An Action Recognition Framework For Uncontrolled Video Capture based on a Spatio-Temporal Video Graph

Iveel Jargalsaikhan

B.E, M.Tech

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University

School of Electronic Engineering

Supervisors: Prof. Noel E O'Connor, Dr. Suzanne Little

May 2017

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Sign: Student ID No.: 12210695

Date: May 24, 2017

Acknowledgements

This dissertation represents my research work during my PhD studies. I have been accompanied and mentored by numerous people, and it is now with great pleasure I acknowledge their support.

I sincerely thank my supervisor, Prof. Noel E. O'Connor. He has supported me continuously during my PhD studies and has afforded me with patience and knowledge. In addition, I would like to acknowledge the support of Dr. Suzanne Little, who has always mentored me on my PhD journey. It has been a great privilege for me to work along with many good friends and colleagues at Insight@DCU, especially folks at N207. Also, I would like to express my gratitude to my family.

Finally, I would like to thank Insight Centre for Data Analytics funded through Science Foundation Research for funding me during my studies and NVIDIA corporation for providing me the technical support.

Contents

List of Publications

[1] *Peer-reviewed:* Iveel Jargalsaikhan, Suzanne Little, Remi Trichet and Noel E O'Connor "Action recognition in video using a spatial-temporal graph-based feature representation" In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Germany, 2015.

[2] *Peer-reviewed:* Iveel Jargalsaikhan, Cem Direkoglu, Suzanne Little and Noel E O'Connor "An evaluation of local action descriptors for human action classification in the presence of occlusion" In *MultiMedia Modeling International Conference (MMM)*, Springer, 2014.

[3] *Peer-reviewed:* Iveel Jargalsaikhan, Suzanne Little, Cem Direkoglu and Noel E O'Connor "Action recognition based on sparse motion trajectories" In *International Conference on Image Processing (ICIP)*, IEEE, 2013.

[4] *Workshop:* Suzanne Little, Iveel Jargalsaikhan, Rami Albatal, Cem Direkoglu, Noel E O'Connor et al. "SAVASA project@ TRECVid 2013: semantic indexing and interactive surveillance event detection" In TRECVID, 2013.

[5] *Workshop:* Suzanne Little, Iveel Jargalsaikhan, Kathy Clawson, Hao Li, Marcos Nieto, Cem Direkoglu, Noel E O'Connor et al. "SAVASA Project@ TRECVid 2012: Interactive surveillance event detection." In TRECVID, 2012.

List of Symbols

K Visual dictionary size

H Bounding box height

W Bounding box width

G Spatio-temporal video-graph

V Set of video-graph nodes

E Set of video-graph edges

$\mathcal{E}(\mathcal{L})$ Total energy with label configuration $\mathcal{L}$

$\mathcal{E}_1$ Energy first term, *likelihood*

$\mathcal{E}_2$ Energy second term, *prior*

h Bag-of-Features (BOF) histogram

c_i Bag-of-Features (BOF) visual code word i^{th}

u_i Local descriptor vector

(x, y) Coordinate vector in spatial dimension

t Temporal index

$\mathbf{x}_i$ Local descriptor position vector in space and time

β Learned bias constant for SVM classifier

α Learned weight vector for SVM classifier

$\mathcal{C}$ Set of supervoxel regions

m Spatial cell size dimension

$\mathbf{I}$ Set of video frames

r_t Action candidate region at video frame t

f_c Color similarity term

f_d Descriptor similarity term

f_g Geometric similarity term

l Action category label

L Set of labels

w Optical flow field

$\mathcal{S}$ Video sub-volume

λ Smoothing constant for energy function

$d(i, j)$ Distance between nodes i,j

g_i Cluster group i

q_i Grid cell i

ϵ Maximum search radius

r_{sp} Spatial radius

r_{tmp} Temporal radius

v_{cat} Visual category

σ IOU (intersection over union) parameter

List of Tables

List of Figures

Abstract

" An Action Recognition Framework For Uncontrolled Video Capture based on a Spatio-Temporal Video Graph"

Iveel Jargalsaikhan

The task of automatic categorization and localization of human action in video sequences is valuable for a variety of applications such as detecting relevant activities in surveillance video, summarizing and indexing video sequences or organizing a digital video library according to the relevant actions. However it remains a challenging problem for computers to robustly recognize action due to cluttered backgrounds, camera motion, occlusion, view point changes and the geometric and photometric variances of objects.

An important question in action recognition is how to efficiently and effectively represent a video scene while maintaining the discriminative appearance, motion and contextual cues of the scene. Recently, local feature-based action recognition methods have gained popularity due to their simplicity and the-state-of-the-performance with various benchmarking datasets. However, the existing feature representation schemes e.g, Bag-of-Features, Fisher and VLAD, ignore the the spatial and temporal cues in the local features e.g, the spatio-temporal location and relationship. Inspired by this fact, this thesis aims to overcome the underlying limitation of the feature representation by proposing a new way to construct graph structure that aims to capture the spatial and temporal relationship between the local features while maintaining discriminative power. The key contributions can be summarized as follows (i) comprehensive evaluation of the several key elements in the recognition pipeline (ii) novel video graph based human action recognition framework (iii) evaluation of the different techniques involved in the video graph construction process and (iv) extension of the proposed video graph based video analysis to the challenging problem of action localization.

Chapter 1

Introduction

1.1 Overview

With the rapid increase in the number of digital videos and archives, the intelligent management and retrieval of video data have become one of the most active research topics in the field of computer vision. In particular, human action recognition is crucial in understanding the semantic meaning of a video sequence. Therefore, extensive research efforts have been devoted to developing novel approaches for action-based video analysis. A video analytics system is a critical component for many video management applications especially in surveillance and security, sports video and video archive search and indexing. In recent years, researchers have tackled the problem in various ways. However, due to the complex nature of the video, it is still considered an open problem.

There are various levels of abstraction for describing human action. Human activity can be categorized into three different levels [1] based on their complexity: *action primitive (atomic)*, *action* and *activity*. An *action primitive* is defined as a basic and atomic motion movement such as raising a hand, stretching a leg etc. In contrast, an *action* is built using these *action primitives*. For instance, walking, running and punching are examples of *action*. An *activity* is a higher semantic act such as playing sports or dancing that is comprised of a set of *actions*. Along with the categories of action, many works have varying degrees of explicit or implicit assumptions on the environment where the video data is captured. According to the taxonomy of Klaser [50], this environment can be classified into *controlled*,

Figure 1.1: Examples of three different data capture environment: *controlled* (first row), *constrained* (second row) and *uncontrolled* (last row)

constrained and *uncontrolled* as shown in Figure 1.1. *Controlled* video data is recorded in a way to encourage for automated processing. For instances, specific markers can be attached to human actor joints to capture their movement in space and time. In contrast, *constrained* video data can influence environmental parameters to a limited degree. This is the case for commercial video game platforms based on visual interfaces, such as the Microsoft Kinect, which makes certain assumptions such as a single person is fully visible or favourable lighting conditions, etc. *Uncontrolled video data* is recorded under conditions that cannot be influenced. This is the case for, i.e., TV and cinema style movie data, sports broadcasts, music videos, or user generated content. Only very few assumptions, if any, of a rather general nature can be made, such as humans are present and visible. The main challenges for analysing *uncontrolled* video data include multiple potential subjects of interest; changes in viewpoint, scale, and lighting conditions; partial occlusion of humans and objects, cluttered backgrounds, abrupt movement;

The thesis focuses on the problem of *action* recognition in *uncontrolled* video

data such as surveillance footage, movie and youtube videos. In particular, we are interested in the case where a video has been temporally segmented to contain only the execution of human actions (i.e.,., post action detection). The objective is to identify the action class for such video accurately. The remainder of this chapter continues with the motivation of this dissertation and introduces how action recognition from video data works and discusses the potential applications. The problem statement and the research questions are introduced followed by the thesis structure.

1.2 Motivation

Over the past two decades, information technology has been one of the most successful industries throughout the globe. From government to personal level, the digital revolution has had a tremendous influence on virtually all aspect of daily life. Growing storage capacity and network infrastructure have enabled us to store and analyse a large amount of data that was impractical a decade ago. Today, video data has become more accessible, and its role in our society, in general, is increasing. Video content can be routinely created by commonly used consumer devices such the mobile phone and digital cameras and the digital ecosystem now have the capacity to store and share the content in very efficient ways. For instance, the social media site Facebook has reported that the video content upload increased by 75% compared to 2014[1]. According to Cisco Mobile Traffic forecast, it is projected that the video content bandwidth will reach 15 exabytes[2] per month by 2020[3].

