et al.* [118] used sparse coding to quantize the features and a spatio-temporal pyramid is used to represent an action. Recent advances in machine learning approaches using multiple instance learning resulted in several advanced clustering algorithms. M^3MIL proposed by Zhang *et al.* [119] and M^3IC proposed by Zhang *et al.* [117] try to maximize the bag-level margin, while Xinggang *et al.* [98] proposed a method to maximize the instance level margin with multiple instance learning constraints. In this chapter, three state-of-the art clustering algorithms [98, 117] were developed based on multiple instance learning for local feature-based human action recognition and demonstrate significantly improved performance compared to the baseline.

For a given codebook, each feature is encoded with a single codebook element or multiple elements (*i.e.* distribution) and the final video representation is obtained by combining all the encoded feature vectors. Vector Quantization (VQ) is popularly used to encode the features into codebook elements. Yang *et al.* [108] proposed sparse coding (SC) instead of VQ to obtain non-linear codes. To improve the locality compared to the sparsity for successful non-linear codes, Local Coordinate Coding (LCC) was proposed by Yu *et al.* [111]. Locality-constrained linear coding (LLC) proposed by Jinjun *et al.* [99] is a fast implementation of LCC that adopts sparse coding (SC) and projects each descriptor into its local-coordinate system. This representation is highly robust and discriminative compared to vector quantization. In our proposed framework, we incorporate LLC to encode features into the generated library. Finally, spatio-temporal pyramid

pooling is applied to capture the informative spatio-temporal statistics.

5.4 Proposed method

As shown in Figure (5.1), the proposed method consists of four steps. In the first step, each video is densely sampled at different scales and each patch is described using HOG and HOF descriptors. In the second step, several multiple instance learning (see Figure 5.2) techniques have been investigated to learn robust, highly discriminative dictionaries (Dictionaries are learned separately for each action classes in miSVM + kmeans and MMDL approaches and the common dictionary is learned using M^3IC approach). Afterwards, LLC is used to encode each feature vector as a combination of multiple elements in the codebook, which achieves a better representation than Vector Quantization (VQ) because it captures the correlation between descriptors. Then, a spatio-temporal pyramid is used to pool multiple codes from each sub region. Finally, histograms from each subregion are concatenated to form the final descriptor for classification.

5.4.1 Feature Extraction

Dense sampling is used to extract video blocks at regular positions and different scales in space and time. The HOG descriptor encodes the appearance, while the HOF descriptor describes the local motion in the sampled patches. The histograms are created by accumulating space-time neighbourhoods of interest points. Each cuboid region is subdivided into an $n_x \times n_y \times n_t$ grid of cells. For each cell, a 4-bin HOG histogram (four directions) and a 5-bin HOF histogram (four directions and an additional bin for no motion) are calculated. Cell histograms are normalised and combined into a HOG/HOF descriptor. The original

implementation available on-line[1] and standard parameter settings are used.

5.4.2 Multiple Instance Dictionary Learning

In MIL based dictionary learning, each video is considered as a bag and features generated from the video are treated as instances corresponding to that bag. In the MIL problem, given a set of bags $X = \{X_1, X_2, ..., X_n\}$, each bag contains a set of instances $X_i = \{x_{i1}, x_{i2}..., x_{im_i}\}$, where m_i is the total number of instances in this bag. Each instance corresponds to a d-dimensional feature vector extracted from a video, $x_{ij} \in \mathbb{R}^{d\times 1}$. Each instance is associated with a instance level label $y_{ij} \in \{0, 1\}$; and the bag is associated with a bag level label, $Y_i \in \{0, 1\}$. The basic assumption of MIL, is that a bag is positive if at least one of the instances in that bag is positive (the true positive instance inside a positive bag is referred to as the "witness" or the "key"). On the other hand, the bag is considered negative if all instances inside the bag are negative. The MIL assumption can be summarized as follows,

$$Y_i = \begin{cases} 1 & \text{if } \exists j \text{ s.t } y_{ij} = 1, \\ 0 & \text{if } \forall j \text{ s.t } y_{ij} = 0. \end{cases} \tag{5.1}$$

Hence the key challenge in MIL is to cope with the ambiguity of not knowing which of the features in a positive bag are the actual positive features that indicate the presence of the target event. For example, the KTH dataset [86] consists of six action categories. If the 'running' class is treated as the positive class then all other actions are deemed negative, despite other events such as 'walking' and 'jogging' potentially having features in common with 'running'. The goal of MIL

[1] http://www.irisa.fr/vista/Equipe/People/Laptev/download.html#stip

is to find the actual positive features present in the positive bags for each action category separately.

Given the positive and negative bags, mi-SVM [6] is used to learn actual positive instances inside the positive bags, hence eliminate common instances present in multiple classes. Then we compute k-means on the positive instances identified by mi-SVM to generate a codebook for a particular action class. This process is repeated for all activity classes to generate a unique dictionary for each action class. This approach is referred to as 'mi-SVM + kmeans' in experiments.

5.4.3 M^{3}IC Approach

Most of the clustering methods try to find a clustering solution via single instance clustering, while the same problems can be better solved as a MIL problem. For an example in a given action video only a portion of the video contains a particular activity while most of the region may be irrelevant for the activity. In the multiple instance clustering (MIC) approach, each video is treated as a bag and each instance in this bag represents different regions in the video. The MIC approach helps to partition those bags automatically and has been successfully applied in text clustering and drug clustering applications.

In the M^3IC approach, the dataset is partitioned into k clusters, in such a way that each cluster represents different characteristics and is distinct from each other. Each cluster has its own weight vector w_p. f_i represents the cluster assignment for bag X_i. Rather than running an SVM on all possible clusters like in a Max-margin clustering (MMC) [106] approach, in the M^3IC approach bags are clustered using several large margin classifiers that maximize margins on bags and bag margin associated with the bag X_i defined as follows,

$$\max_{j \in X_i}(\mathbf{w}_{u_{ij}^*}^\intercal \mathbf{X}_{ij} - \mathbf{w}_{v_{ij}^*}^\intercal \mathbf{X}_{ij}), \tag{5.2}$$

where $u_{ij}^* = \text{argmax}_p(\mathbf{w}_p^\intercal \mathbf{X}_{ij})$ and $v_{ij}^* = \text{argmax}_{p \neq u_{ij}^*}(\mathbf{w}_p^\intercal \mathbf{X}_{ij})$. *i.e.* most discriminative instance determines the bag margin and the M^3IC learning approach is formulated as follows:

$$\begin{aligned} \min_{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w_3}, \ldots, \mathbf{w}_k, \xi_i \geq 0} & \frac{1}{2} \sum_{p=1}^{k} \parallel \mathbf{w}_p \parallel^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i \\ s.t. \qquad & i = 1, \ldots, n, \\ \max_{j \in X_i}(\mathbf{w}_{u_{ij}^*}^\intercal \mathbf{X}_{ij} - \mathbf{w}_{v_{ij}^*}^\intercal \mathbf{X}_{ij}) & \geq 1 - \xi_i \\ \forall p, q \in \{1, 2, 3, \ldots, k\} & \\ -l \leq \sum_{i=1}^{n} \sum_{j \in X_i} I_{ij} \mathbf{w}_p^\intercal \mathbf{X}_{ij} - \sum_{i=1}^{n} \sum_{j \in X_i} I_{ij} \mathbf{w}_q^\intercal \mathbf{X}_{ij} & \leq l \end{aligned} \tag{5.3}$$

Where

$$I_{ij^*} = \begin{cases} 1 & if j^* = \text{argmax}_{j \in X_i}(\mathbf{w}_{u_{ij}^*}^\intercal \mathbf{X}_{ij} - \mathbf{w}_{v_{ij}^*}^\intercal \mathbf{X}_{ij}) \\ 0 & Otherwise \end{cases}$$

parameter l is used to control the cluster balance to avoid trivially optimal solutions. The bag X_i can be assigned to a specific cluster based on $f_i = \text{argmax}_p I_{ij} \mathbf{w}_p^\intercal \mathbf{X}_{ij}$. However the optimization problem in 5.4 is difficult to solve due to two constraints. In the first constraint *i.e.* $\max_{j \in X_i}(\mathbf{w}_{u_{ij}^*}^\intercal \mathbf{X}_{ij} - \mathbf{w}_{v_{ij}^*}^\intercal \mathbf{X}_{ij}) \geq 1 - \xi_i$, the convexity of $\mathbf{w}_{v_{ij}^*}^\intercal \mathbf{X}_{ij}$ is unknown and the second constraint becomes non convex due to the indication function I_{ij}. These constraints are relaxed and $M^3IC - MBM$ [117] is used to solve the resulting optimization problem.

5.4.4 MMDL Approach

In the MMDL approach [98], max-margin classifier is learned to classify all features into different clusters and learned classifiers (G-codes) are used as the video representation for classification. MMDL uses multi-class SVM to maximize the margins between different clusters. Each cluster is associated with a linear classifier $f(x) = \mathbf{w}^\intercal \mathbf{x}$. In MMDL the latent variable $z_{ij} \in \{0, 1, 2, \ldots, K\}$ is assigned to each instance and $z_{ij} = k \in \{1, \ldots, K\}$ if instance x_{ij} is in the k^{th} positive cluster; otherwise $z_{ij} = 0$, x_{ij} is in negative cluster. Moreover, a weighting matrix $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K], \mathbf{w}_k \in \mathbf{R}^{d\times 1}$ is defined as linear classifiers, where $\mathbf{w}_k$ represents the k^{th} cluster model and $\mathbf{w}_0$ denotes the negative cluster model. Finally the instance (*i.e.* feature vector) x_{ij} is assigned to the latent variable z_{ij} using the following formula,

$$z_{ij} = \text{argmax}_k \mathbf{w}_k^\intercal x_{ij}. \tag{5.4}$$

The objective function is defined as follows,

$$\begin{aligned} &\min_{\mathbf{W}, z_{ij}} \sum_{k=0}^{K} \|\mathbf{w}_k\|^2 + \lambda \sum_{ij} max(0, 1 + \mathbf{w}_{r_{ij}}^\intercal x_{ij} - \mathbf{w}_{z_{ij}}^\intercal x_{ij}) \\ &\text{s.t. if} \quad Y_i = 1, \sum_j z_{ij} > 0, \quad \text{and if} \quad Y_i = 0, z_{ij} = 0, \end{aligned} \tag{5.5}$$

where $r_{ij} = \text{argmax}_{k\in\{0,\ldots,K\}, k\neq z_{ij}} \mathbf{w}_k^\intercal x_{ij}$. In Equation 5.6, the term $\sum_{k=0}^{K} \|\mathbf{w}_k\|^2$ is used for margin regularization and second term is multi-class hinge-loss. Parameter λ controls the significance of the second term relative to the first term. The objective function 5.6 is a non-convex optimization problem and becomes convex if the latent information of the instances in the positive bags and number

of positive instances in each of the positive bags are known. Since we don't have both pieces of information, the optimization problem becomes harder to solve. Interested readers are referred to Xinggang *et al.* [98] for more details about optimization.

5.4.5 LLC Feature Encoding

In the feature encoding phase, D-dimensional feature descriptors, *i.e.* $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{D\times N}$, extracted from videos, are mapped to a codebook $B = [b_1, b_2, ...b_M] \in \mathbb{R}^{D\times M}$, of length M. Though several coding methods exist in literature vector quantization (VQ) is the most popular method used in action recognition. VQ solves the following least square fitting problem,

$$\begin{aligned} \arg\min_{c} \sum_{i=1}^{N} \parallel x_i - Bc_i \parallel^2, \\ s.t. \parallel c_i \parallel_{l^0} = 1, \parallel c_i \parallel_{l^1} = 1, c_i \succeq 0, \forall i, \end{aligned} \tag{5.6}$$

where $C = [c_1, c_2.., c_N]$ is the set of codes for a video. Since this method only finds a single nearest neighbour it generates large quantization errors. In addition, VQ ignores the relationship between different bases and we need expensive non-linear kernel projections to improve the recognition accuracy. To improve the quantization errors and to obtain a non-linear representation, Sparse coding Spatial Pyramid Matching (ScSPM) [108] was proposed. In scSPM, the coding problem becomes a standard sparse coding (SC) problem,

$$\arg\min_{c} \sum_{i=1}^{N} \parallel x_i - Bc_i \parallel^2 + \lambda \parallel c_i \parallel_{l^1}. \tag{5.7}$$

In the SC approach, the sparsity regularization term allows the learned representation to capture salient patterns of local descriptors and achieve much lower quantization error compared to VQ. In this framework, Locality-constrained Linear Coding (LLC) is adopted, which treats locality as more important than sparsity as locality leads to sparsity. The LLC optimization goal is as follows,

$$\arg\min_{c} \sum_{i=1}^{N} \| x_i - Bc_i \|^2 + \lambda \| d_i \odot c_i \|^2. \tag{5.8}$$

The second term represents element-wise multiplication, and d_i is the locality adaptor that gives different freedom for each basis vector proportional to its similarity to the input vector, x_i. Compared to VQ, SC and LLC minimize the quantization error by representing an input with multiple elements from the codebook. Furthermore, LLC captures locality information and correlation between similar descriptors.

5.4.6 Spatio-Temporal Pooling

We adapt the spatial pyramid matching (SPM) [47] approach for spatio-temporal pooling, which considers temporal information in conjunction with spatial locations to encode the spatio-temporal relationship. The spatio-temporal pyramid partitions a video into 3D grids in space and time, and calculates the weighted sum of codes in each sub region. The video is partitioned into increasingly finer sub-regions, and computes histograms of local features for each sub region. We use $2l \times 2l \times l$ sub regions, where $l = 0, 1, 2$. The video is first viewed as a whole, then, in the second level it is segmented into four sub regions spatially without any temporal segmentation. In the third level, each part in the previous level is partitioned into four sub-regions spatially and two sub-regions temporally.

The final descriptor is formed by concatenating all histograms from each sub-region.

5.5 Experiments and Results

In experiments, each video is densely sampled into 3D patches with different scales of $18 \times 18 \times 9$, $36 \times 36 \times 12$ and $48 \times 48 \times 15$. Spatial and temporal sampling is done with 30% overlap. For each sampled cuboid, HOG and HOF features are extracted, as described in Section (5.4.1). We compare three proposed feature representations based on multiple instance learning with the popular bag-of-feature based representation.

Finally the classification is done with a non-linear support vector machine with a χ^2 kernel as shown in Figure 3.1,

$$K(H_i, H_j) = exp\big(-\frac{1}{\alpha} D(H_i, H_j)\big), \tag{5.9}$$

where H_i and H_j are the histograms of word occurrences, $D(\cdot)$ is the χ^2 distance defined by,

$$D(H_i, H_j) = \frac{1}{2} \sum_k \frac{(H_i(k) - H_j(k))^2}{H_i(k) + H_j(k)}, \tag{5.10}$$

and α is the average distance between all training examples.

A 'one against the rest' approach is used and the class with the highest score is selected.

Experiments were carried out with two popular benchmark datasets with varying

Approach	Average Accuracy
M^3IC Approach	89.3%
MMDL Approach	93.7%
mi-SVM + k-means Approach	92.83%
Wang *et al.* [96]	86.10%
Laptev *et al.* [45]	91.8%
Xiaojing *et al.* [118]	92.59%
Niebles *et al.* [69]	81.50%

Table 5.1: Comparison of recognition accuracy on the **KTH** Dataset using different approaches. Different feature descriptors were used in [118] and [69].

complexity: KTH and Hollywood2. KTH is selected to demonstrate the effectiveness in the static environment and Hollywood2 is selected to demonstrate the effectiveness in the complex environment. The KTH [86] dataset was recorded in a well-controlled environment with a single person performing the action with a clean background, and on average each video lasts for 20 seconds. The Hollywood2 [60] dataset consists of actions taken from movies, where complicating factors such as complex scenes with a moving background, illumination changes, multiple actors and camera motion are present.

5.5.1 KTH

Table 5.1 shows comparison of recognition accuracy on the **KTH** Dataset using different approaches. Figure 5.3 shows the confusion matrix obtained with different representations on the KTH dataset with dense HOG+HOF descriptors. In the k-means approach, similar to [45, 96], 100,000 random training features are chosen and the code book learnt with the number of clusters set to $k = 4000$. Then vector quantization is used to assign each feature to its closest codeword followed by a histogram of visual word representation. From the confusion matrices, it is obvious that MIL based representations clearly outperform the baseline

	Running	Boxing	Walking	Jogging	Waiving	Clapping
Running	0.76	0.00	0.18	0.06	0.00	0.00
Boxing	0.00	0.91	0.00	0.00	0.03	0.06
Walking	0.02	0.00	0.94	0.04	0.00	0.00
Jogging	0.09	0.00	0.08	0.83	0.00	0.00
Waiving	0.00	0.02	0.00	0.00	0.86	0.12
Clapping	0.00	0.07	0.00	0.00	0.02	0.91

(a) K-means Algorithm

	Running	Boxing	Walking	Jogging	Waiving	Clapping
Running	0.82	0.00	0.14	0.04	0.00	0.00
Boxing	0.00	0.98	0.00	0.00	0.00	0.02
Walking	0.00	0.00	1.00	0.00	0.00	0.00
Jogging	0.05	0.00	0.04	0.91	0.00	0.00
Waiving	0.00	0.00	0.00	0.00	0.94	0.06
Clapping	0.00	0.08	0.00	0.00	0.00	0.92

(b) miSVM + K-means Algorithm

	Running	Boxing	Walking	Jogging	Waiving	Clapping
Running	0.82	0.00	0.15	0.03	0.00	0.00
Boxing	0.00	0.92	0.00	0.00	0.02	0.06
Walking	0.02	0.00	0.94	0.04	0.00	0.00
Jogging	0.06	0.00	0.08	0.86	0.00	0.00
Waiving	0.00	0.01	0.00	0.00	0.89	0.10
Clapping	0.00	0.05	0.00	0.00	0.02	0.93

(c) M^3IC Algorithm

	Running	Boxing	Walking	Jogging	Waiving	Clapping
Running	0.85	0.00	0.11	0.04	0.00	0.00
Boxing	0.00	0.97	0.00	0.00	0.00	0.03
Walking	0.00	0.00	0.99	0.01	0.00	0.00
Jogging	0.02	0.00	0.04	0.94	0.00	0.00
Waiving	0.00	0.00	0.00	0.00	0.94	0.06
Clapping	0.00	0.07	0.00	0.00	0.00	0.93

(d) MMDL Algorithm

Figure 5.3: Average classification accuracy of different feature representation methods applied on *KTH* Datasets with Dense HOG/HOF descriptor.

Approach	mean AP
M^3IC Approach	47.4%
MMDL Approach	52.3%
mi-SVM + k-means Approach	51.8%
Wang *et al.* [96]	47.4%
Laptev *et al.* [45]	45.2%
Le *et al.* [48]	53.3%
Wang *et al.* [94]	58.2%

Table 5.2: Comparison of mean Average Precision (mAP) on the **Hollywood2** Dataset using different approaches. Different feature descriptors were used in [94], [48]

in all action categories and improve the overall accuracy, which indicates the importance of efficient feature representation in addition to the actual feature itself.

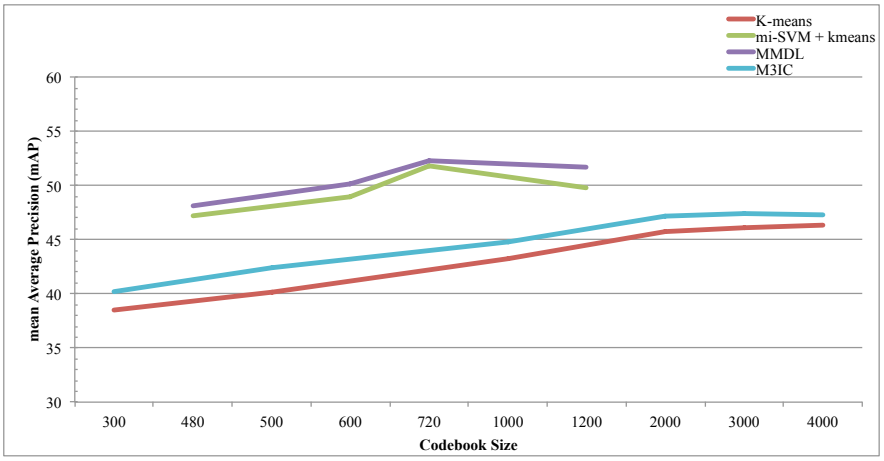


Figure 5.4: Performance comparison of several Multiple instance learning (MIL) techniques against the k-means clustering approach with varying codebook sizes in **Hollywood2** dataset.

5.5.2 Hollywood2

The Hollywood2 dataset is presented in Section 2.5.3. Mean Average Precision over all classes (mAP) is reported as a performance measure [60]. As shown in Table 5.2, our proposed MIL based feature representations outperform the baseline k-means in Hollywood2 dataset with dense HOG+HOF descriptors. Similar to the KTH dataset, class-specific codebooks generated by 'MMDL' and 'mi-SVM + k-means' achieved superior performance compared to a single codebook generated by M^3IC and k-means algorithms. Other methods [48, 94] proposed different feature descriptors such as hierarchical spatio-temporal features and dense trajectories to improve the classification performance, where the actual features itself contribute towards the performance improvement in contrast to the improvement from advanced feature representation.

Figure 5.4 demonstrates the efficiency of our proposed representations in a more complex dataset, Hollywood2. Similar to KTH, all codebooks in the bag-of-features representation are outperformed by our representations, which also allows us to represent each activity with a smaller codebook size. Peak performance is

obtained in 'mi-SVM + k-means' and 'MMDL' with a codebook size of 720 (*i.e.* 12 action classes, 60-codes per class) and M^3IC consistently outperforms across all codebook sizes.

5.6 Summary

In this chapter, another new feature representation framework based on Multiple instance learning technique is presented. Three popular MIL techniques have been investigated and they outperform the popular bag-of-features based method to represent videos in local feature-based activity recognition systems. Experimental results validate the effectiveness of the feature representation. This demonstrates that a multiple instance dictionary learning method can serve as a potential replacement for the popular bag-of-features method and it helps to further boost the performance of the state-of-the art descriptors. In the mean-time, class-specific codebooks generated by 'mi-SVM + k-means' and 'MMDL' approaches not only provide compact, discriminative representation but also achieve memory efficiency. In the MIL learning approach, since each code word in the codebook is represented as a linear-classifier, it involves only a dot product operation to encode patch level features. This time is almost negligible with modern computers with lots of GPU power. MIL representation not only generates compact codebook but also captures rich semantic information from the patch level features.

Chapter 6

LDA Based Local Feature Representation

6.1 Introduction

Most of the vision-based human action systems consist of three basic phases: 1) Encode the appearance and motion information from the videos as a set of features, 2) Reduce the dimensionality of the extracted features while retaining the discriminative power, 3) Classify using either generative or discriminative methods. Probabilistic generative models consider the activity as a sequence of states while discriminative methods ignore the order of features during classification. The classification performance of the local feature-based action recognition systems not only depend on the effective video features but also depend on converting the features appropriate for classification. The popular Bag-of-visual words model suffers from various challenges, such as assignment of each feature descriptor to a single dictionary element, which is inadequate to capture the relationship with other dictionary elements.

This chapter addresses the problem of efficiently representing the extracted features for classification to improve the overall performance. Latent Dirichlet Allocation (LDA) has recently gained popularity to project a large amount of documents into a lower dimensional space spanned by a set of *topics*, which capture the semantic characteristics of the document. For a given dictionary of words, LDA models uses soft-assignment to assign each feature descriptor to many dictionary elements by a mixture probability over words as opposed to hard assignment. This representation is appropriate because of the following reasons:

- When a new test video is presented with a new set of features it would be effectively modelled with a mixture of words rather than finding a single closest dictionary element.
- Assignment of a feature to a single descriptor leads to higher quantization errors compared to a probabilistic mixture of dictionary elements. This representation not only improves the efficacy of the representation but also contributes towards the performance improvement.

Several LDA models have been investigated to efficiently capture the spatio-temporal relationships and to improve the overall classification performance. In this piece of work, the focus has been given to replace 'vector quantization followed by sum-pooling' in the bag-of-words framework with the latent topic vector obtained from different LDA Models.

6.1.1 Motivation & Proposed Approach

Latent Dirichlet Allocation (LDA) was introduced by Blei *et al.* [11] and recently gained popularity to classify collections of documents and images into a low dimensional space spanned by a set of topics, which capture the semantic aspects of

the documents. Each document is represented as a mixture of topics, known as a topic vector, which is modelled as a latent Dirichlet random variable and provides a low dimensional representation for tasks such as classification, summarization and clustering.

The generative process in the LDA assigns a number of topics, where each document is sampled from a mixture of topics, and defined by some unique multinomial probability over the words in the dictionary. When fitting a corpus of documents with the LDA model, the topics which are discovered often reveal insightful information about the relations and shared structure between documents.

Similar to LDA, Probabilistic latent semantic indexing (pLSI) was introduced by Hoffman *et al.* [32] and models each word in a document as a sample from a mixture model, and mixture components are random variables that can be viewed as topics. The pLSI approach suffers from a number of problems, such as the model parameters increasing with the number of training samples and creating over fitting problems and difficulty in assigning probability to a document out of the training sample set. On the other hand in LDA, the k-topic LDA model doesn't grow with the number of training samples and is not prone to overfitting problems.

In addition to several advantages of LDA over other topic models, recently several works in image and text classification demonstrated that incorporating a supervised approach to the feature representation improves the discriminative power and overall classification accuracy. Several max-margin-based techniques such as max-margin dictionary learning [98, 117, 119] and supervised LDA techniques [62, 75, 79, 120] have significantly improved the classification accuracy in image and text classification. Unsupervised LDA models disconnect topic discovery from the classification task, hence yield poor results compared to the baseline Bag-of-words framework. On the other hand, supervised LDA techniques learn

the topic structure by considering the class labels and improve the recognition accuracy significantly. This motivates us to investigate several supervised LDA techniques for the local feature-based action recognition system.

In this work, several supervised topic models have been explored and two generative supervised topic models, maximum entropy discrimination LDA (MedLDA) and class-specific simplex LDA (css-LDA), have been proposed as an alternative for activity representation, incorporating valuable class label information during topic discovery and representation. The first representation is based on MedLDA [120], a supervised LDA model incorporating both the max-margin principle and maximum likelihood function over the data to generate a more discriminative latent topic representation. The second representation is based on css-LDA [79], which learns multiple class-specific topic simplexes rather than a single set of topics for the entire dataset by introducing supervision at the feature level. MedLDA maximizes likelihood and within class margins using max-margin techniques and yields a sparse highly discriminative topic structure; while in css-LDA, separate class specific topics are learned instead of a common set of topics across the entire dataset.