Despite the rapid accumulation of the video content and on-going improvements in computer vision, there is still no vision-based analytics system that is capable of performing the range of vision tasks at a human level. However, there are promising projects in the field of deep neural networks such as DeepMind

[1]http //tubularinsights.com/facebook-video-uploads-overtake-youtube/
[2]it would take an individual more than 5 million years to watch such amount of video.
[3]http //www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/

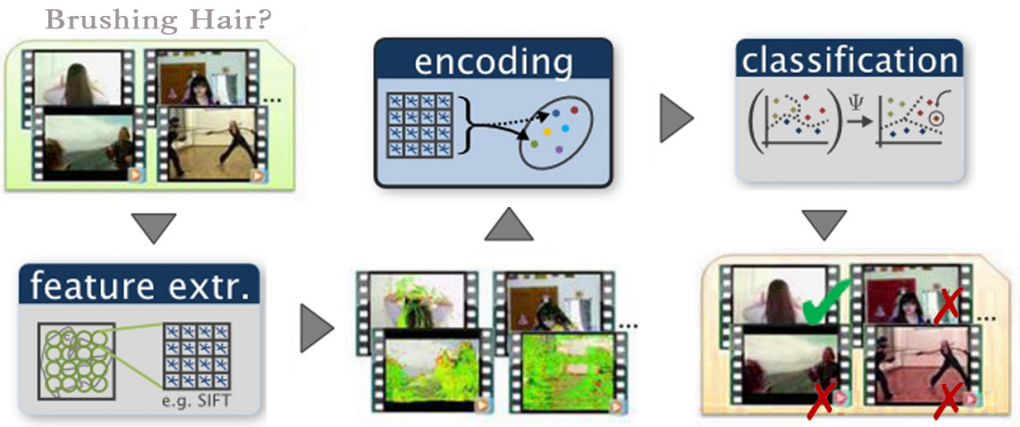


Figure 1.2: A typical processing pipeline for the local features-based approaches

AlphaGo[4] which has outperformed humans. The ability to automatically analyse and mine the semantic insight from videos is still in its development stage and has not progressed significantly into the application. For instance, video search in large scale database archives requires considerable effort associate with manual annotation. Web search engines commonly rely mainly on textual information, such as content descriptions or metadata, to retrieve relevant videos. In surveillance domain, intensive work to analyse the CCTV video footage is still mainly performed using human effort[5].

1.3 Problem Statement and Research Questions

Vision-based recognition of human action is becoming one of the most prominent fields of study in computer vision. There exists a vast body of literature on the subject. However, due to the complicated nature of the problem, it is still regarded as a challenging task especially in realistic scenarios captured in unconstrained circumstances. The difficulties arise from a significant amount of intra-class variation, occlusions, background clutter and illumination changes.

Numerous research has been developed over the last three decades in the action

[4]http //en.wikipedia.org/wiki/AlphaGo/
[5]https //www.newscientist.com/article/dn3918-smart-software-linked-to-cctv-can-spot-dubious-behaviour/

recognition domain. Earlier methods were focused on the structure of the entire human body, and the video representation is strongly related to the features such as silhouette and modelling the action regarding the evolution of shapes over time. More recently, local feature-based methods have gained popularity due to their simplicity and the-state-of-the-performance [57] [101] [50] in the various datasets with different degrees of complexity.

The majority of local feature-based action recognition pipeline consist of three main steps as shown in Figure 1.2 : feature extraction, video representation (encoding) and classification. In the first step, a set of local features is extracted from a given video. These features encode the image cues that is useful for recognition of the action in a numerical vector form. This is followed by the feature encoding step where the extracted features are aggregated to form a final representation of a video. This representation may be as simple as a histogram of feature occurrence or a high-level model such as action poses. Finally, a discriminative model for each action of interest is learnt using the obtained representation using the labelled training videos. Once a model is trained, given a test video, the system identifies the class of the given video sequence.

The local features have been proved to be not only efficient but also highly discriminative and less computationally intensive. They are robust even if there are variations in the scale, illumination and viewpoint and does not require the actor localisation in comparison to the global features. Therefore it is well suited for any scenario. However, the existing feature representation (encoding) schemes, e.g., Bag-of-Features, Fisher and VLAD, ignore the spatial and temporal cues in the local features e.g., the spatio-temporal location and relationship. The resulting representation not only limits the discriminative power of the action modelling but also prevents the identification of the region where the action is taking place. In particular, it hinders the performance significantly [89] for the *uncontrolled* video dataset that consists of videos captured in the complex environment that have significant background clutter and multiple actors or actions.

Recently, many authors noted that incorporating the spatial and temporal cues in an action recognition process can improve the system performance [74] [24]. In this thesis, the objective is to overcome the underlying limitation of the feature representation by proposing a graph structure that aims to capture the spatial and temporal relationship between the local features while maintaining its discriminative power. The graph is a simple and powerful representation method to model the relationship between the individual elements of the system. In particular, for 2D image understanding, it is a well-established data abstraction approach due to its intuitive representation of capturing the underlying image spatial structure. Consequently, based on the challenges and potential solutions discussed about the task of action recognition, the following research questions are addressed in this thesis:

RQ1. **Does incorporating spatio-temporal cues in the video representation stage increase action recognition performance?**

A novel video representation is the fundamental research problem in the action recognition that aims to build the compact description of the video. The research question (RQ1) explores the effectiveness of incorporating the spatial and temporal cue in the video representation process and investigate the impact on the overall system accuracy. To address this question, the following implementation and experimental works have been studied in the thesis.

- Implementation of the baseline recognition framework using a Bag-of-Features (BOF) representation approach that captures no spatio-temporal structure and the comprehensive evaluation of the key processing components in the pipeline. (Chapter 3)
- Implementation of the proposed video-graph action recognition framework that incorporates the spatio-temporal cue in the video representation process and comparative evaluation with the baseline

using several datasets (Chapter 4).

RQ2. **Can a graph-based video representation provide an effective method for incorporating spatio-temporal cues?**

As earlier mentioned, the major limitation of associated with the popular encoding schemes (i.e. the Bag-of-Features and Fisher) is that they only take account of the global statistics of local features for video representation, ignoring their spatio-temporal information. There are various ways to overcome this limitation. However, the research question explores the effectiveness of the graph-based representation to model the relationship between the local features explicitly. As part of the research question, we also propose a novel recognition process well suited for the video graph representation using a graph-cut optimisation. The following works have been performed to explore the research question:

- Implementation of the proposed video graph based recognition process and the video graph construction techniques (Section 4.4, Chapter 4)
- Formulation of the action recognition problem on the constructed video graph (Section 5, Chapter 4)
- Experimental evaluation of the proposed approach to the state-of-the-art (Section 6, Chapter 4)

RQ3. **What is the effective technique for constructing the video graph ?**

In practice, there are many ways to achieve a graph representation of video that encodes the spatio-temporal relationship between the local descriptors. The research question (RQ3) will explore various techniques involved in this process. In particular, we investigate their advantages and limitations in the context of action recognition task and their impact on the recognition accuracy. The following experimental works have been outlined to address the question:

- Comparative analysis of the different graph construction techniques and performance evaluation (Section 4.1.1, Chapter 5).
- Parameter sensitivity analysis of the selected graph construction techniques and their associated computational complexity (Section 4.1.2, Chapter 5).

RQ4. **Can the video graph be further improved to address the challenging problem of action localisation?**

The research question (RQ5) explores the effectiveness of the video graph-based technique for action localisation task. In this context, we develop a video graph that accommodates not only local features but also region-based features to encourage the localisation. The action localisation is performed by maximising the score associated with the node and the edge in the video graph. The following implementation and experiment are conducted to address the research question:

- Implementation of the improved video graph with the regional features i.e. region-based convolutional features for action localisation task (Section 4.1, Chapter 6).
- Implementation of the maximum-path finding (MPF) technique for action localisation on the constructed graph (Section 5, Chapter 6).
- The experimental evaluation of the proposed approach with two popular action localisation datasets (Section 6, Chapter 6).

1.4 Applications

Despite the challenging nature of the task, the successful recognition of human activities from videos enables several important applications in many interesting areas. The potential application fields are examined in this section.

Figure 1.3: Video surveillance camera and an example office for CCTV camera monitoring

1.4.1 Video surveillance

Video surveillance is increasingly becoming an integral part of modern society. Video surveillance systems aim to monitor people and object of interest using a variety video capturing devices. In particular, the closed-circuit television (CCTV) camera is widely adopted at large scale, and it can be found at airports, metro stations, public buildings and parking lots. For instance, the city of London has installed about 1 million closed-circuit television (CCTV) cameras and the capital, China has 0.8 million security cameras operating daily basis.

With the increased demands for and dependence on networks of devices, the effective video-based surveillance system that is capable of monitoring, identifying and preventing security breach is becoming a necessity. In particular, for human induced security threats such as fighting, burglary, disputes, harassment, etc. are the scenarios where an action recognition system is well suited. Furthermore, given the availability of vast amounts of surveillance videos, the labour intensive analytics process will be greatly improved by an automatic action detection system.