Simultaneously learning the optimal dictionary and topics is a non-convex optimization problem. Therefore, in this proposed approach, a dictionary was learned using k-means and the descriptors have been appropriately modelled using a mixture of discovered topics. The dictionary learning has been done prior to LDA modelling and each dictionary consists of a mixture of feature vectors. Each video is represented as a topic proportion vector, *i.e.* it can be comparable to a histogram of topics. Finally a discriminative classifier, SVM, is applied on the learned topic proportion vector. The efficiency of the above two representation techniques has been demonstrated through the experiments carried out in two popular datasets. Experimental results demonstrate that both topic repre-

sentations significantly improve the overall classification accuracy in challenging datasets compared to the baseline bag-of-features and unsupervised LDA representation.

The reminder of this chapter is organized as follows: Section 6.2 presents several LDA models and their applications. Section 6.4 provides details of several LDA-based representations. Section 6.5 explains the experimental framework used in the experiments. Experimental results for various datasets are presented in the 6.6. Finally, Section 6.7 concludes the chapter.

6.2 LDA variations and Applications

Though LDA was originally developed as an unsupervised model which ignores class label information during topic discovery, since then several supervised LDA models have been proposed to incorporate class label information to discover more relevant and discriminative topics.

Supervised LDA (sLDA) was introduced by Blei *et al.* [62] and maximizes the joint likelihood of both the training data and the label information. DiscLDA [42] maximizes the conditional likelihood of the label information given the documents. Several other models that incorporate class label information at different stages of LDA exist, such as classLDA [23] for scene classification; labelled LDA [75] for credit attribution in multi-labelled corpora; correspondence LDA [10] for image annotation; and multi-class sLDA [93] for image classification.

Several LDA variants have been explored in the action classification domain as well. Niebles *et al.* [69] applied unsupervised pLSA and LDA to represent spatio-temporal words as intermediate topics for action classification. Wang *et al.* in-

troduced semi-LDA [95], as a semi-supervised way to represent human action in videos. This work is different from the above two approaches in two ways: (1) This chapter explores several supervised latent topic models to the application of human action recognition with discriminative classifiers rather than generative classifier models, because with larger datasets, discriminative classifiers yield stronger performance over generative classifiers. (2) Unlike other methods where the number of topics are set to the number of classes, in this work experiments have been carried out with a wide range of topic sizes to optimize the topic structure for a given feature and dataset.

In this work, after a comprehensive set of investigations, two recent, supervised LDA variants, MedLDA and css-LDA, are proposed for efficient video representation for the purpose of action classification. Recent work in [92, 98] shows improved recognition performance by incorporating max-margin dictionary learning. Inspired by these results, this chapter introduces another max-margin based LDA technique, MedLDA. MedLDA extends LDA to learn discriminative topics by employing a max-margin technique within the probabilistic framework. On the other hand, css-LDA [79] introduces the supervision at the feature level and enables class specific topic simplexes and class-specific topic distributions to capture much richer intra- class information, which provides more discrimination within the representation than a single set of topics for the entire data set.

6.3 Introduction to LDA

Latent Dirichlet Allocation (LDA) is an unsupervised, hierarchical Bayesian model and was initially proposed by Blei *et al.* [11] for text processing and has been successfully extended to several computer vision applications. This section presents a brief overview of the LDA model presented by Blei *et al.* In this model,

a corpus is considered to be a collection of documents, whereas each document is a collection of words. The following terms have been defined in the LDA model:

- A *word* is defined as an element from a vocabulary indexed by $\{1, \ldots, V\}$. *i.e.* a word is represented as a unit basis vector with a single non-zero element.
- A *document* is a sequence of N words represented by $\mathbf{w} = (w_1, w_2, w_3, \ldots, w_N)$, w_n is the nth words in the document.
- A *corpus* is a collection of M documents represented by $D = \{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \ldots, \mathbf{w}_M\}$

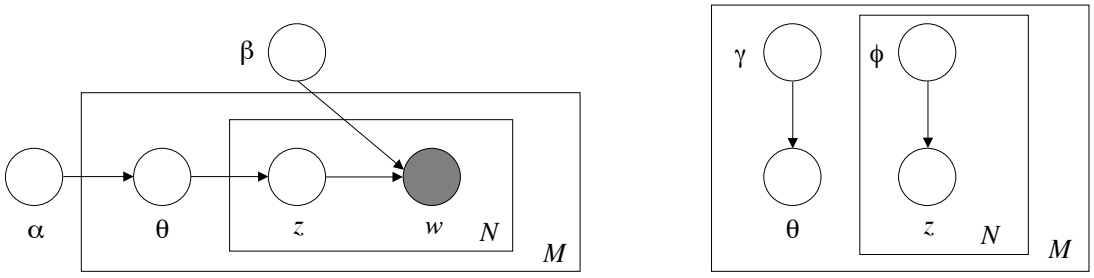


Figure 6.1: (Left) Graphical representation of LDA. (Right) Graphical model representation of the variational distribution used to approximate the posterior in LDA. [11]

In the LDA model, documents are modelled as a mixture of discovered latent topics and each topic is characterised by a multinomial distribution of words from a vocabulary. The graphical representation of the unsupervised model is given in Figure 6.1. The following generative process is used to model the documents.

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the words w_n:

(a) Choose a topic $z_n \sim Multinomial(\theta)$.

(b) choose a word $w_n \sim p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n

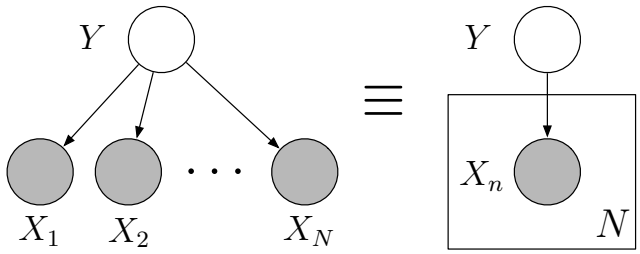


Figure 6.2: Graphical model representation and plate representation

Several assumptions have been made to simplify the basic LDA model. The following are the main assumptions used in the LDA models:

- The dimensionality of the topic variable z *i.e.* the dimensionality k of the Dirichlet distribution is known and fixed.
- The word probabilities are represented by $K \times V$ matrix β where $\beta_{ij} = p(w^j = 1|z^i = 1)$ assumed to be fixed and to be calculated.
- The LDA assumes the ex-changeability property in Bayes networks. Figure 6.2 shows the graph representation of a single layer Bayes network. The nodes are random variables, where the observed variables are shaded; the edges represent possible dependence; and a plate indicates replicated structure. The property of ex-changeability is also termed "conditional independence", indicating that with the condition of the variable of the parent node, the variables of the child nodes are independent and given by

$$P(X_1, X_2, X_3, \ldots, X_N|Y) = \prod_{n=1}^{N} p(X_i|Y). \tag{6.1}$$

In Figure 6.2, Y represents the document and X represents the words. Under the condition of the same document, the words are independent of each other. This ex-changeability in graphical models is referred to as the "bag of words" assumption in language processing. The "bag of words", is popularly used terminology in computer vision applications derived from text processing applications.

The plate notation of the LDA model is shown in the Figure 6.1, where M is the number of documents; N is the number of words in a document; K is the number of topics; w represents the words; β is a matrix which stores the word probabilities and z is the topic assigned for each word; α is the Dirichlet parameter and θ is the per document topic distribution, which is drawn from the Dirichlet distribution with parameter α. A k dimensional Dirichlet variable θ takes the values in $k-1$ simplex with the following probability density function,

$$p(\theta|\alpha) = \frac{\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)}{\prod_{i=1}^{k}\Gamma(\alpha_i)}\theta_1^{\alpha_1-1}\ldots\theta_k^{\alpha_k-1}, \tag{6.2}$$

where α is a k vector with components $\alpha_i > 0$, and the $\Gamma(x)$ is a Gamma function. The Dirichlet distribution is used to model the topic distribution because the Dirichlet distribution is the conjugate prior to the multinomial distribution to model word distribution of a topic, and providing convenience to the Bayes inference process. The α parameter is used to control the shape of the distribution, when the α_i is set to a constant the distribution becomes a symmetric Dirichlet distribution. Given a set of parameters α and β, the joint distribution of the topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is given by,

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha)\prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta). \tag{6.3}$$

The following marginal distribution is obtained by integrating over θ and summing over $\mathbf{z}$,

$$p(\mathbf{w}|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta)\right)d\theta_d. \tag{6.4}$$

Finally, the probability of a corpus is obtained by taking the product of the marginal probabilities of individual documents,

$$p(D|\alpha,\beta) = \prod_{d=1}^{M}\int p(\theta_d|\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta)\right)d\theta_d. \tag{6.5}$$

6.3.1 Inference and parameter Estimation

The only observable variable in the LDA model shown in Figure 6.1 is w. In the learning phase, the Expectation Maximization (EM) algorithm is used to train the parameters α and β. In the EM algorithm, the parameters α and β are initialized and then the following steps are performed:

- Based on α and β, maximized the posterior distribution of the hidden variable $p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta)$;
- Update α and β based on $p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta)$.

The key following inference problem needs to be solved to compute the posterior distribution of the hidden variable,

$$p(\theta,\mathbf{z}|\mathbf{w},\alpha,\beta) = \frac{p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta)}{p(\mathbf{w}\alpha,\beta)}. \tag{6.6}$$

The above distribution is intractable for exact inference and several approximate inference algorithms such as variational approximation, Markov chain Monte Carlo (MCMC) and Laplace approximation have been used. In this chapter similar to Blei *et al.* [11], Variational inference is used to infer the posterior distribution of latent variables $\{ \theta_d, z_d \}$ by maximizing the marginal likelihood of $p(w|\alpha, \beta)$. The family (See Figure 6.1) is characterised by the following variational distribution:

$$q(\theta, \mathbf{z}|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n), \tag{6.7}$$

where variational parameters are γ and θ. The following optimization procedure generates the parameters (γ^*, ϕ^*) which are function of w.

$$(\gamma^*, \phi^*) = argmin_{\gamma,\phi} D(q(\theta, \mathbf{z}|\gamma, \phi) \| p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \beta)). \tag{6.8}$$

More details can be found in [11].

6.4 Proposed Feature Representation Framework

The proposed method consists of four sections. In the first step, each video is densely sampled and HOG and MBH features have been extracted to capture both appearance and motion information. Then, extracted features are modelled into context aware topics using two supervised topic models: MedLDA and css-LDA. The discovered topics are treated as bases to represent each video in

the dataset as a low dimensional topic proportion vector. (*i.e.* In Bag-of-words representation the histogram of words is replaced with topic proportion vector θ in our framework). Finally, classification is done with a linear SVM classifier.

6.4.1 Feature extraction

Local features are extracted to represent each video. The appearance information is captured using Histogram Oriented Gradients (HOG) and the motion information is captured using the Motion Boundary Histogram (MBH). Each video is sampled using dense trajectories [94] [1], with default parameters.

For appearance, the HOG descriptor is calculated along the trajectory and the cuboid region is subdivided into a $2 \times 2 \times 3$ grid of cells. For each cell, an 8-bin HOG histogram is calculated and normalised into a HOG descriptor. The robust optical flow based MBH [94] descriptor is used to capture the motion information along the trajectories.

6.4.2 Latent Dirichlet Allocation for videos

In this section, a brief review of LDA model in the context of video representation is presented. Videos are treated as a random variable X, spanned by a feature space χ of visual measurements. In our case, the feature space is defined by both HOG and MBH features. Each video is represented as a set of N feature vectors $V = \{x_1, x_2, ..., x_N\}$, $x_n \in \chi$. Then the feature space is quantized into high dimensional n bins, defined by a set of cluster centroids, $C = \{c_1, c_2, ...c_n\}$ *i.e.* the vocabulary. Finally each feature x_n is mapped to the closest centroid and each video in the dataset is represented as a set of words, $V = \{w_1, w_2..., w_N\}, w_n \in C$,

[1] Code publicly available at http://lear.inrialpes.fr/people/wang/densetrajectories

where w_n is the bin (visual word) containing the feature x_n.

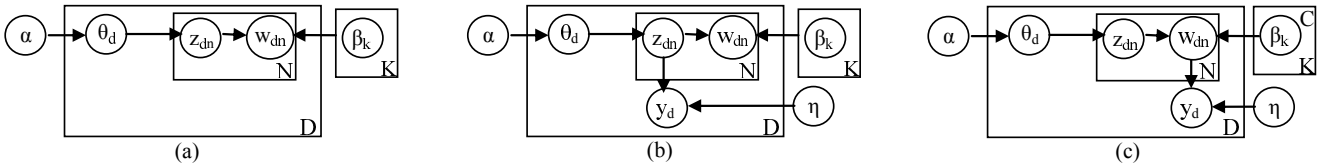


Figure 6.3: Graphical representation of LDA Models. (a) unsupervised LDA Model (b) MedLDA Model (c) css-LDA Model

Words in the LDA model are the same as the set of centroids or vocabulary, and each feature vector can be spanned by the vocabulary. Each video is treated as a document with N words and is denoted by $\boldsymbol{w} = (w_1, w_2 ... w_n)$, where w_n is the n^{th} word in the sequence. A corpus is the entire dataset consisting of M documents denoted by $D = \{\boldsymbol{w_1}, \boldsymbol{w_2 ...}, \boldsymbol{w_M}\}$. In LDA each topic is represented as a multinomial distribution over the vocabulary and each video is represented as a random mixture over the latent topics. LDA representation is shown in Figure 6.3. The parameters α and β remain the same for the entire dataset. The variable θ_d is a video-level parameter representing video specific topic distribution, sampled once per video. The variable per word topic distribution z_{dn} and the n^{th} word in the d^{th} document w_{dn} are word/feature level parameters sampled once for each feature.

The following generative process is applied by LDA for each video $\boldsymbol{w}$ in a corpus D.

1. Choose $N \sim Poisson(\xi)$.
2. Choose $\theta \sim Dir(\alpha)$.
3. For each of the words w_{dn}:
   (a) Choose a topic $z_{dn} \sim Multinomial(\theta)$. $z_{dn} \in T = \{1, 2, ..., K\}$

(b) choose a word $w_{dn} \sim p(w_{dn}|z_{dn}, \beta)$, a probability conditioned on the topic z_{dn}

Document level topic distribution θ_d has been used as the low dimensional representation of a video in experiments.

6.4.3 Supervised LDA (SLDA) and MedLDA Approach

Unsupervised LDA doesn't consider the class label information of the videos and supervised LDA models introduce a response variable, y, to each document as shown in Figure 6.3. Both label information and document content influence the topic learning in SLDA and MedLDA, whereas LDA uses the likelihood of document contents w. MedLDA is an extension of the SLDA model, and generates discriminative topics by directly optimizing both margin-based loss function and likelihood-based objective; while SLDA are only trained to optimize the likelihood objective. The following generative process is used in both SLDA and MedLDA approaches :

1. Choose a topic mixing proportion vector θ_d from a Dirichlet distribution with a parameter $\alpha : \theta_d|\alpha \sim Dir(\alpha)$
2. For each word w_{dn} in the document:
   (a) Choose a topic assignment $z_{dn} : z_{dn}|\theta_d \sim Multi(\theta_d)$, $z_n \in T = \{1, 2, ..., K\}$
   (b) Choose a word instance $w_{dn} : w_{dn}|z_{dn}, \beta \sim Multi(\beta_{z_{dn}})$.
3. Choose a response variable $y_d : y_d|z_{1:N}, \eta, \sigma^2 \sim \mathcal{N}(\eta^\top \bar{z}, \sigma^2)$; where $\bar{z} = \frac{1}{N_d} \sum_{n=1}^{N_d} zdn$ and the response parameters η and σ^2.

The joint distribution of the SLDA is given by the following:

$$p(y, w|\alpha, \beta, \eta, \sigma^2) = \prod_{d=1}^{D} p(\theta_d|\alpha) \left(\prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta) \right) p(y_d|\eta^\top \bar{z}, \sigma^2) \quad (6.9)$$

In sLDA the unknown constants $\alpha, \beta_{1:K}, \eta$ and σ^2 are estimated by maximizing the joint likelihood $p(y, D|\alpha, \beta, \eta, \sigma^2)$, y is the label of all videos in D. Similar to LDA detailed above, maximizing the joint likelihood is intractable and SLDA maximizes its lower bound. For a given video $w_{1:N}$ and its response variable y it can be described as the following:

$$logp(\mathbf{w}, y|\alpha, \beta, \eta, \sigma^2) \geq \mathscr{D}(q) = \mathbb{E}[logp(\theta, \mathbf{z}, \eta, y, \mathbf{w}] + \mathscr{H}(q)) \quad (6.10)$$

Variational distribution $q(\theta, \mathbf{z}|\gamma, \phi)$ is used to approximate the posterior distribution $p(\theta, \mathbf{z}|\alpha, \beta, \sigma^2, \mathbf{w})$. The expectation $\mathbf{E}$ in Equation 6.10 is derived from the variational distribution $q(\theta, \mathbf{z}|\gamma, \phi)$. More details about inference and parameter estimation in SLDA can be found in Blei *et al.* [62].

Recently, max-margin based techniques have gained popularity and have been incorporated into the MedLDA topic learning process. MedLDA integrates the max-margin prediction models with the hierarchical Bayesian topic models to learn latent topic representations, which are more discriminative and suitable for classification tasks. *i.e.* MedLDA employs maximum-likelihood learning and max-margin learning jointly to discover topics.

MedLDA uses a similar generative process like SLDA to infer the latent variables θ_d and z_{dn}. Unlike SLDA, which draws the label y from the normal distribution, MedLDA learns the label information given the topic assignment $\mathbf{z} = \{z_1, z_2..., z_n\}$ through the latent linear discriminant function below:

$$F(y, \mathbf{z}, \eta) = \eta_y^\top \bar{z}, \tag{6.11}$$

where $\bar{z} = \frac{1}{N}\sum_n z_n$; η_y is a class-specific k-dimensional parameter vector associated with class y. Then the latent topics are discovered through the optimization problem, which combines both max-margin learning and maximum-likelihood estimation. More details on optimization can be found in [120].

6.4.4 css-LDA Approach

Unlike other supervised topic models, css-LDA is a mixture of LDA models that learns a separate topic simplex for each class separately. Topic discovery is done under class supervision and enables it to capture more complex intra class structure and a separate set of topics for each class, which increases the intra class discriminatory power in the topic based representation framework. Other topic model variants use a common topic simplex for the entire dataset and fail to capture the inter and intra class variations. The graphical model representation is shown in the Figure 6.3. The following generative process is similar to LDA, but instead it learns separate topics for each class:

1. Choose $\theta \sim Dir(\alpha)$.
2. Choose a class label $y \sim P_Y(y; \eta)$, $y_i \in Y = \{1, 2, ..., C\}$
3. For each of the words w_{dn}:
   (a) Choose a topic $z_{dn} \sim Multinomial(\theta)$, $z_{dn} \in T = \{1, 2, ..., K\}$
   (b) choose a word w_{dn} from $p(w_{dn}|z_{dn}, \beta)$, a probability conditioned on the topic z_n

where $P_Y()$ is a categorical distribution over the class labels y with the parameter η and other parameters the same as LDA. The main difference in css-LDA is that the word topic distribution is defined by the class specific topics as opposed to a common topic-simplex for all classes. Similar to standard LDA, posterior inference is intractable and approximated variational EM is used to learn the parameters η, α and $\beta_{1:K}^{1:C}$.

6.5 Experimental setup

This section presents four different video representations that have been used in our experiments to evaluate the effectiveness of supervised topic models in local feature-based activity recognition.

Baseline 1: Building histograms around the k-means cluster centres is a popular method of representation and is being widely used in low level activity recognition systems. This method provides a benchmark for evaluating new feature detectors, descriptors, representations and classification algorithms [20, 45, 48, 94, 96]. This framework comprises video feature extraction, vector quantization with K-means, histogram of video feature representation followed by SVM classification.

In K-means clustering, the number of clusters are set to K and each feature is assigned to the nearest cluster centroid based on their Euclidean distance. This hard vector assignment allows each feature vector to be associated with a single cluster. The final feature vector to represent a video is the histogram of cluster assignments. The classification is done with a non-linear, multi-class SVM with a linear kernel. This method is referred as k-means+SVM in experiments.

Baseline 2: In this method the unsupervised LDA is used to reduce the dimensionality of the feature vector. The parameters of an LDA model are estimated

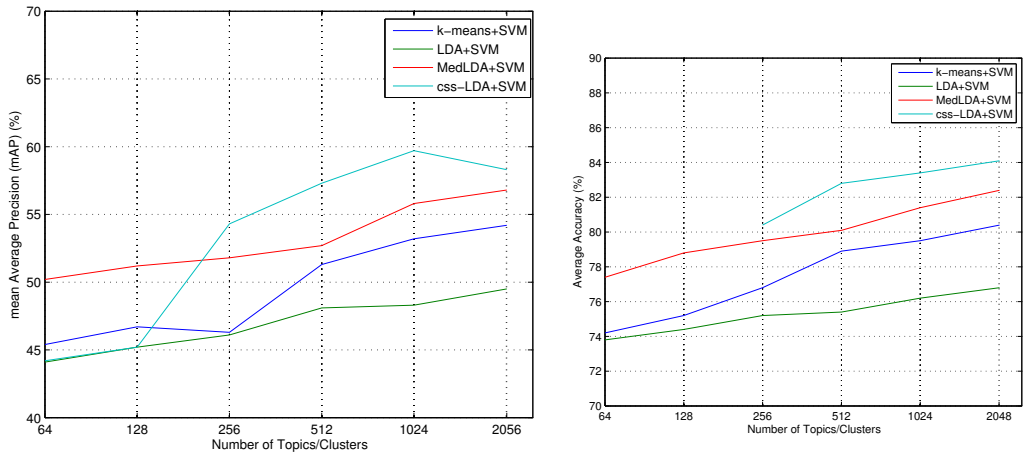


Figure 6.4: Figure (**left**) shows the mean Average Precision (mAP) of **Hollywood2** under 4 different experimental settings with varying number of topics, Figure (**right**) shows the average accuracy of **UCF50** dataset with different number of topics under 4 different experimental settings

using all the training videos without the class label information. The LDA model is learned with different set of topics (K) and the document specific topic distribution (θ_d) is used as a feature vector of the video to learn a multi-class SVM classifier. Topic weights of the testing samples are used to classify the testing samples and this method is referred to in experiments as LDA+SVM.

MedLDA Representation: In this representation, all training samples are used to infer model parameters and the inferred topic proportion vector (θ_d) is used as the video representation to train and test the multi-class SVM classifier. This method is referred to as MedLDA+SVM in the experiments.

css-LDA Representation: In css-LDA, the latent topical distribution is used as a representation for each video. Each action class has a separate topic simplexes and the concatenated topics are used as the final feature vector ($C \times K$ elements) to train and test the multi-class SVM classifier, and this method is referred to as css-LDA+SVM.

6.6 Experimental Results

This section presents experimental results on two popular challenging datasets: Hollywood2 [60] and UCF50 [80]. These two datasets have been chosen because they were collected from different sources with occlusion, viewpoint changes, background clutter, moving background and illumination changes and include a wide range of activity classes.

6.6.1 Hollywood2 Dataset

The Hollywood2 dataset is presented in section 2.5.3. Average precision (AP) for each action classes is calculated and mean AP (mAP) over all classes is reported as a performance measure.

The experimental results on Hollywood2 dataset are shown in Figure 6.4. From the experimental results it can be clearly observed that MedLDA and css-LDA based topic representation outperforms the popular, bag-of-features based representation. Also, it can be noted that the unsupervised LDA performs poorly across all topic structures. The experimental results demonstrates that peak mAP of 59.7% is achieved by the css-LDA, which is 5.5% improvement over baseline Bag-of-words representation. MedLDA also improves the baseline performance by 2.6%. Though the css-LDA representation outperforms other methods, it poorly performs with a lower number of topics compared to MedLDA. It can be explained by the fact that css-LDA builds separate topic simplexes around each class, and it requires a large amount of topics to capture intra and inter class variations, while the MedLDA builds a single topic structure for the entire dataset.

Experimental setup	mAP
k-means+SVM	54.2%
LDA+SVM	49.5%
MedLDA+SVM	56.8%
css-LDA+SVM	59.7%

Table 6.1: Mean Average Precision (mAP) on the **Hollywood2** Dataset using the four different experimental setups

6.6.2 UCF50 Dataset

The UCF50 dataset is presented in Section 2.5.4. The reported results used leave-one-out cross validation and the average accuracy over all classes as the performance measure.

The average accuracy in the UCF50 dataset with different topic representations is shown in Figure 6.4. MedLDA performs well with a small number of topics and css-LDA outperforms all the representations with a large number of topics. Similar to Hollywood2, unsupervised LDA performance is poor across all the topics. As shown in Table 6.2, the best average accuracy is achieved by the css-LDA based representation, which achieves a 3.7% improvement over the baseline.

Experimental setup	Average Accuracy
k-means+SVM	80.4%
LDA+SVM	76.8%
MedLDA+SVM	82.4%
css-LDA+SVM	84.1%

Table 6.2: Average Accuracy on the **UCF50** Dataset using the four different experimental setups

In both action datasets, unsupervised LDA performs poorly because the learned topics fail to capture the underlying class structure. On the other hand, MedLDA explicitly employs the label information during the topic discovery and uses an effective max-margin learning technique in addition to the likelihood-based prob-

abilistic inference. Therefore these topics incorporate more semantic patterns to boost the classification performance. Also it can be noted that MedLDA provides a compact representation of a video without compromising performance, making it ideal when a low dimensional representation is required. On the other hand, even though css-LDA yields best performance, its computational complexity increases with the number of action classes as it builds a separate topic simplex for each activity classes.

Superior performance in both datasets was achieved in css-LDA representation where a separate topic simplex for each activity class demands high dimensional topics to effectively capture both the intra and inter class variations. As the dimension and complexity of css-LDA increases with the number of classes, it is well suited for small numbers of classes with similar spatio-temporal relationships. *e.g.* Sit Down and Stand Up. This provides an interesting direction to explore with hierarchical tree-structures, where compact MedLDA can used in the top part of the tree and css-LDA can be employed down the tree to separate closely co-related activity classes.

6.6.3 KTH Dataset

Experimental setup	Average Accuracy
k-means+SVM	91.8%
LDA+SVM	82.9%
MedLDA+SVM	85.4%
css-LDA+SVM	89.7%

Table 6.3: Average Accuracy on the **KTH** Dataset using the four different experimental setups

The KTH dataset is presented in Section 2.5.1. The average accuracy in the KTH dataset with different topic representations is presented in Table 6.3.

As opposed to dynamic datasets, in KTH dataset the topic representation failed to capture enough variations in discovered topics, hence they yield poor performance compared with the K-means algorithm. This is because this dataset was recorded in static environments and the discovered topics are not well representing the classes to provide enough discriminality for classification. Though class-specific LDA models provide significant boost in performance compared to unsupervised LDA models, they are not outperforming k-means. Topic models employ unsupervised learning approach and they require large amounts of data to discover discriminative topics, therefore static datasets such as KTH and Weizmann are not suitable for topic based approaches because of their limited number of classes and learning samples.