1.4.2 Home Monitoring and Elderly care

According to data from UN's World Population Ageing report 2015 [6], almost every country in the world is experiencing a rapid growth in the proportion of older people

[6]http //www.un.org/en/development/desa/population/theme/ageing/WPA2015.shtml

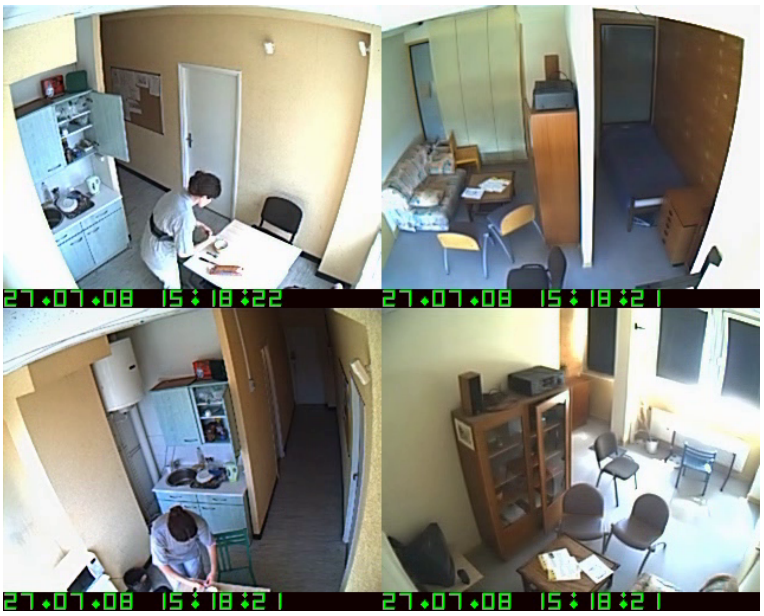


Figure 1.4: Sample video frames of Health Smart Home (HIS) dataset contain different activity scenarios such as Sleeping, Resting, Eating and Hygiene etc.

in their population. Globally, the number of people aged over 60 years is projected to grow by 56 percent from 901 million to 1.4 billion between 2015 and 2030. By 2050, the number is expected to double its size reaching nearly 2.1 billion and people aged 60 will outnumber the children aged between 0 - 15. Population ageing is poised to become one of the most significant societal transformation of the twenty-first century, with implications for nearly all sectors of society, from healthcare to economics.

The recently, the application of human action and behaviour recognition is becoming more relevant for the purpose of health monitoring of elderly people in the home environment. The automatic recognition of human action in daily life such as food preparation, walking, exercise will allow medical experts to devise strategies related to exercise, diet and treatment adherence. Related projects in this area are GatorTech [41] and Dem@care [9]. The system supports the wellbeing of the people in the home environment by providing feedback on the daily activities in the house, raising alarms when unexpected activities happen.

Figure 1.5: The application of gesture and action recognition technology for user interaction with computer without controller devices.

1.4.3 Human computer interaction

A further application area is human-computer interaction such as the computer games, for which video-based gesture and action analysis has gained a lot of attention. Popular related project is Microsoft's Project Natal[7]. The project's framework allows for full-body motion capture, facial recognition, voice recognition, and acoustic source localisation. This is achieved by combining information from several sensors: a video camera, a depth sensor and a microphone. This allows users to play video games without controller devices and to interact in a virtual world using their full bodies in a natural way. Therefore, the improved gesture and action recognitions will provide a natural and intuitive method of human communication with devices.

1.5 Thesis Structure

The remainder of the thesis is structured as follows:

- **Chapter 2** introduces a review of the literature in the action recognition domain. In this chapter, the prominent research works are structured into three class of approach. Also, a comprehensive summary of popular benchmarking datasets and their evaluation protocols is presented.

[7]https //techcrunch.com/tag/project-natal/

- **Chapter 3** presents a baseline recognition system based on the Bag-of-Features (BoF) model and comprehensively evaluates it. In particular, we discuss how different encoding schemes, codebook sizes and other parameters have an impact on the recognition performance using several datasets. Also, the baseline system performance will act as a performance reference to the proposed approaches in the following chapter.

- **Chapter 4** introduces a novel action recognition framework based on the video graph. The proposed approach aims to overcome the underlying weakness of the Bag-of-Features (BoF) by incorporating the spatio-temporal relationship among the local features as a graph structure. Furthermore, we explore the application of the *Graph-Cut* optimisation method from $2D$ image segmentation to $3D$ spatio-temporal volumes to investigate its effectiveness for action recognition in video. The effectiveness of the proposed framework is investigated and benchmarked with the state-of-art using the popular datasets.

- **Chapter 5** introduces the various strategies on the video graph construction and investigates the associated effect on the performance of the recognition process. In particular, we examine three techniques as a representative to the identified taxonomy of approaches and perform a comparative analysis regarding their impact on the recognition accuracy, parameter sensitivity and computational complexity.

- **Chapter 6** propose an action localisation framework based on the video graph. In the framework, we develop a video graph that accommodates not only local features but also region-based features to encourage localisation. The additional cues such as local, motion and region geometry are captured as the graph edges. We perform the localisation by the maximising the path score in the constructed video graph. The effectiveness of the proposed approach is investigated and benchmarked to the state-of-the-art using two different

localisation datasets.

- **Chapter 7** summarises and concludes the thesis highlighting the research contributions, addresses the research questions and discusses future research directions.

Chapter 2

Literature Review

2.1 Overview

This chapter introduces a review of the prominent works in action recognition (Section 2.2) and the related benchmarking datasets (Section 2.3). In particular, we identify the taxonomy of the existing methodologies (*global*, *local* and *mid-level*) based on the type of action feature and their associated video representation strategies in the recognition process.

2.2 Action Recognition Methods

Action recognition methods can be divided into three classes: *global*, *local* and *mid-level*. Global methods encode the observation in a holistic manner. These representations are typically obtained in a top-down fashion and on the basis of accurate human detector, background subtraction or tracking. In contrast, *local* approaches are based on a set of local features extracted from an image region called a *patch* in sparse or dense manner. This representation is considered to be resistant to the viewpoint changes and scales. *Mid-level* representations attempt to model an action using intermediate level parts designed to encode the spatio-temporal structure within the video data.

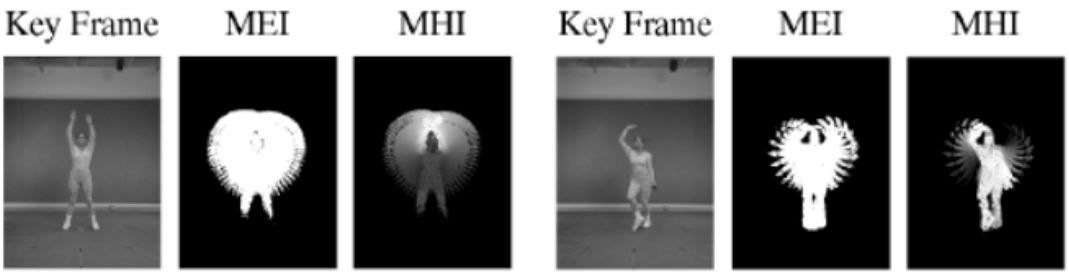


Figure 2.1: Motion History Images (MHI) and motion energy images (MEI) [14]

2.2.1 Approaches based on global features

Global representations encode the region of interest of an action in a holistic manner. The region of interest is typically captured through background subtraction or tracking. This class of approaches typically build an action model using silhouettes, counter or optical flow. The earliest works examining the use of silhouettes were done by Yamato et al [118] and Bobick et al [14]. In the work [14], authors integrated the silhouettes to form the accumulated frame differences over time, called the binary motion energy image (MEI) and motion history images (MHI) as shown in Figure 2.1. The MEI and MHI essentially encode the region where motion takes place. In [106], the scale and translation invariance technique is proposed for effective silhouettes representation. To deal with the scenes where background segmentation is difficult, Weinland et al [108] proposed to use the *chamfer distance* to eliminate the need for background subtraction. Instead of silhouettes, some works [26] [105] explored the use of contour based representation. In [26], the star skeleton model is used to represent the action contours. Wang et al [105] proposed the approach that integrates the contour and silhouettes into a single representation.

When there are multiple cameras available, silhouettes can be obtained from each to increase the effectiveness silhouettes based representation. In this context, Huang et al [116] used the camera setup where two orthogonally placed cameras at similar height and distance to the actor to obtain envelope shape. Another work [28] also

Figure 2.2: Space-time volumes for action recognition generated from silhouette information [110]

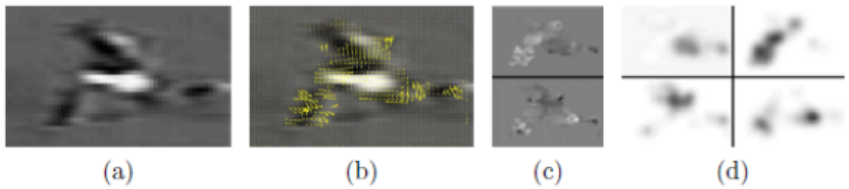


Figure 2.3: Optical flow based human-centered action modelling [34]

use orthogonally placed cameras to achieve the combined silhouettes features. Such methods are view-invariant but fail when the arms cannot be distinguished from the body. Weinland et al. [110] proposed to combine silhouettes from multiple cameras into a three-dimensional voxel model (See Figure 2.2). The method is discriminative however it requires accurate camera calibration. For action representation, authors use motion history volumes, which is an extended version of the MHI [14] and its view-invariance is achieved using Fourier transforms.

Instead of silhouettes, the global action representation can be built using motion information. This type of approaches do not rely on background subtraction unlike the silhouettes-based representations and encode the region of interest using dense optical flow or the frame difference from consecutive frames. An early work using optical flow is done by Polana et al [69] where the authors propose *temporal-texture* model that integrates both the first- and second-order motion statistics. The resulting representation can be used when background subtraction cannot be performed. Efros et al [34] calculate optical flow in

human-centered frames (See Figure 2.3) in sports footage. They divide the optical flow field into different components to form 4 distinct features which are separately matched. Ahad et al. [3] use these four flow components to address the self-occlusion artefact in an MHI approach. Furthermore, inspired by the physics concept, Ali and Shah [5] proposed the kinematic features (divergence, vorticity and symmetry) using the optical flow field. In the works [49, 59], the Viola-Jones object detection is extended by computing image features on the dense optical flow fields.

2.2.2 Approaches based on local features

This class of methods describe the video scene as a collection of local features or patches. The key advantages of these approaches are: (i) do not require accurate human detection/localisation and background subtraction, (ii) invariant to changes in viewpoint/scale, person appearance and (iii) able to deal with partial occlusions. Local features can be computed either densely or at a sparse set of regions. Local action recognition approaches, based on space-time representation [56] 2.4 , were introduced to extend the concept of local 2D image features already used in image classification to action recognition domain.

Space-time approaches are based on bottom-up strategies, and they build the action model by analysing spatio-temporal local regions in videos. A typical pipeline for space-time approach is as follows. Given a video sequence, first, detect and describe the salient points and then assign each region to a set of pre-computed dictionary features. Next, the action model is built using the statistics of the dictionary feature within the video sequence and used to assign an action class for unlabelled input video.

In this context, Laptev [56] first developed the space-time interest point detector by extending the notion of Harris Corner Detector into 3D space (See Figure 2.4). Scovanner et al. [79] proposed the 3-D extension of the SIFT descriptor, similar to cuboid features [32]. Liu et al. [64] presented a methodology to prune cuboid features

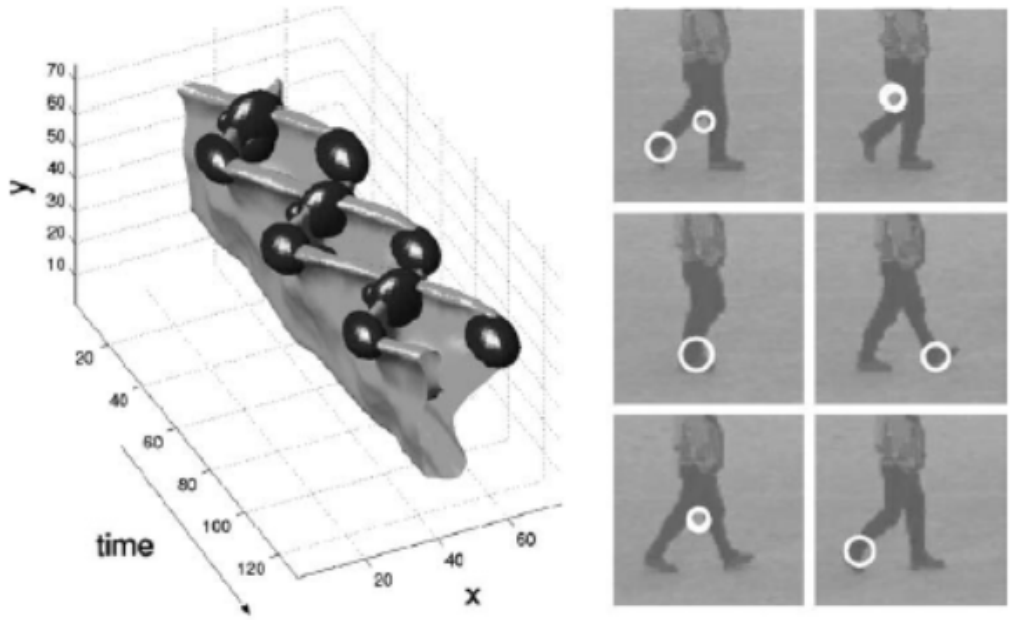


Figure 2.4: Spatio-temporal interest points from the motion of the legs of a walking person: (left) 3D plot of a leg pattern and the detected local interest points; (right) interest points overlaid on the single frames in the original video sequences [56]

to choose important and meaningful features. Bregonzio et al. [17] proposed an improved detector for extracting cuboid features, and presented a feature selection method similar to [64]. Rapantzikos et al. [73] extended the cuboid features to utilise colour and motion information as well, in contrast to previous features that only used intensities.

Trajectory based action recognition methods have gained significant interest. In these approaches, an action is interpreted as a set of space-time trajectories. The typical procedure for this type of approaches is that they compute dense or sparse trajectories and then they process these trajectories to represent and recognise actions. In this context, Sheikh et al. [82] represent an action as a set of joint trajectories in a four-dimensional space. To derive the view-invariant similarity between two sets of trajectories, they use an affine projection to obtain normalised trajectories of an action. Yilmaz et al [120] also use a set of joint trajectories to compare actions in videos obtained from moving cameras. In the work of [20],

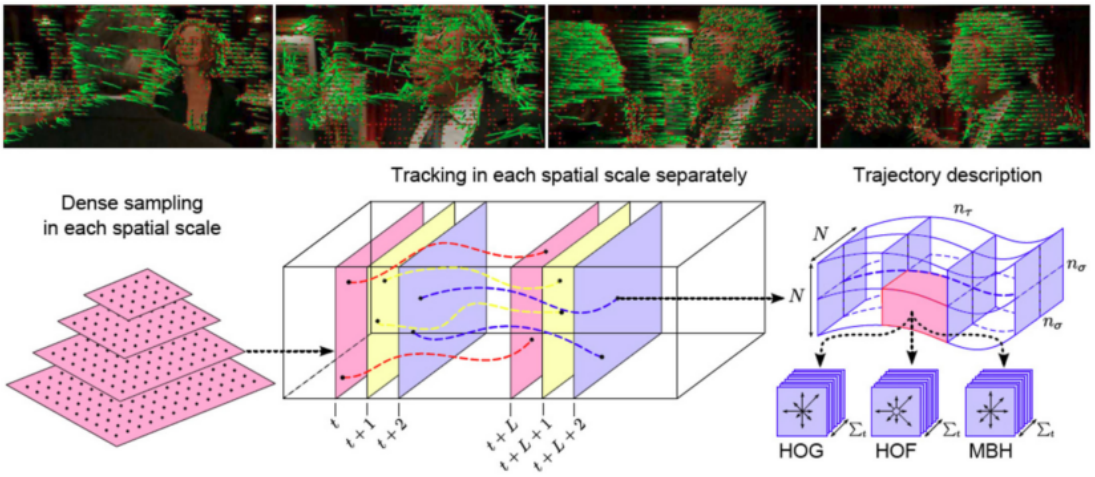


Figure 2.5: (Top) Visualization of human actions with dense trajectories. (Bottom) The dense trajectory feature generation pipeline; First, dense feature sampling is performed. Then, features are tracked using dense optical flow and feature descriptors are computed [101]

authors transform the joint trajectories into low-dimensional phase spaces to achieve view invariance and represent human actions. Rao and Shah [72] extract meaningful curvature patterns from the trajectories for action representation. Wang et al. [101] proposed the state-of-art approach based on the dense trajectories (See Figure 2.5. The dense points are sampled and tracked using the displacement information from a dense optical flow field for a short duration. Local descriptors of HOG, HOF and MBH are calculated along the resulting trajectories.

Recently, deep learning features for action recognition [46, 48, 84, 93] have been explored due to the great success in image based tasks [52, 85, 123]. Taylor et al. [93] used Gated Restricted Boltzmann Machines (GRBMs) to learn the motion features in an unsupervised manner and then resorted to convolutional learning to fine tune the parameters. Ji et al. [46] extended 2D ConvNet to video domains for action recognition on relatively small datasets, and Karpathy et al. [48] tested ConvNets with deep structures on a large dataset, called Sports-1M. However, these deep models achieved lower performance compared with shallow hand engineered local features [32, 50, 101]. This might be ascribed to two facts: firstly, available action datasets are relatively small for deep learning; secondly, learning

complex motion patterns is more challenging. Simonyan et al. [84] designed two-stream ConvNets containing spatial and temporal networks by exploiting the large ImageNet dataset[1] for pre-training and explicitly calculating optical flow for capturing motion information, and it achieved performance on par with the state-of-the-art. Weinzaepfela et al. [111] introduced a method to combine both local features and deep learning features by fusing with a track descriptor and achieved further improvement. This shows that combining the deep learning features with local features (hand-engineered) that are complementary to each other improves the performance.

2.2.3 Approaches based on mid-level representations

The approaches based on mid-level representation aim to recognise an action based on the video parts that encodes the spatial or temporal structure in the data. Early works have been focused on the stochastic methods to capture the temporal variability [112] [87] [42] [71]. Robertson et al. [78] model human action as a stochastic time sequence based on HMM. Brendel and Todorovic [19] used a time series of activity features for identification of the salient action region in the time domain using a Markov chain. These approaches, however, make the firm presumption that the duration of the action spans the whole length of the video and furthermore their action model is limited encoding only the temporal pairwise relationship between the video parts.