6.7 Chapter summary

In this chapter, several LDA models have been investigated and two supervised topic model-based feature representations are proposed for the local feature-based activity recognition framework. Both MedLDA and css-LDA models provide latent discriminative representations and demonstrate superior performance in two challenging datasets compared to the baseline bag-of-words approach. These supervised topic-based representations are not only compact, but also effectively capture both intra and inter class variations.

From the experiments it was found that MedLDA provides highly efficient, discriminative and sparse topical representation compared to supervised LDA models. The topic proportion vector θ inferred for each video using MedLDA provides robust, significantly improved accuracy of classifying videos compared to the θ resulted from the unsupervised and supervised LDA approaches. MedLDA with variational inference yields efficient topic representation with comparable compu-

tational complexity to unsupervised LDA and significantly lower than supervised approaches. Since MedLDA already employs a max-margin function during topic inference, the second stage max-margin SVM classifier in the MedLDA+SVM framework contributes only a slight improvement in the classification performance. High dimensional topic distributions yield good performance compared to a lower amount of topics because of more semantic information encapsulated inside a large amount of topics. In terms of computational complexity, k-means with SVM classifiers are faster compared to topic models. MedLDA's time complexity is comparable to the LDA+SVM model while css-LDA+SVM is very expensive due to multiple topic discovery based on the number of activity classes present in a dataset.

The introduced topic representations provide an alternative to the "histogram of features" and can be considered as a potential baseline to benchmark new local feature detectors and descriptors.

Chapter 7

Representing activities using class-specific sparse codes

7.1 Introduction

In the popular bag-of-words representation, each feature is assigned to a single codebook element, produces large quantization errors and reduces the overall performance. To address this issue, another efficient feature representation technique based on sparse coding is proposed.

Sparse representation has gained much attention among researchers to successfully analyse a large class of signals such as audio, image, video *etc*. Sparse representation enables us to represent a signal as a linear combination of a small number of basis functions. Unlike other conventional basis functions, sparse representation uses over complete basis (*i.e*. The dimensionality of basis vectors is greater than the dimensionality of the input vector) to represent a signal. This over complete representation facilitates the capture of important information of

a signal with only a small portion of basis vectors. This compact, sparse representation is not only very useful in data compression in telecommunication and data communication networks but also in classification, where sparsity of the signal significantly improves the classification performance compared to the dense counterpart.

Finding an over-complete basis vector creates an under determined system of linear equations $\mathbf{x} = \mathbf{D}a$, where the dictionary matrix $\mathbf{D} \in \mathbb{R}^{n \times m}, (n < m)$ and has an infinite number of solutions. The sparsest solution, $a \in \mathbb{R}^n$ will contain k ($k \ll n$) non-zero elements. Even though this problem is NP hard, several advanced methods have been developed using greedy algorithms and linear programming to solve this problem. Unlike other methods such as wavelets, curvelets, *etc.*, where a pre-defined basis is used, in sparse-based representation the dictionary $\mathbf{D}$ is learnt from the actual signal itself. This allows the flexibility to learn different dictionaries depending on the signal distribution and to capture the inter and intra class structures present in the signal as well as better data fit compared with the off-the-shelf dictionaries. In addition to that, the learned dictionaries are more discriminative and compact compared to pre-defined dictionaries.

In this work, the effectiveness of sparse-representation to create an over complete dictionary to encode video patches in the context of activity recognition is investigated. Recently proposed sparse representation methods have been shown to effectively represent features as a linear combination of an over complete dictionary by minimizing the reconstruction error. In contrast to most of the sparse representation methods, which focus on Sparse-Reconstruction based Classification (SRC), this work focuses on a discriminative classification using an SVM by constructing class-specific sparse codes for motion and appearance separately. Experimental results demonstrate that separate motion and appearance specific sparse coefficients provide most effective and discriminative representation for

each class, compared to shared and class-specific sparse representations.

In recent years, sparse representation has been extensively used in a wide range of computer vision applications, such as image de-noising, image restoration, texture classification, face recognition, object recognition and action recognition [30, 57, 72, 78, 105]. Although sparse representation mainly focuses on learning an over complete dictionary to represent the signal, with only a few elements from the dictionary to minimize the reconstruction error, recently several approaches have been proposed in object recognition that not only minimize the reconstruction error, but also to improve the discriminative power of the sparse coefficients to improve the overall classification performance. Ramirez *et al.* [78] incorporate an incoherence promoting term to make the dictionaries for different classes as independent as possible. Mairal *et al.* [58] proposes to simultaneously learn a classifier by embedding a logistic loss function. Discriminative K-SVD [116] and label consistent K-SVD [36] focused on improving the discriminatory power of the sparse codes with a good representation.

Several sparse representation methods have been extended to solve the action classification problem. Zhu *et al.* [121] introduced sparse representation to classify actions with a shared dictionary with single scale max-pooling and linear SVM classifier. Guha *et al.* [29] explored shared, class-specific and concatenated dictionaries with different reconstruction error-based classification. Sparse Reconstruction-based Classification (SRC) with different features has been explored in [30, 50, 52]. SRC with L_1 and L_2 regularization ($SR{-}L_{12}$) was proposed by Gao *et al.* [25].

Amongst several variations of sparse coding methods proposed for action recognition, the method of this research differs in two ways:

1. Unlike other methods, where a single dictionary for a class is built, in

this approach, separate dictionaries are built for motion and appearance features.

2. In this work, the focus is on discriminative classification and demonstrating better results compared to the SRC method. Also, a comprehensive evaluation has been carried out with a different set of sparse representation techniques, such as SRC, shared-dictionary, class-specific dictionary and proposed appearance and motion specific dictionary.

In the proposed method, first, dense Histogram of Gradient (HOG) features and Motion Boundary Histogram (MBH) [94] features are extracted at different scales. Then, we learn a separate over complete dictionary for appearance and motion vectors is learnt to approximately represent them as a weighted sum of sparse coefficients. These appearance and motion sparse coefficient vectors from several classes are concatenated and pooled to represent each video uniquely. Finally a linear SVM classifier is used for classification.

The rest of this chapter is organized as follows. Section 7.2 provides an overview of the sparse representation framework. A shared dictionary learning approach is presented in Section 7.2.1 and the class-specific dictionary learning approach is presented in Section 7.2.2. Our proposed approach is presented in detail in Section 7.2.3. Details of the experiments carried out on the KTH and UCF dataset is presented in Section 7.3. Finally, Section 7.4 concludes this chapter.

7.2 Dictionary Learning and Sparse Representation

In sparse coding, data samples are modelled linearly as $\mathbf{X} \approx \mathbf{DA}$. Sparse coding is popularly used to represent a signal as a linear combination of an over complete basis where a few elements of the dictionary are used to represent the signal. Sparse representation is defined as follows: for a given signal $\mathbf{x} \in \mathbb{R}^n$ and a dictionary $\mathbf{D} \in \mathbb{R}^{n \times k}$, $(k > n)$, the sparse representation of the signal $\mathbf{x}$ is obtained as the solution to the following optimization problem,

$$a^* = \underset{a}{\operatorname{argmin}} \|a\|_0, \quad s.t. \frac{1}{2}\|\mathbf{Da} - \mathbf{x}\|_2^2 \leq \epsilon, \tag{7.1}$$

where $\|a\|_0$ is the l_0 norm of the coefficient vector, which counts the number of non-zero entries, and $\mathbf{a} \in \mathbb{R}^k$ are the approximation weights *i.e.* minimizing the number of non-zero elements present in the coefficient vector. Minimizing the l_0 norm is an NP-hard problem and greedy algorithms don't guarantee an optimal solution. Under the assumptions on the sparsity of the signal and the structure of the dictionary $\mathbf{D}$ there exists $\lambda > 0$ such that the l_0 pseudo-norm can be replaced with an l_1 norm and the following optimization problem can be solved instead,

$$a^* = \underset{a}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{Da} - \mathbf{x}\|_2^2 + \lambda\|a\|_1, \tag{7.2}$$

where the parameter λ is used to establish balance between the sparsity and reconstruction error. The above optimization problem becomes convex and can be solved easily using modern convex optimization techniques. The Equation 7.2 is known as LASSO. The l_1 norm induces the sparse solution for the code vector

a. Sparse modelling is done via an alternative minimization technique, where first **D** is fixed and obtain the sparse code $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n] \in \mathbb{R}^{k \times n}$, then by fixing **A** and minimizing with respect to **D**. Both of these sub-optimization problems are convex and the process is continued until the local minimum is obtained,

$$(\mathbf{D}^*, \mathbf{A}^*) = \underset{a}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{DA} - \mathbf{X}\|_2^2 + \lambda \sum_{i=1}^{n} \|a_i\|_1, \tag{7.3}$$

The generated dictionary is shared across all the action classes and the corresponding sparse representation for each feature vector is obtained by minimizing the l_1 norm.

7.2.1 Shared dictionary Approach

In this representation a single shared dictionary **D** is learned using all the training samples. This approach is computationally efficient during training and testing phases compared to class-specific approaches, because only a single dictionary is required for the entire dataset regardless of the number of activities present in the dataset. However, on the other hand, it is not as discriminative as class-specific dictionaries. For a given set of features **X** extracted from a dataset, the **A** represents the corresponding sparse coefficients obtained from the dictionary **D**. Finally, the video representation is obtained by calculating the sparse-coefficient histogram over the set of features representing the video. Take the i^{th} video having a set of r features and their corresponding sparse representation $A_i = \{a_k\}_{k=1}^{r}$. Then the sparse coefficient histogram h_i is defined as follows,

$$h_i = \frac{1}{r} \sum_{k=1}^{r} a_k. \tag{7.4}$$

These histograms of coefficient representation of training video samples are used to train the multi-class SVM classifier.

7.2.2 Class-specific dictionary learning

In this framework, for a dataset consisting of C action classes, C dictionaries ($\{D_1, D_2, D_3, \ldots, D_C\}$) are learned, one for each class. Unlike a shared dictionary, the computational complexity of this representation increases with the number of action classes. On the other hand, the dictionary learned for a given class is efficient for representing activities for this class and less efficient for representing activities from different class (*i.e.* the sparse representation obtained via the dictionary corresponding to that class has low reconstruction error and is more sparse compared to the representation obtained via a different dictionary).

Let $\mathbf{X}^j = [\mathbf{x}_1^j, \mathbf{x}_2^j, \ldots, \mathbf{x}_{n_j}^j]$ the n_j features extracted from the j^{th} action class and the dictionary corresponding to the j^{th} action class $\mathbf{D}^j \in \mathbb{R}^{m \times k_j}$ is obtained by solving the following optimization problem,

$$\mathbf{D}_j^* = \arg \min_{(\mathbf{D}^j, \mathbf{A}^j) \geq 0} \frac{1}{2} \|\mathbf{D}^j \mathbf{A}^j - \mathbf{X}^j\|_2^2 + \lambda \sum_{i=1}^{n_j} \sum_{i=1}^{k_j} a_i^j. \tag{7.5}$$

A separate class specific dictionary is learnt for all C action classes. Then all the class-specific dictionaries are combined to form a block-structured dictionary $\mathbf{D} = [\mathbf{D}^1, \mathbf{D}^2, \mathbf{D}^3, \ldots, \mathbf{D}^C] \in \mathbb{R}^{m \times k}$, where $k = \sum_{j=1}^{C} k_j$. Then, each feature vector is represented as a linear combination of the block-structured dictionary,

$$\mathbf{A}^* = \underset{\mathbf{A}>0}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{D}\mathbf{A} - \mathbf{X}\|_2^2 + \lambda \sum_{i=1}^{n} \sum_{j=1}^{k_j} a_i^j, \tag{7.6}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots, \mathbf{a}_n] \in \mathbb{R}^{k \times n}$, $\mathbf{a}_i = [a_i^1, a_i^2, \ldots, a_i^{k_C}]^\intercal \in \mathbb{R}^k$. This structured dictionary approach allows each feature to share different class-specific dictionaries and effectively capture the statistical characteristics compared to a single, shared dictionary approach.

7.2.3 Appearance & Motion specific Dictionary Learning

This proposed approach is similar to the above mentioned class-specific sparse codes but rather than learning a single sparse dictionary for each class this research goes granular to learn separate dictionary for appearance and motion features as they capture different statistics of the video to further discriminate between different actions.

In the proposed method, the appearance vector $\mathbf{X_A} = [x^1, ..., x^m] \in \mathbb{R}^{n_A \times m_A}$, where n_A is the dimension of the appearance vector extracted from a given class and m_A is the number of the appearance vectors. $\mathbf{X_A}$ is sparsely represented by minimizing the following equation:

$$\min_{\mathbf{D_A}, \mathbf{C_A}} \|\mathbf{X_A} - \mathbf{D_A}\mathbf{C_A}\|_2^2 + \lambda |\mathbf{C_A}|_1 \tag{7.7}$$

where, class-specific appearance dictionary $\mathbf{D_A} \in \mathbb{R}^{n_A \times d_A}$ with the size of the dictionary d_A and corresponding sparse coefficients $\mathbf{C_A} \in \mathbb{R}^{d_A \times m_A}$. Appearance vector x^i can be approximated as $x^i \approx \mathbf{D_A} c_A^i$. *i.e.* c_A^i is the sparse coefficient vector corresponding to the appearance feature vector x^i.