To overcome this problem, Wang and Mori [107] were the first to propose a hidden conditional random field (HCRF) part-based model that encodes spatial pairwise relationships. In particular, a human action was modelled as a configuration of parts of video observations whereby pairwise relationships among spatial patches are captured explicitly. The representation of the video part uses the combined features (global and local patch-based features). However, the method adopted the classification process that performs at a frame level while considering only the

[1] http //www.image-net.org/

spatial grouping.

Niebles et al. [68] extended the notion of a mid-level part from a spatial domain [19] [107] to take into account of the temporal evaluation of the video frames. The proposed method can accommodate both the global and local features. The discriminative model was derived from the popular part-based model [37] of the object recognition domain. The local features decompose the video volume into a set of temporal segments to construct temporal composition. However, the resulting model requires the ability to spatially localise action parts.

There is a considerable amount of work [61, 66, 113] that exploits the directed graph network to model the action classes. In particular, various types of Dynamic Bayesian Network (DBN) have been proposed for recognizing different activities in the literature. For instance, Muncaster et al [66] use a dynamic Bayesian network to perform complex event recognition. Wu et al [113] present a DBN that combines RFID and video data to infer the activity and object label. Also, Laxton et al. [61] formulate a hierarchical DBN taking account of temporal, contextual and ordering constraints to recognize complex activities.

Raptis et al [74] proposed an action-part based graphical model and formulated the action recognition task as a Markov random field (MRF) problem. However, this method is not generic and fine-tuned only with trajectory features to construct nodes in the video graph. On the other hand, Chen at al [24] introduced a sub-graph based model for detection and localisation. It uses high-level features, that relies heavily on person and object detection. However, the underlying assumption restricts its applicability where the actor's figure is occluded in the video scene.

2.3 Datasets

In this section, we present the most popular action recognition datasets that are used in evaluating action recognition methods.

Figure 2.6: Sample screen shots of action categories of the Weizmann actions dataset.

2.3.1 Weizmann actions

The Weizmann dataset is one of the earliest datasets created for action recognition analysis and was recorded in 2005. The dataset has a homogeneous background, and a single person is performing in each frame. It contains 90 low-resolution video sequences showing nine different people, each performing ten natural actions such as running, walking, skipping, jumping-jack, jumping forward, jumping in place, gallop sideways, waving two hands, wave one-hand and bending as shown in Figure 2.6. Each frame is accompanied by the respective actor silhouette as ground truth.

2.3.2 KTH actions

The KTH Royal Institute of Technology compiled the dataset [13] in 2004. It is considered as one of the most popular datasets in the computer vision community. It consists of six human actions (walking, jogging, running, boxing, waving, and clapping as shown in Figure 2.7). Each action is performed several times by 25 subjects. The sequences were recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. The background is homogeneous and static in most of the sequences. In total, the data

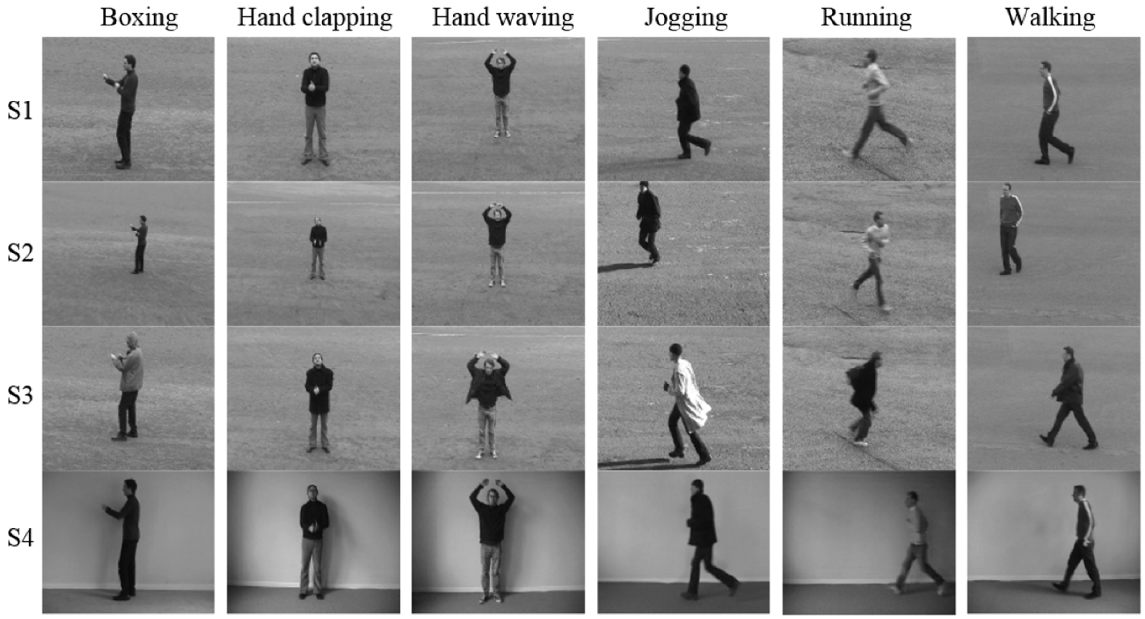


Figure 2.7: Example frames of action classes from the KTH actions dataset.

consists of 2391 video samples. The ground truth is given in a text-file format where each frame is annotated with meta-data that consists of the actor, action type and environment (static homogeneous background (SHB), SHB with scale variation, SHB with different clothing, SHB with illumination variation). The authors provide the evaluation protocol, and the average accuracy is a commonly accepted performance metric for the dataset.

2.3.3 UCF-Sports

The UCF-Sports dataset [21] consists of multiple action snapshots captured from various sporting events in 2008. It comprises an original set of actions performed in several scenes and viewpoints. The video sequences are collected from various sources such as *BBC Motion gallery* and *GettyImages*. The dataset contains 150 videos with 10 different action classes including *diving, golf swinging, kicking, lifting, horse-back riding, running, skating, swinging* and *walking* as shown in Figure 2.8. The dataset is considered challenging due to the large displacements that most of the actions contain, the cluttered background, and the large intra-class variability.

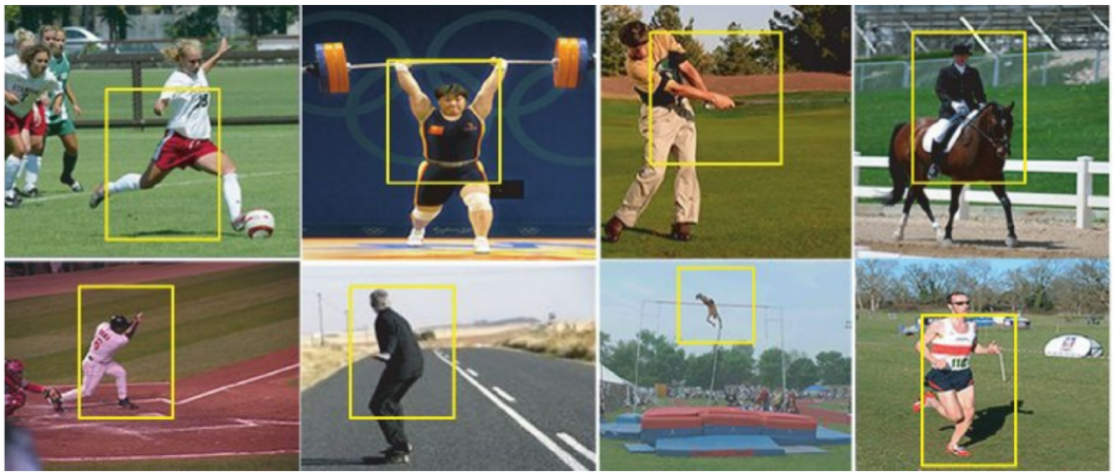


Figure 2.8: Example shots from UCF Sports action dataset.

2.3.4 YouTube Actions

The dataset [63] was compiled in 2009 with videos recorded from a video sharing website *YouTube*. It consists of 11 different action categories: *cycling, basketball shooting, diving, horseback riding, golf swinging, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking* and *walking a dog* as shown in Figure 2.9. The dataset is characterised by large variations in camera motion, object appearance and pose, viewpoint, cluttered background, lighting condition. For each action category, the video sequences divided into 25 groups with more than four action clips. The grouping is based on the similarity based on the actor, background, viewpoint, etc. The ground truth is formatted in VIPER [2] where the bounding box and action category is provided for each frame.

2.3.5 Hollywood Actions

The Hollywood Actions (HOHA) dataset [58] contains 430 videos with 8 different actions (See Figure 2.10). This dataset is extremely challenging: each video sequence, in addition to the action being performed, contains challenging

[2]Language and Media Processing Laboratory, Viper: The video performance evaluation resource, November 2011. http://viper-toolkit.sourceforge.net/

Figure 2.9: Screen shots of the 11 action classes of the Youtube action dataset [63].

conditions such as significant camera motion, rapid scene changes and occasionally significant clutter. Moreover, even though the included actions (e.g., sit down or kiss) can manifest themselves in a wide variety of conditions, only a tiny portion of them are sampled in the training set. Furthermore, many actions are not performed by a single agent (such as sit down) but involve interactions with other agents (kiss) or objects (get out of the car). The dataset is divided into a test set collected from 20 movies and two training sets (*automatic* and *clean* training set) obtained from 12 movies. In particular, the *automatic* training set is obtained using automatic script-based action annotation and contains 233 video samples with approximately 60% accurate action labels. In contrast, the *clean* training set contains 211 manually labelled video clips. The ground truth is generated by frame ranges and the corresponding actions.

2.3.6 HMDB51 and J-HMDB

The Serre lab at Brown University, USA, developed the HMDB dataset. The videos were obtained from several sources, i.e. *Prelinger* archive, *YouTube* and *Google* videos. The dataset comprises of 6849 clips divided into 51 action

Figure 2.10: Two action class examples from the Hollywood dataset: (*kissing* and, *answering a phone*)

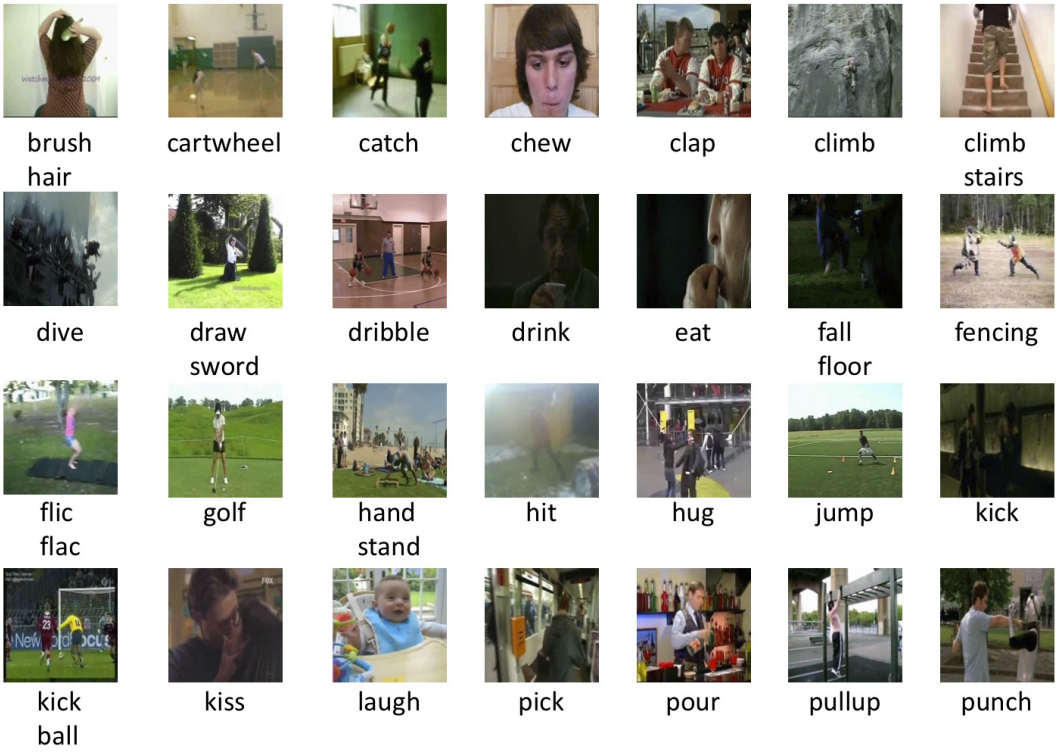


Figure 2.11: Sample video shots from the HMDB dataset.

categories, each containing a minimum of 101 clips (See Figure 2.11). To date, it is the largest and the most diverse action recognition dataset available. The action classes can be divided into five groups: general facial actions, facial actions with object interaction, general body movement, body movement with object interaction and human interaction. Each video is associated with a comprehensive meta-data information that includes visibility of body parts, camera motion, camera viewpoint respective to the actor, the number of people involved in the action and video quality. The dataset is considered to be very challenging due to the use of video clips obtained from real-world videos which suffer from a range of artefacts.

J-HMDB is a subset of the HMDB51 with fewer action categories. This dataset focuses on the single person action and it is considered more challenging than the original dataset. In particular, the dataset contains 21 categories involving a single person in action: brush hair, catch, clap, climb stairs,golf, jump, kick ball, pick, pour, pull-up, push, run, shoot ball, shoot bow, shoot gun, sit, stand, swing baseball, throw, walk, wave.

2.4 Summary

This chapter introduced the literature review of various approaches for action recognition and the related benchmarking datasets. We identify and assess three class of approach: Global, Local and Mid-level. In the literature, the earlier methods were developed focused on the relatively simple video settings that have clean background and single actor in the scene. This led to the development of the global methods that model the action in a holistic manner. These techniques are typically based on human localisation that is challenging to be accurately obtained in the uncontrolled environment. However, the local features based methods can be applied to various settings. The local features are known to be robust under the scale, viewpoint changes and and do not require a human body model or

localisation. The mid-level based approaches aim to model the action decomposed as parts and capture the spatial and temporal structure in the data. These techniques often result in a complex recognition process. There are many research works on-going to attain the effective mid-level representation.

In this thesis, we focus on methods based on local features and targets *uncontrolled* video capture. This class of approach is shown to be not only efficient but also provides excellent results compared to other complex approaches whilst being of a low computational complexity. However the existing local feature encoding schemes, i.e., BoF, Fisher, ignore the information of the spatio-temporal positions of features and relations among the features whereas such cues may be important for action recognition. In the remainder of the thesis, we propose to overcome this limitation by developing the notion of a video graph that aims to capture the spatial and temporal relationship between the local features as a graph structure. Chapter 4 presents a novel action recognition framework based on the video-graph. Chapter 5 focuses on the various video graph construction techniques to encode the spatio-temporal relationship and their potential impact on the recognition performance. Chapter 6 explores the action localisation problem using an extended version of the video graph. For validation and evaluation purpose, we will focus on the J-HMDB dataset in the thesis. This dataset contains videos with a variety of action classes captured in an uncontrolled environment and it is used for both the classification and localisation task. Also, each video is provided with the detailed meta-data information that enables further analysis on the experimental results. Next chapter presents the baseline framework and the evaluation of the different processing steps in the recognition system.

Chapter 3

Baseline System

3.1 Overview

This chapter introduces the baseline action recognition framework used in this thesis. In the literature, Bag-of-Features (BoF) model with SVM is the most popular method used to compare and benchmark the proposed feature detectors, descriptors and even recognition algorithms. In this chapter, this framework has been chosen to investigate the various key components, i.e., the local feature type, SVM kernel function, in the recognition process and the impact on overall performance. The baseline framework will be used as the basis for comparison performance in the following chapters.

3.2 Introduction

Over the years, a considerable amount of research has been conducted into human action recognition. Among the successful methodologies, the local feature based approaches have become the most popular. In particular, the Bag-of-Features (BoF) paradigm has become well-established in the field. Bag-of-Features (BoF) methods adopt a structure-less model that treats the feature descriptors as an orderless collection. Due to its simplicity and discriminative nature, the Bag-of-Features (BoF) based approach has become the standard recognition pipeline in the action recognition community to compare and evaluate the different algorithms and their various processing components

There is research carried out [70] for evaluation of the bag-of-features based recognition framework. Many efforts have been focused on the individual components of the framework in different settings. For instance, Wang et al [102] performed the comprehensive evaluation of various feature descriptors and encoding schemes with a standard bag-of-features model. However they mainly focused on the feature extraction technique of the framework and imposed a restriction on the rest of the framework such as constraining the dictionary size to a certain value and using the same kernel function in all experimental work. In contrast, this chapter aims to provide a comprehensive evaluation of the Bag-of-Features (BoF) framework with datasets of varying degrees of complexity. In particular, we investigate the choice of the local descriptor, kernel function and size of the codebook dictionary. Also, this chapter examines the impact of the choice of the local descriptor has on the performance of human action recognition in the presence of static occlusion. This question is crucial when designing a recognition framework to uncontrolled video data that is noisy, complex and incomplete. To our knowledge, evaluation and comparison of classification performance of local action description methods, in the presence of occlusion, has not been done in the past. However, several authors [109] [33] have evaluated the impact of occlusion on their own work. A number of key survey papers in human action recognition [70] [11] [2] stated the necessity of occlusion tolerant action recognition methods. In particular, Poppe [70] wrote " the question [of] how to deal with more severe occlusions has been largely ignored". The rest of the chapter is organised as follows:

- Section 3.3 gives an introduction to the standard recognition framework and describes each processing stage in the pipeline. Also, we introduce the popular classification technique - Support Vector Machines (SVM) and the several types of kernel and normalisation methods used for building an action model.

- Section 3.4.1 presents a comprehensive evaluation of a various processing

components of the framework and their effect on the overall system performance. In particular, we evaluate four state-of-the-art local descriptors, four different dictionary sizes in three prominent action datasets.

- Section 3.4.2 focuses on the evaluation of the learning step in the pipeline. In this section, we examine the popular kernel methods using a SVM classifier.

- Section 3.4.3 evaluates the local feature descriptors and their possible combinations in the presence of varying degrees of occlusion. The experimental work is focused on *static* occlusion and the objective is to understand how the missing local descriptors, i.e. due to occlusion, affect action recognition performance.