Similar to the appearance encoding, motion vector $\mathbf{Y_M} = [y^1, ..., y^m] \in \mathbb{R}^{n_M \times m_M}$, where n_M is the dimension of the motion vector extracted from a given class and m_M is the number of the motion vectors and is sparsely represented by minimizing

the following equation:

$$\min_{\mathbf{D_M},\mathbf{C_M}} \|\mathbf{X_M} - \mathbf{D_M}\mathbf{C_M}\|_2^2 + \lambda|\mathbf{C_M}|_1 \tag{7.8}$$

where class-specific motion dictionary $\mathbf{D_M} \in \mathbb{R}^{n_M \times d_M}$ with the size of the dictionary d_M and the corresponding sparse coefficients $\mathbf{C_M} \in \mathbb{R}^{d_M \times m_M}$. Motion vector y^i can be approximated as $y^i \approx \mathbf{D_M} c_M^i$. *i.e.* c_M^i is the sparse coefficient vector corresponding to the motion feature vector y^i.

The class-specific dictionary is given by the concatenation of motion (c_M^i) and appearance (c_A^i) sparse coefficient vectors. Then the block-structured dictionary is constructed by combining all the class-specific appearance and motion dictionaries (*i.e.* similar to class-specific dictionary in Section 7.2.2). Then the final representation of an interest point (I^i) is given by the linear combination of block-structured dictionaries.

7.3 Experiments and Results

A comprehensive set of experiments have been carried out with different sparse-representation approaches to validate the proposed method. Two popular action recognition datasets with varying complexity: KTH [86] dataset is used to demonstrate the effectiveness of sparse representation in simple environmental settings and UCF sports [80] dataset is used to demonstrate the effectiveness in complex and cluttered environments. The following experimental set-up is used to evaluate different sparse representations.

Sparse Representation-based Classification (SRC): The SRC method [72,

107] assigns each feature to the action class based on the reconstruction error: $\mathscr{R}(x, \mathbf{D}) = \|x - \mathbf{D}a\|_2^2$, where $x \in \mathbb{R}^n$ is the feature vector, $\mathbf{D}$ is the dictionary and the sparse code vector, $a \in \mathbb{R}^k$, is calculated from Equation 7.2. For a K class classification problem, each class i has a dictionary $\mathbf{D}^i$ and a code a^i is calculated for each dictionary. Finally the feature vector x is assigned to the class i^* which minimizes the reconstruction error $\mathscr{R}$:

$$i^* = \underset{i}{\operatorname{argmin}} \mathscr{R}(x, D^i) \tag{7.9}$$

Shared dictionary with an SVM classifier: A single shared dictionary $\mathbf{D}$ is learned to sparsely encode each feature vector, followed by spatio-temporal pooling and a linear SVM classifier is applied for classification.

Class-specific dictionary with SVM classifier: We learnt C separate dictionaries $\{\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_C\}$ for each class, followed by spatio-temporal pooling and linear SVM classification.

In feature extraction, we densely sample each video and extract Histogram Oriented Gradients (HOG) and Motion Boundary Histogram (MBH) features to represent each video. For each cell, an 8-bin HOG histogram is calculated and normalised into a HOG descriptor. The robust optical flow based MBH [94] descriptor is used to capture the motion information present in the spatio-temporal volume.

The parameter λ in the optimization function 7.2 controls the sparsity of the the sparse coefficient vector while minimizing the reconstruction error. The parameter λ is set to 10% in all experimental settings, which yields better results. Randomly selected HOG and MBH features are used from each class to generate

Experimental setup	Average accuracy (%)
SRC	86%
Shared Dictionary + SVM	92%
Class Dictionary + SVM	94.5%
Proposed method	96.8%

Table 7.1: Average Accuracy on the **KTH** Dataset using the four different experimental setups

the appearance and motion specific dictionaries. Once the shared, class-specific and appearance and motion specific dictionaries are learnt, each feature vector is mapped to the sparse coefficient vector via l_1 minimization.

7.3.1 KTH Dataset

The KTH dataset is presented in Section 2.5.1. The same experimental setting proposed by Schuldt *et al.* [86] is used. Table 7.1 shows the average accuracy obtained with four different sparse representations. The proposed sparse representation outperforms the class-specific dictionary by 2.3%. Confusion matrices for class-specific representation and the proposed method is shown in Table 7.1. The proposed representation not only performs well across all the classes but also reduces the confusion among closely related classes by increasing the discriminatory power.

	Running	Boxing	Walking	Jogging	Waiving	Clapping
Running	0.91	0.00	0.02	0.07	0.00	0.00
Boxing	0.00	0.96	0.00	0.00	0.00	0.04
Walking	0.00	0.00	0.97	0.03	0.00	0.00
Jogging	0.03	0.00	0.04	0.93	0.00	0.00
Waiving	0.00	0.00	0.00	0.00	0.95	0.05
Clapping	0.00	0.03	0.00	0.00	0.02	0.95

(a) Class-specific sparse dictionary

	Running	Boxing	Walking	Jogging	Waiving	Clapping
Running	0.92	0.00	0.00	0.08	0.00	0.00
Boxing	0.00	0.98	0.00	0.00	0.00	0.02
Walking	0.00	0.00	1.00	0.00	0.00	0.00
Jogging	0.02	0.00	0.05	0.93	0.00	0.00
Waiving	0.00	0.00	0.00	0.00	1.00	0.00
Clapping	0.00	0.00	0.00	0.00	0.02	0.98

(b) Appearance & Motion specific dictionary

Figure 7.1: Confusion matrices for the **KTH** dataset with different sparse representations. (a) Class-specific sparse dictionary and (b) Appearance & Motion specific sparse dictionary.

Experimental setup	Average accuracy (%)
SRC	84%
Shared Dictionary + SVM	87%
Class Dictionary + SVM	89 %
Proposed method	92.3%

Table 7.2: Average Accuracy on the **UCF-Sports** Dataset using the four different experimental setups

7.3.2 UCF Sports Dataset

The UCF-Sports dataset is presented in Section 2.5.4. The Leave-one-out cross validation and average accuracy is reported in Table 7.2.

The overall classification rate of 92.3% is obtained, which is 3.3% higher compared to the class-specific dictionary. Confusion matrices for class-specific sparse codes and the proposed method are shown in Figure 7.2.

	Driving	Golf Swinging	Kicking	Lifting	Horse riding	Running	Skating	Swinging	Walking
Driving	0.97	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00
Golf Swinging	0.00	0.93	0.03	0.00	0.00	0.00	0.00	0.04	0.00
Kicking	0.03	0.00	0.90	0.00	0.00	0.00	0.00	0.00	0.07
Lifting	0.00	0.00	0.00	0.90	0.00	0.00	0.00	0.00	0.10
Horse riding	0.00	0.00	0.00	0.00	0.82	0.00	0.06	0.00	0.12
Running	0.00	0.00	0.00	0.00	0.00	0.82	0.05	0.00	0.13
Skating	0.00	0.00	0.12	0.00	0.00	0.00	0.88	0.00	0.00
Swinging	0.00	0.05	0.08	0.00	0.00	0.00	0.00	0.87	0.00
Walking	0.00	0.05	0.00	0.00	0.00	0.10	0.00	0.00	0.85

(a) Class-specific sparse dictionary

	Driving	Golf Swinging	Kicking	Lifting	Horse riding	Running	Skating	Swinging	Walking
Driving	0.99	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Golf Swinging	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.04
Kicking	0.00	0.00	0.94	0.00	0.00	0.00	0.00	0.04	0.02
Lifting	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.04
Horse riding	0.00	0.00	0.00	0.00	0.85	0.00	0.07	0.00	0.08
Running	0.00	0.00	0.02	0.00	0.00	0.88	0.00	0.00	0.10
Skating	0.00	0.00	0.03	0.00	0.00	0.00	0.94	0.00	0.03
Swinging	0.00	0.05	0.02	0.00	0.00	0.00	0.00	0.93	0.00
Walking	0.00	0.05	0.00	0.00	0.00	0.05	0.00	0.00	0.90

(b) Appearance & Motion specific dictionary

Figure 7.2: Confusion matrices for the **UCF** dataset with different sparse representations. (a) using Class-specific sparse dictionary and (b) Appearance & Motion specific sparse dictionary.

Experimental results in two datasets demonstrate that class-specific dictionaries provide a better, sparse and discriminative representation for their own class compared to a shared dictionary approach. Further, the shared nature of the class-specific appearance and motion dictionaries allow other classes to effectively capture common spatio-temporal elements present in their action sequences. For example, some atoms in the motion dictionary built for the running class can be used to represent temporal elements of the walking or jogging class and atoms in the appearance dictionary built for the boxing class can be used to represent some spatial elements in the boxing class. This rich dictionary structure allows the focus on and capturing of minor spatio-temporal elements, which are important to differentiate between two closely related classes.

Improvement in performance is obtained without adding any additional term in the optimization function. Availability of parallel processing hardware will allow the building of appearance and motion specific dictionaries simultaneously. Therefore the computational requirement is almost the same as building a class-specific dictionary with combined motion and appearance features.

7.4 Summary

In this chapter, several sparse representation approaches in local feature based activity recognition have been investigated and an efficient way of constructing sparse dictionary for representing activities for discriminative action classification has been presented. A comprehensive set of experiments have been carried out and the experimental results on two popular datasets demonstrated; building separate appearance and motion specific dictionaries for each class significantly improves the classification performance compared to a shared dictionary and class-specific dictionary. It is also an interesting observation that as the research

went further granular in designing the over-complete dictionary (*i.e.* from shared to class-specific to appearance & motion specific) the discriminative ability of the feature representation increased. In addition to that, this proposed representation adds more discriminative power to the video representation and can be extended to different video based applications.

Chapter 8

Binary-Tree SVM for Representing Activities

8.1 Introduction

This chapter presents an effective classification structure to improve the discriminative activity classification based on Support Vector Machines (SVM). SVMs are popularly used because of their simplicity and efficiency; however the common multi-class SVM approaches applied suffer from limitations, including having easily confused classes and being computationally inefficient.

As mentioned in earlier chapters, efficient and effective video representation and classification plays an important role in recognizing human activities from video sequences. This chapter addresses the classification problem by proposing a binary tree SVM to address the shortcomings of multi-class SVMs in activity recognition. This chapter also presents a new method of constructing a binary tree using Gaussian Mixture Models (GMM), where activities are repeatedly allocated

to sub-nodes until every newly created node contains only one activity. Then, for each internal node a separate SVM is learned to classify activities, which significantly reduces the training time and increases the speed of testing compared to popular the 'one-against-the-rest' multi-class SVM classifier. Experiments carried out on the challenging and complex Hollywood2 dataset demonstrate comparable performance over the baseline bag-of-features method.

Local feature-based methods incorporate the Bag-of-visual-words (BoV) representation to consolidate the local features for the purpose of action classification. In local feature-based action recognition, classification is done with SVM classifiers, often in combination with a χ^2 kernel. Although support vector machines were originally developed for binary classification problems, two main variations of multi class SVM classifiers [33] are popularly used in the context of action recognition: 'one-against-rest' and 'one-against-one'. The 'one-against-rest' method is a popularly used multi-class classifier for action recognition and requires N classifiers for a N class classification problem. In the training phase, a particular class is considered as positive and the remaining $N-1$ classes are treated as negative. Since all SVMs are trained with all the training samples, this consumes more computational resources and reduces the performance due to a large amount of negative samples. In the testing phase, all N SVMs are required to predict the sample data point. On the other hand, a 'one-against-one' approach requires $N(N-1)/2$ SVM classifiers, each trained with a pair of classes. While this improves performance compared to a 'one-against-rest' approach, it still requires $N(N-1)/2$ binary decisions to predict the test sample based on majority voting.

This chapter addresses the above mentioned problems in multi-class SVMs by using a binary-tree SVM [16]. In the first stage, to convert the problem into a binary decision tree, Gaussian Mixture Model (GMM) clustering is used. At

the beginning, all the training samples are assigned to the root node of the tree and a GMM is used to separate the training samples into two clusters, and the activities belonging to each cluster are assigned to the left and right sub-nodes respectively. The GMM is continuously applied at sub-nodes to further split the activities into pairs, until every newly created node contains only one class. In the second stage, SVM training, each internal node in trained with a SVM to make a binary decision. In the training phase, it requires only $N-1$ SVMs to be trained for an N class problem; and the amount of time required for training also reduces as the tree is traversed downwards as the number of classes (and amount of data) at each node is reduced. When performing classification, the proposed approach requires only $log_2 N$ SVMs to predict the sample due to the binary nature of the decision tree.

The reminder of the chapter is organized as follows: Section 8.2 describes the feature extraction and representation. Section 8.3 provides details of the proposed classification method. Experimental results for the popular Hollywood2 dataset is presented in Section 8.4. Finally, Section 8.5 concludes the paper.

8.2 Video representation

In this section, the feature extraction and feature encoding scheme used in experiments is described.

8.2.1 Feature extraction

The video is encoded using low level, local features incorporating static appearance and motion information. The Histogram Oriented Gradients (HOG) descriptor is used to encode the appearance information and the Motion Boundary Histogram (MBH) is used to capture motion information. Instead of using a dense space-time cuboid, video is sampled using dense trajectories [94], with default parameters. Trajectories of length 15 frames are extracted on a dense grid with 5-pixel spacing.

For appearance, the HOG descriptor is calculated along the trajectory and the cuboid region is subdivided into a $2 \times 2 \times 3$ grid of cells. For each cell, an 8-bin HOG histogram is calculated and normalised into a HOG descriptor. Motion information is captured using the MBH [94] along the trajectories.