3.3 Baseline Recognition Framework

The standard recognition pipeline is adopted for the baseline recognition system as shown in Figure 3.1. The recognition pipeline consists of four main processes: *feature detection*, *feature description* and *video representation* followed by *classification*. The feature detection process aims to find a point or a set of points of the video that corresponds to a potentially informative local region. *Feature description* captures the local statistics of the patch or region. For the video representation, firstly, the visual dictionary is constructed by clustering the features extracted from the training stage. Given a video, feature descriptors are assigned to their nearest matching cluster centre (*visual-word*) from the visual dictionary. The histogram of the quantized feature descriptors will form a vectorized representation of the video sequence that is used to train or validate the classification model.

3.3.1 Feature Detection

Intelligent selection and detection of feature points plays an important role in an action recognition system. According to the sampling strategy, the feature detection process can be divided into two categories: sparse and dense. In the former methods,

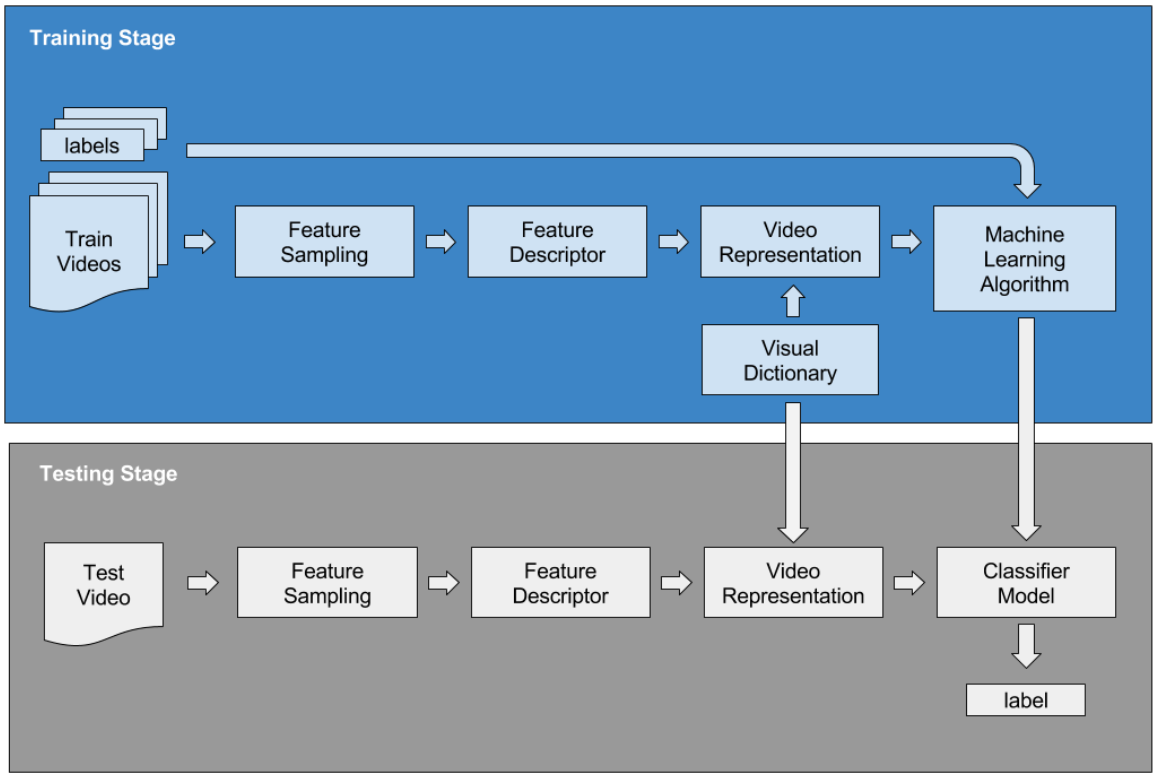


Figure 3.1: The typical *Bag-of-Features* (BoF) based action recognition framework pipeline

the feature points are selected from the salient regions determined by intensity changes. The popular methods include the Harris3D detector [86], and the Cuboid detector [115]. In contrast, the dense approaches sample the feature points at regular grids in space and time. According to the local feature survey [102], it was found that dense sampling methods typically outperform the sparse representation as it is more discriminative and robust. In particular, the dense trajectory-based representation resulted in outstanding performance. Hence in this research, the dense trajectory was adopted in the experimental work.

Dense Trajectory (DT) Detector

Wang et al [101] proposed a feature detection technique based on dense motion trajectories (DT). In this method, the dense motion trajectories are extracted at multiple different spatial scales using optical flow fields. They applied the global smoothness constraints to obtain more robust trajectories compared to tracking or matching points independently. Each point P_t is matched to the next frame by a median filtered optical flow as follows:

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * w)|_{(\overline{x}_t, \overline{y}_t)} \tag{3.1}$$

where w is the optical flow field, M is the median filter kernel and $(\overline{x}_t, \overline{y}_t)$ is the approximated position of (x_t, y_t). Then tracked points form a trajectory. To avoid the drifting problem[1], the length of the trajectory is limited to a fixed L long. When a trajectory length reaches this threshold, it is removed from the tracking process. A new trajectory is initialized when there are no tracking points found in a $W \times W$ spatial neighbourhood. To deal with the homogeneous area, the method adopts the criterion proposed by Shi and Tomasi [83]. During the feature point sampling process, the smaller eigenvalue of the autocorrelation matrix is checked. If the value is below a threshold, these points will not be included in the tracking. The dense trajectories are shown to be more robust than the trajectories obtained

[1] the error associated with the trajectory registration in the consecutive frames

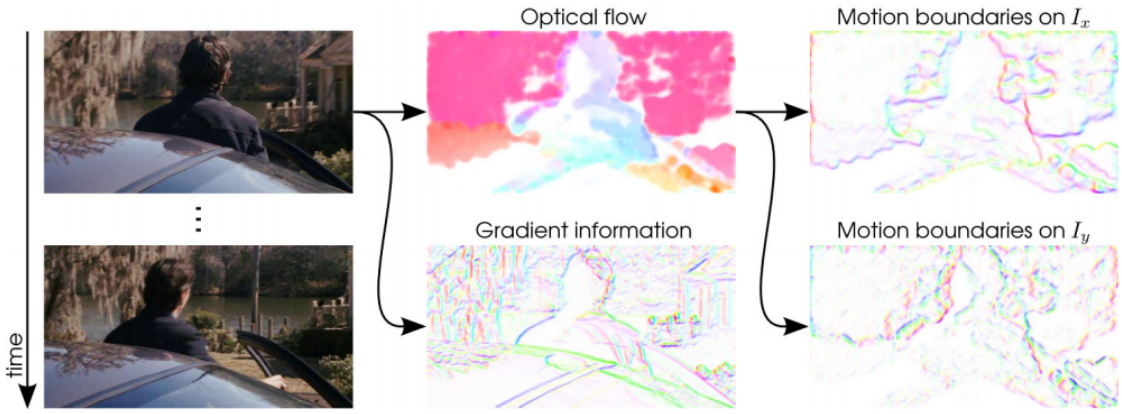


Figure 3.2: The information captured by the different descriptors; HOG (Gradient Information), HOF (Optical Flow) and MBH (Motion Boundaries) [101]

via the KLT (Kanade-Lucas) tracker.

3.3.2 Feature Description

Along with the determination of where feature exists in a video, there is a separate field of study focused on how to represent the local spatio-temporal region of the detected feature. The process is called feature description. The naive approach is to use the pixel intensity values of the particular region. Ideally, a local descriptor should have a compact representation of the local neighbourhood that is robust to various changes such as illumination, view points and camera angle. In practice, a spatio-temporal video patch is extracted from the detected features and information inside the volume is used to form a representative descriptor vector. The most popular descriptor methods typically use a histogram of responses of various gradient filters.

In the experiment, four different descriptors are used to describe the features of the detected trajectory features. The local neighbourhood is chosen as spatio-temporal volumes along the detected trajectories. The size of the volume is $N \times N$ pixels and L frames, with $N = 32$ and $L = 15$ in our experiments. For each trajectory, the four different types of descriptors are calculated, in a constructed 3D volume, to capture the different aspects of motion trajectory. Among the

existing feature descriptors, HOG and HOF [57] have shown to give excellent results on a variety of datasets. Therefore we have computed HOG and HOF along our trajectories. HOG (Histograms of oriented gradient) [29] captures the local appearance around the trajectories whereas HOF (histograms of optical flow) captures the local motion. Additionally, MBH (motion boundary histogram) which is proposed by Dalal et al. [30] and TD (trajectory descriptor) [101] are computed to represent the relative motion and trajectory shape. The feature vector dimensions of HOG, HOF, MBH and TD are described in detail in [101].

Trajectory descriptor (TRAJ)

The trajectory descriptor is proposed in the work of Wang et al. [101]. The descriptor encodes the shape characteristic of a given motion trajectory. Since motion is an important cue in action recognition, this representation allows motion characteristics to be exploited. The descriptor is straight-forward to compute using the points sampled on the trajectory in the image domain. Given a trajectory of length L, the shape is described by a descriptor vector S :

$$S = \frac{\Delta P_t, ..., \Delta P_{t+L-1}}{\sum_{t+L-1}^{j=t} |\Delta P_j|} \tag{3.2}$$

where $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. In our experiment, the trajectory length was chosen to be $L = 15$ video frames as recommended in [101].