8.2.2 Feature encoding

Once the two local features are extracted, the popular, standard bag-of-visual-words (BoV) approach for representation is used to make fair comparison with other methods. This approach requires the construction of a visual vocabulary and the K-means algorithm was used, with the number of clusters set to $k = 4000$, to generate the required vocabulary, which has been a popular choice amongst researchers. Then, each video feature is assigned to the closest cluster based on the Euclidean distance, and is represented by a histogram of visual word occurrences over a video sub-volume defined by a dense trajectory.

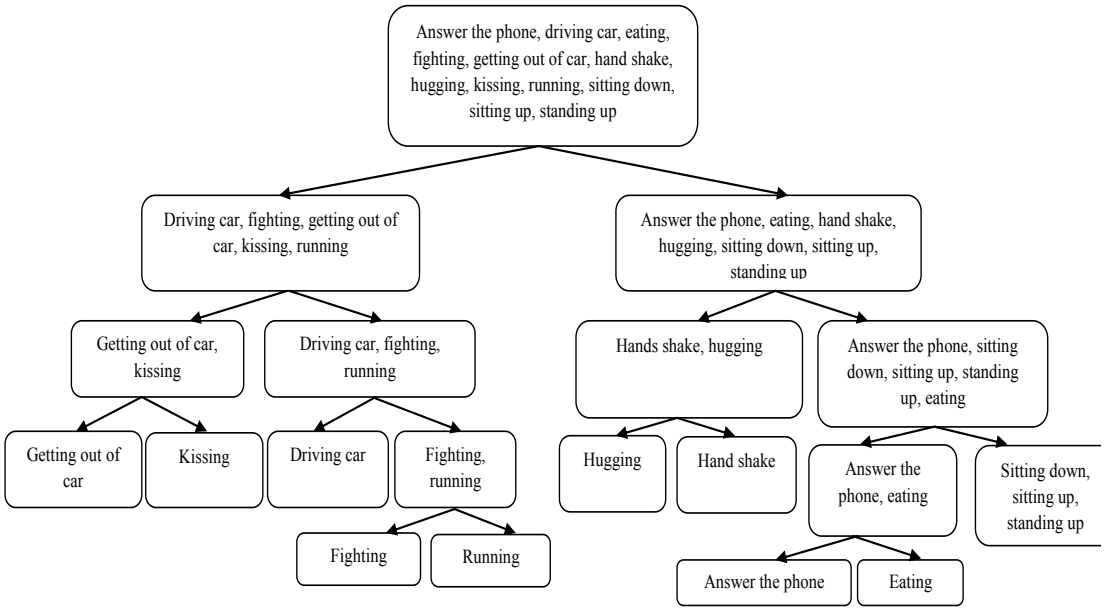


Figure 8.1: Binary tree structure for support vector machine classification in the Hollywood2 dataset.

8.3 Binary Tree Construction with GMM

The organization of the binary decision tree is vital, as errors have the potential to propagate down the tree. A GMM clustering algorithm is used to convert the multi-class problem into a binary decision tree. In this work, the overlapping of classes is avoided to make the classification framework as simple as possible.

GMMs are considered to be a soft clustering approach, which uses the EM algorithm to assign features to mixture components, based on their posterior probabilities, $p(k|x)$. But unlike the k-means, which performs a hard assignment of features to a cluster, GMM considers the shape of the distribution as well. A GMM is a generative model to describe the distribution of feature space as follows:

$$p(x;\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x;\mu_k,\Sigma_k), \tag{8.1}$$

where K is the number of mixtures, model parameters are $\theta = \{\pi_1,\mu_1,\Sigma_1,\ldots,\pi_k,\mu_k,\Sigma_k\}$ and $\mathcal{N}(x;\mu_k,\Sigma_k)$ is a D-dimensional Gaussian distribution. Given a set of features $X = \{x_1,\ldots,x_M\}$, the EM algorithm is used to learn the optimal parameters through maximum likelihood, $lnp(X;\theta) = \Sigma_m lnp(x_m;\theta)$.

Initially, all activity samples are allocated to the root node and the GMM algorithm is applied to split the activities into two clusters, where majority voting is used to assign the classes to appropriate clusters. After step one, two sub-nodes denoted as N_L and N_R are created, each containing a portion of action classes from its parent. The process is illustrated in Figure 8.1. In Figure 8.1, at the first level five activities are allocated to the left node (N_L) and the remaining seven activities are allocated to the right node (N_R). This clustering procedure continues recursively at sub-nodes, N_L and N_R, until every newly created node contains only one class.

8.3.1 SVM classification

After the binary tree is constructed, a separate SVM is trained for each node, except leaf nodes. For an N class problem it requires $N-1$ SVMs. Also, as we go down the tree, its computational complexity reduces and discriminatory power increases as a result of each node comparing fewer and fewer classes. In the testing stage, due to the binary nature of the tree, only half of the SVMs are employed in the decision making instead of all SVMs as in other multi-class SVM methods.

Level		Error (%)	
		K-means	GMM
Root		1.7	1.2
Level 1	L_{11}	6.2	5.4
	L_{12}	11.9	4.8
Level 2	L_{21}	4.2	3.4
	L_{22}	7.4	7.8
	L_{23}	8.9	6.4
	L_{24}	22.6	4.1
Level 3	L_{31}	15.2	10.4
	L_{32}	7.4	8.2
	L_{33}	26.3	5.2
Level 4	L_{41}	7.9	5.3

Table 8.1: The clustering results for constructing the Binary Tree (see Figure 8.1). The error represents the percentage of misclassified feature vectors in each node. The root node consists of all activities, the L_{11} node consists of {Driving car, fighting, getting out of car, kissing, running}, L_{12} consists of {Answer the phone, eating, hand shake, hugging, sitting down, sitting up, standing up} and so on.

8.4 Experimental results

The Hollywood2 dataset is used to validate our proposed classification method. This dataset has been chosen because of its complexity and the activities are related closely in the spatial and temporal domains. For the clustering, k-means (hard clustering algorithm) and Gaussian Mixture Model (GMM) are used. Table 8.1 shows the clustering results. We find that GMM clustering demonstrates best clustering with minimal overlapping of classes and organizes the tree such a way that the classes are easy to differentiate first, and complexity increases down the tree.

Table 8.2 compares the results of the proposed method against the state-of-the-art method [94], where they combined HOG, HOF and MBH features using the multi-channel approach and use 'one-against-rest' multi-class classification. The proposed Binary-Tree SVM method not only achieves comparable performance,

Action class	Wang *et al.* [94]	Our method
AnswerPhone	32.6%	30.5%
DriveCar	88.0%	87.4%
FightPerson	81.4%	80.1%
GetOutCar	52.7%	51.3%
Kiss	65.8%	66.4%
Run	82.1%	83.2%
Eat	65.2%	67.2%
SitDown	62.5%	63.8%
SitUp	20.0%	21.3%
StandUp	65.2%	67.2%
HandShake	29.6%	27.6%
HugPerson	54.2%	52.3%
mAP	58.3%	58.2%

Table 8.2: Average Precision(AP) per action class for the **Hollywood2** dataset compared against [94]

but also significantly reduces the computational complexity in testing to $log_2 N$, as opposed to N in other methods.

In addition, it can be noted that activities are separated into spatial and temporal events along the tree and more complex activities, such as sitting down and sitting up, are pushed down the tree (see Figure 8.1). This enables the SVM to easily classify activities which are similar in nature spatially or temporally, compared to the 'one-against-other' approach where one activity is classified against all other activities. Also, this tree structure potentially allows different sets of features to be used at each internal node to further improve performance.

Binary-tree based approach is well suited for complex datasets with large number of classes and training samples as the number of training samples reduces when traverse down the tree. Static datasets such as KTH dataset is limited to 6 different classes, each contains 100 different samples, this limits the number of samples available to train/learn the SVM model, hence the learned SVM failed

to differentiate different classes.

8.5 Conclusion

This chapter presented a new efficient classification approach for Bag-of-feature based activity recognition. In the proposed binary tree SVM approach, first GMM clustering is used to construct a binary decision tree; after which a separate SVM is trained for each node of the tree. This approach is not only efficient, but also useful in classifying a large amount of activities, which are otherwise difficult to distinguish spatially and temporally. Also this allows different sets of features to be used for different activities within a given dataset, which is particularly useful when the dataset contains a large amount of activity classes.

As digital information is exploding day by day, this presents a huge amount of data for researchers to carry out experiments. The classification method presented in this chapter is effective for larger datasets containing a larger amount of activity classes, because it significantly reduces the training and testing time as the number of activities decreases along the tree. Also, this method can be further explored to design optimal features to use at each node, based on the activities observed at that node. In this way, further improvement in the classification accuracy can be achieved.

Chapter 9

Conclusions and Future Directions

9.1 Introduction

This chapter presents a summary of the work presented in this thesis and the conclusions drawn from it. Even though several activity recognition frameworks exist in literature, local feature-based systems are very popular due to their simplicity and their superior performance. In this thesis, local feature-based action recognition has been extensively studied and several advances have been proposed. The summary follows the three main research themes and areas of contribution identified in Chapter 1: (1) providing a comprehensive evaluation of the local feature-based action recognition system (2) improving the system performance by developing new, efficient spatio-temporal features (3) developing new feature representation and classification techniques to improve the overall recognition performance. Possible future research directions that could be pursued as a natural extensions of this work are also pointed out.

9.2 Conclusions

Below is the Summary of the contributions in this thesis:

1. **Chapter 3** provides an comprehensive investigation of several popular local feature descriptors with challenging datasets. In recent times, several features, representations and classification methods have been proposed, but these methods were evaluated with different experimental settings and are difficult to compare with other methods. In this work, this problem has been addressed with a comprehensive evaluation of the popular local feature descriptors under a common framework. In addition, several elements in the pipeline such as impact of code book sizes, encoding methods and kernel matrices were also extensively studied and several advanced techniques have been proposed to improve performance. In this chapter it was found that different stages in the pipeline play a significant role and the performance of the raw features can be increased by 3-7 % by properly choosing appropriate techniques in the pipeline.

2. **Chapter 4** proposes a novel video detector/descriptor based on the BRISK descriptor. In this proposed approach, the binary feature detector BRISK is applied to detect the key points on a frame-by-frame basis followed by a sparse optical flow algorithm to choose potential candidate points. Finally, appearance information of these points are encoded with BRISK descriptor and motion information is encoded with MBH descriptor. Experimental results demonstrate that this final descriptor is not only computationally efficient but also provide comparable performance to other state-of-the art descriptors. Even though this descriptor has been evaluated on activity datasets, this can be extended to other video-based applications as well.

3. **Chapter 5** presents a novel feature representation method based on Multi-

ple Instance learning (MIL) for activity representation. In this work, three MIL techniques such as 'mi-SVM + k-means', 'M^3IC' and MMDL are introduced to create codebooks and to encode features for discriminative activity recognition. These representations are shown to be more discriminative compared to bag-of-words representation; from the experiments it was also found that the MMDL approach produces more discriminative codebooks compared to mi-SVM + k-means and M^3IC approach, at the expense of computational complexity.

- **mi-SVM + k-means Approach:** In this approach, features corresponding to a particular activity class is treated as positive and all the features are assigned to a set of positive bags, and the rest of the classes are treated as negative and their features are assigned to negative bags. Then SVM is learned on positive and negative bags to identify the positive features in the positive bags followed by the K-means algorithm to cluster the positive instances. In this approach, codebooks are learned per class basic as opposed to a single class learned using bag-of-words approach and are shown to be more effective.
- **M^{3}IC Approach:** In this approach, the K-means clustering algorithm has been replaced and a single dictionary is built using MIL techniques. This approach produces an efficient, compact representation as the codebook size doesn't grow with the number of activity classes.
- **MMDL Approach:** In this approach, class-specific dictionaries are learned, but instead of separately performing multiple instance learning and mixture modelling as two steps in 'mi-SVM + kmeans' both steps are carried out simultaneously. This approach demonstrates best performance amongst the three proposed MIL approaches in two popular activity recognition datasets.

4. **Chapter 6** presents two novel supervised LDA variants to convert raw appearance and motion features more suitable for classification. In contrast to unsupervised LDA, where topics are discovered without the knowledge of the label information, supervised LDA variants discover more informative and discriminative topics by incorporating label information. In the proposed approach the histogram of features are replaced with the topic proportion vector of a particular video. The following two proposed representations are found to be effective and significantly improve the recognition accuracy.

   - **MedLDA Approach:** In this approach, MedLDA learns discriminative topics by employing a max-margin technique within the probabilistic framework. MedLDA provides highly efficient, discriminative and sparse topical representation compared to other supervised LDA models. The topic proportion vector inferred for each video using MedLDA provides robust, significantly improved accuracy of classifying videos compared to unsupervised and supervised LDA approaches. MedLDA with variational inference yields efficient topic representation with comparable computational complexity to unsupervised LDA and significantly lower than supervised approaches.
   - **css-LDA Approach:** In this approach the supervision is introduced at the feature level and enables class specific topic simplexes to capture much richer intra-class information and provides a single set of topics for the entire data set. As the dimension and complexity of css-LDA increases with the number of classes, it is well suited for small numbers of classes with similar spatio-temporal relationships. In the meantime, MedLDAs time complexity is comparable to the LDA+SVM model while css-LDA+SVM is very expensive due to multiple topic discovery based on the number of activity classes present in a dataset.

5. **Chapter 7** presents a novel sparse-representation technique representing activities. Class-specific appearance and motion dictionaries are proposed to encode raw features into a sparse coefficient vector suitable for discriminative SVM classification. This proposed representation is shown to be effective by evaluating against other sparse representation techniques such as shared and class-specific dictionary approaches. In addition, the suitability of a shared and class-specific dictionary in discriminative classification is also extensively investigated.

6. **Chapter 8** presents a binary-tree SVM, which is highly scalable to wild datasets containing complex activities. This approach also reduces the training and testing time by building a binary-tree while providing the flexibility to design customized features at every leaf node separately.

9.3 Future work

In this section, several different directions for potential future work that can be extended from this thesis are presented and potential directions suggested that can be pursued in video-based human activity recognition. In this research program, a local feature-based activity recognition system has been extensively studied and several features and advanced machine learning techniques have been proposed. We propose further investigations in the following areas to improve the performance further.

- **Feature fusion using multiple camera inputs:** In this thesis we investigated the activities that contain single view information at a given point in time. This information is not enough to capture spatio-temporal relationships. On the other hand, a multiple view of video footage allows us to

capture more granular details of the activity. In addition, the availability of cheap 3D recording devices such as Kinect also allows us to capture 3D information and further research can be done with 3D key point detectors and descriptors in addition to 2D detector/descriptors. In this thesis, the research has not been undertaken because of lack of 3D datasets and we believe the 3D keypoint detector/descriptor will be the potential future direction to further explore local features. Also, multiple camera networks also provide more information to fuse features to obtain rich and discriminative representation.

- **Explore temporal sequences:** In this research, most of the focus has been given to improve representation by incorporating spatio-temporal relationships. This thesis has presented three different representation techniques to effectively capture the spatio-temporal relationships. However the temporal order of the features is not explicitly explored due to lack of information present in the feature space. Future research can be carried out such a way to find methods to incorporate temporal order of the spatio-temporal features, which is an integral part of an action sequence.

- **Big Data Analysis:** The current performance obtained in activity recognition does not satisfy the demand from real world applications. This is due to two major reasons, such as lack of performance in real world settings and lack of a fully annotated database containing a significant amount of activities.

  Even though various datasets such as Hollywood2 and UCF50, can be seen as real world activities, they have a limited amount of training samples and fail to capture a wide range of activities. The advances in virtual reality platforms allow simulation of a large amount of activities under different conditions and we believe that in the future, computer graphics and computer vision techniques can be combined to generate large amount

of actions in real world situations. These datasets with fully developed evaluation protocols, will enable effective algorithms to be developed to recognize human activities, with a level of accuracy required for real world deployment.

- **Combine local features with high level representations:** Even though local features are popular among researchers, they are reductive; and rich visual temporal-spatial structures (such as those associated with golf-swinging) can be hardly characterized by one single class label and would be better represented by considering multiple high-level semantic concepts such as action attributes and part-based models describing the action, to enable the construction of more descriptive models for human activity. This research study leads to the belief that the proposed advanced representations in this thesis can be explored further with high level representations to improve recognition performance.

Bibliography

[1] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 1, pp. 44–58, 2006.

[2] J. Aggarwal and Q. Cai, "Human motion analysis: A review," in *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pp. 90–102, IEEE, 1997.

[3] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, pp. 16:1–16:43, Apr. 2011.

[4] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510–517, IEEE, 2012.

[5] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[6] S. Andrews, I. Tsochantaridis, and T. Hofmann, "Support vector machines for multiple-instance learning," in *Advances in neural information processing systems*, pp. 561–568, 2002.

[7] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[8] P. Beaudet, "Rotationally invariant image operators," in *Proceedings of the International Joint Conference on Pattern Recognition*, pp. 579–583, 1978.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, pp. 1395–1402, IEEE, 2005.

[10] D. M. Blei and M. I. Jordan, "Modeling annotated data," in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 127–134, 2003.

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

[12] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 3, pp. 257–267, 2001.

[13] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: binary robust independent elementary features," in *Computer Vision–ECCV 2010*, pp. 778–792, Springer, 2010.

[15] Y. Chen, J. Bi, and J. Z. Wang, "Miles: Multiple-instance learning via embedded instance selection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 1931–1947, 2006.

[16] S. Cheong, S. H. Oh, and S.-Y. Lee, "Support vector machines with binary

tree architecture for multi-class classification," vol. 2, pp. 47–51, KAIST Press, 2004.

[17] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *In European Conference on Computer Vision*, Springer, 2006.

[18] 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, Jan. 1997.

[19] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pp. 65–72, IEEE, 2005.

[20] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005.*, pp. 65–72.

[21] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 726–733, IEEE, 2003.

[22] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[23] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, vol. 2, pp. 524–531, 2005.

[24] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[25] Z. Gao, A.-A. Liu, H. Zhang, G. ping Xu, and Y. bing Xue, "Human action recognition based on sparse representation induced by l1/l2 regulations," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1868–1871, Nov 2012.

[26] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua, "A mfom learning approach to robust multiclass multi-label text categorization," in *Proceedings of the twenty-first international conference on Machine learning*, p. 42, ACM, 2004.

[27] D. Gavrila, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[28] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 12, pp. 2247–2253, 2007.

[29] T. Guha and R. K. Ward, "Learning sparse representations for human action recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 8, pp. 1576–1588, 2012.

[30] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 188–195, IEEE, 2010.

[31] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15, p. 50, Manchester, UK, 1988.

[32] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, 1999.

[33] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," vol. 13, pp. 415–425, IEEE, 2002.

[34] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 128–135, IEEE, 2009.

[35] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 3, pp. 334–352, 2004.

[36] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent k-svd," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1697–1704, IEEE, 2011.

[37] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 166–173, IEEE, 2005.

[38] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[39] A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3d-gradients," 2008.

[40] A. Kläser, M. Marszalek, , and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," *BMVC*, 2008.

[41] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2046–2053, IEEE, 2010.

[42] S. Lacoste-Julien, F. Sha, and M. I. Jordan, "Disclda: Discriminative learning for dimensionality reduction and classification," in *Advances in neural information processing systems*, pp. 897–904, 2009.

[43] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.

[44] I. Laptev and T. Lindeberg, "Local descriptors for spatio-temporal recognition," *Spatial Coherence for Visual Motion Analysis*, pp. 91–103, 2006.

[45] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[46] I. Laptev and P. Pérez, "Retrieving actions in movies," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[47] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2169–2178, IEEE, 2006.

[48] Q. Le, W. Zou, S. Yeung, and A. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3361–3368, IEEE, 2011.

[49] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2548–2555, IEEE, 2011.

[50] A. Liu and D. Han, "Spatiotemporal sparsity induced similarity measure for human action recognition.," *JDCTA*, vol. 4, no. 8, pp. 143–149, 2010.

[51] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1996–2003, IEEE, 2009.

[52] C. Liu, Y. Yang, and Y. Chen, "Human action recognition using sparse representation," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 4, pp. 184–188, IEEE, 2009.

[53] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[54] W. Lu and J. Little, "Simultaneous tracking and action recognition using the pca-hog descriptor," in *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, pp. 6–6, IEEE, 2006.

[55] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[56] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Computer Vision–ECCV 2010*, pp. 183–196, Springer, 2010.

[57] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[58] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, pp. 1033–1040, 2009.

[59] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification.," in *ICML*, vol. 98, pp. 341–349, Citeseer, 1998.

[60] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2929–2936, IEEE, 2009.

[61] P. Matikainen, M. Hebert, and R. Sukthankar, "Representing pairwise spatial and temporal relations for action recognition," *Computer Vision–ECCV 2010*, pp. 508–521, 2010.

[62] J. D. Mcauliffe and D. M. Blei, "Supervised topic models," in *Advances in neural information processing systems*, pp. 121–128, 2008.

[63] A. McCallum, "Multi-label text classification with a mixture model trained by em," in *AAAI?99 Workshop on Text Learning*, pp. 1–7, 1999.

[64] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 104–111, IEEE, 2009.

[65] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence,*, vol. 27, no. 10, pp. 1615–1630, 2005.

[66] K. Mikolajczyk and H. Uemura, "Action recognition with motion-appearance vocabulary forest," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[67] T. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[68] T. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2, pp. 90–126, 2006.

[69] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[70] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 3, pp. 710–719, 2005.

[71] V. Parameswaran and R. Chellappa, "View invariance for human action recognition," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 83–101, 2006.

[72] G. Peyré, "Sparse modeling of textures," *Journal of Mathematical Imaging and Vision*, vol. 34, no. 1, pp. 17–31, 2009.

[73] R. Polana and R. Nelson, "Low level recognition of human motion (or how to get your man without finding his body parts)," in *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pp. 77–82, IEEE, 1994.

[74] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[75] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, "Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora," in

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1, pp. 248–256, 2009.

[76] D. Ramanan, D. Forsyth, and A. Zisserman, "Tracking people by learning their appearance," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 1, pp. 65–81, 2007.

[77] E. Ramasso, C. Panagiotakis, M. Rombaut, D. Pellerin, G. Tziritas, *et al.*, "Human shape-motion analysis in athletics videos for coarse to fine action/activity recognition using transferable belief model," *Electronic Letters on Computer Vision and Image Analysis*, vol. 7, no. 4, pp. 32–50, 2009.

[78] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3501–3508, IEEE, 2010.

[79] N. Rasiwasia and N. Vasconcelos, "Latent dirichlet allocation models for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2665–2679, 2013.

[80] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," *Machine Vision and Applications*, vol. 24, no. 5, pp. 971–981, 2013.

[81] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, june 2008.

[82] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*, pp. 430–443, Springer, 2006.

[83] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2564–2571, IEEE, 2011.

[84] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[85] K. Schindler and L. Van Gool, "Action snippets: How many frames does human action recognition require?," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[86] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3, pp. 32–36, IEEE, 2004.

[87] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," 2007.

[88] J. Stöttinger, B. Goras, T. Pöntiz, A. Hanbury, N. Sebe, and T. Gevers, "Systematic evaluation of spatio-temporal features on comparative video challenges," in *Computer Vision–ACCV 2010 Workshops*, pp. 349–358, Springer, 2011.

[89] J. Sullivan and S. Carlsson, "Recognizing and tracking human action," *Computer VisionECCV 2002*, pp. 629–644, 2002.

[90] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 11, pp. 1473–1488, 2008.

[91] N. Ueda and K. Saito, "Parametric mixture models for multi-labeled text," in *Advances in neural information processing systems*, pp. 721–728, 2002.

[92] S. Umakanthan, S. Denman, S. Sridharan, C. Fookes, and T. Wark, "Multiple instance dictionary learning for activity representation," in *International Conference on Pattern Recognition*, 2014.

[93] C. Wang, D. Blei, and F.-F. Li, "Simultaneous image classification and annotation," in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, pp. 1903–1910, 2009.

[94] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176, 2011.

[95] Y. Wang, P. Sabzmeydani, and G. Mori, "Semi-latent dirichlet allocation: A hierarchical model for human action recognition," in *Human Motion–Understanding, Modeling, Capture and Animation*, pp. 240–254, Springer, 2007.

[96] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," BMVC, 2009.

[97] H. Wang, M. M. Ullah, A. Klser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *University of Central Florida, U.S.A*, 2009.

[98] X. Wang, B. Wang, X. Bai, W. Liu, and Z. Tu, "Max-margin multiple-instance dictionary learning," 2013.

[99] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3360–3367, IEEE, 2010.

[100] D. Weinland and E. Boyer, "Action recognition using exemplar-based embedding," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–7, Ieee, 2008.

[101] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[102] G. Willems, J. Becker, T. Tuytelaars, and L. Van Gool, "Exemplar-based action recognition in video," in *British Machine Vision Conference*, pp. 1–11, 2009.

[103] G. Willems, T. Tuytelaars, and L. Van Gool, *An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector*, vol. 5303, pp. 650–663. ECCV, 2008.

[104] S. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[105] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.

[106] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Advances in neural information processing systems*, pp. 1537–1544, 2004.

[107] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Feature selection in face recognition: A sparse representation perspective," tech. rep., 2007.

[108] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pat-*

tern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 1794–1801, IEEE, 2009.

[109] A. Yilma and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 150–157, IEEE, 2005.

[110] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 984–989, IEEE, 2005.

[111] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Advances in Neural Information Processing Systems*, pp. 2223–2231, 2009.

[112] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2442–2449, Ieee, 2009.

[113] Q. Zhang and S. A. Goldman, "Em-dd: An improved multiple-instance learning technique," in *Advances in neural information processing systems*, pp. 1073–1080, 2001.

[114] Q. Zhang, S. A. Goldman, W. Yu, and J. E. Fritts, "Content-based image retrieval using multiple-instance learning," in *ICML*, vol. 2, pp. 682–689, Citeseer, 2002.

[115] Z. Zhang, Y. Hu, S. Chan, and L. Chia, "Motion context: A new representation for human action recognition," *Computer Vision–ECCV 2008*, pp. 817–829, 2008.

[116] Q. Zhang and B. Li, "Discriminative k-svd for dictionary learning in face recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2691–2698, IEEE, 2010.

[117] D. Zhang, F. Wang, L. Si, and T. Li, "M3ic: Maximum margin multiple instance clustering.," in *IJCAI*, vol. 9, pp. 1339–1344, 2009.

[118] X. Zhang, H. Zhang, and X. Cao, "Action recognition based on spatial-temporal pyramid sparse coding," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1455–1458, 2012.

[119] M.-L. Zhang and Z.-H. Zhou, "M3miml: A maximum margin method for multi-instance multi-label learning," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pp. 688–697, IEEE, 2008.

[120] J. Zhu, A. Ahmed, and E. P. Xing, "Medlda: maximum margin supervised topic models," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2237–2278, 2012.

[121] Y. Zhu, X. Zhao, Y. Fu, and Y. Liu, "Sparse coding on local spatial-temporal volumes for human action recognition," in *Computer Vision–ACCV 2010*, pp. 660–671, Springer, 2011.