The HOG/HOF descriptor

The HOG/HOF descriptors were proposed by Laptev et al. [57] for action recognition. To characterize local motion and appearance, the authors compute histograms of spatial gradient and optical flow accumulated in space-time neighbourhoods of the selected points. The points can be detected using any interest point detectors [56] [32]. In our experiment, these points are selected along the motion trajectory as in [101]. For the combination of HOG/HOF descriptors with interest point detectors, the descriptor size is defined by

$\Delta x(\sigma) = \Delta y(\sigma) = 18\sigma$, $\Delta t(\tau) = 8\tau$ where σ, τ are spatial and temporal scale factors. Each volume is subdivided into a $n_x \times n_y \times n_t$ grid of cells. The histogram of gradient orientations (HOG) and histogram of optical flow (HOF) are computed for each cell. Then the normalized histograms are concatenated into HOG/HOF descriptor vectors similar to the image descriptor SIFT. In the experiment, we use the grid parameters $n_x = n_y = 3, n_t = 2$ as recommended by the authors [57].

The Motion Boundary Histogram (MBH) descriptor

Dalal et al. [30] proposed the Motion Boundary Histogram (MBH) descriptor for human detection, where the second-order motion components are computed for the horizontal and vertical channel of the optical flow. The descriptor encodes the higher-order motion between consecutive frames. The MBH descriptor separates the optical flow field $I_\omega = (I_x, I_y)$ into its x and y component. Spatial derivatives are calculated for each of them, and orientation information is quantized into histograms, similarly to the HOG descriptor. We obtain an 8-bin histogram for each component and normalise them separately with the L_2 norm. Since MBH represents the gradient of the optical flow, constant motion (i.e., camera movement) information is suppressed, and only information on changes in the flow field (i.e., motion boundaries) is retained. In our evaluation, we used the MBH parameters used in the work of Wang et al. [101].

3.3.3 Video Representation

The video representation method used to describe the compact description of the scene. In particular, for local-features based methods, this process is known as encoding and corresponds to aggregating local features into a fixed-sized vector representation. This allows leveraging standard classification algorithms. In the literature, there is a number of existing local feature representation methods. However, the Bag-of-Bag-of-Features approach is the most popular due to its simplicity and good performance.

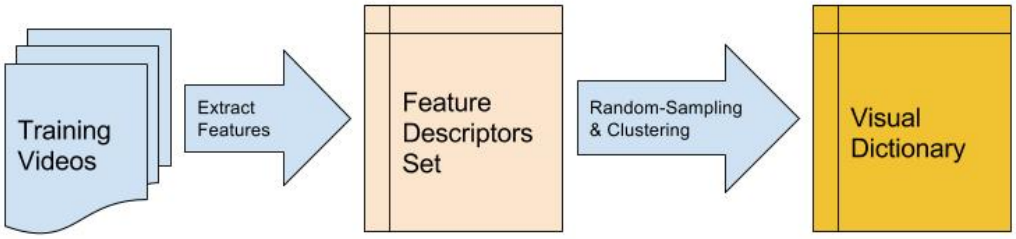


Figure 3.3: A typical process to construct the codebook dictionary

3.3.3.1 Bag-of-Features (BoF)

A popular video representation method based on the local features is the Bag-of-Features (BoF) model. It was originally adopted from the document retrieval domain and it is an orderless representation of frequencies of *visual-words* from a *dictionary*. In the BOF representation, the video is encoded in two main steps: dictionary construction and BoF-vector generation as shown in Figure 3.3.

- **Dictionary Construction**: There are many dictionary construction methods using different clustering methods such as k-means, hierarchical and spectral. Among them, k-means is the most popular method to construct a dictionary. Given a set of local features, the goal is to partition the feature set into clusters then each cluster will be referred as visual word.

- **BoF-vector generation**: Given a dictionary with K visual words, the objective of encoding is to compute a K-dimensional BoF histogram h for input features.

3.3.4 Classification Technique

In the last stage, a general model for each action of interest is learnt using the computed representation in a supervised manner. Once a model is learned, given a query video, the system classifies the video into given action classes. In the

literature, a support vector machines (SVM) is the most popular method used to compare and benchmark the proposed computer vision algorithms.

3.3.4.1 Support Vector Machines

Support vector machines (SVMs) become a standard classifier across the domain of image and video recognition approaches. SVMs have several key advantages such that its objective function is convex, although the training involves nonlinear optimisation. Hence the optimisation achieves the global minimum. Also, the number of support vectors in the resulting model is typically much smaller than the number of training points. There are many SVM implementations available [22] [47]. In the experimental works, an SVM classifier is chosen due its the popular use in many action recognition works [103] [59] [24] and because it allows the performance of the recognition pipeline to be compared with these prominent works.

An SVM constructs a hyperplane or a set of hyperplanes in an N-dimensional space that divides a set of given examples into two regions with maximum margin. In practice, SVMs perform poorly with a dataset that is not linearly separable. Hence for achieving non-linear classification, kernel methods are used that enables the classifier to operate on a higher dimensional implicit space without actually computing the coordinates of the data points in that space.

3.3.4.2 Kernel Methods

Kernel methods provide a systematic and principled approach for training learning algorithm and have a good supporting theoretical background. In a nutshell, this technique uses kernel functions, which transform the low dimensional feature space into higher dimensions to uncover more hidden characteristics about the data points.

For an input feature vector x, the kernel method $\phi(\cdot)$ maps the feature vector from input space $\mathcal{X}$ into a higher-dimensional space $\mathcal{H}$ where the goal is to separate two sets of points. The Figure 3.4 shows the effect of the mapping of

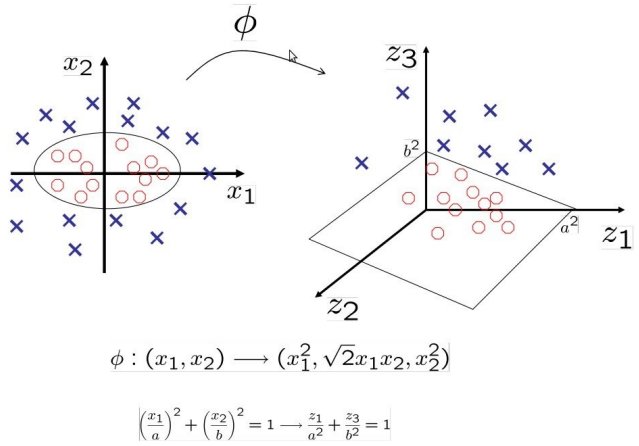


Figure 3.4: Let us assume that data separating surface is the ellipse (non-linear) drawn on the left figure. However, transforming the data into a 3-dimensional space through the mapping shown in the figure would make the problem much easier since, now, the points are separated by a linear plane. This embedding on a higher dimension is called the *kernel trick*

$\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ where the two dimensional feature vector is transformed into higher dimension where linear structure emerges. In practice, the explicit mapping is computationally expensive. In the execution of an SVM, only inner products between data vectors are considered i.e expressed in $\langle\phi(\mathbf{x}, \mathbf{x}')\rangle = \phi^T(\mathbf{x})\phi(\mathbf{x}')$. The specific map function $\phi(\cdot)$ is used that allows to compute inner product directly from $\mathbf{x}$ and $\mathbf{x}'$ without explicitly calculating $\phi(\mathbf{x})$ and $\phi(\mathbf{x}')$. This computational technique is known as the *kernel trick*.

A kernel is a symmetric function of two arguments and its main property is that it implicitly defines a mapping $\phi(\cdot)$ from $\mathcal{X}$ to a Hilbert space $\mathcal{H}$ such that $k(\mathbf{x}, \mathbf{x}') = \langle\phi(\mathbf{x}, \mathbf{x}')\rangle$. Due to the fact that they correspond to the inner products in higher dimensional space, a kernel can be considered as a similarity measurement between data points. Many different types are kernels known and the popular functions used in the action recognition task are shown in Table 3.1.

	Type	Formulation
Kernel	Linear	$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
	Jenson	$K(\mathbf{x}, \mathbf{x}') = \frac{x}{2} log_2 \frac{x+x'}{x} + \frac{x'}{2} log_2 \frac{x+x'}{x'}$
	Radial Basis Function (RBF)	$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \mathbf{x} - \mathbf{x}'^2)$
	Intersection	$K(\mathbf{x}, \mathbf{x}') = \min(x, x')$
	Chi-Square (χ^2)	$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \sum_i \frac{(\mathbf{x}_i - \mathbf{x}'_i)^2}{(\mathbf{x}_i + \mathbf{x}'_i)})$

Table 3.1: Popular kernel functions and feature normalization techniques

3.3.4.3 Normalization

Normalization, also referred as *"re-scaling"* or *"standardizing"* in machine learning literature, is an effective method used to standardize the range of the feature vector attributes. Since the range of the raw training data varies widely, the learning algorithm results in poor performance without normalisation. For instance, normalisation is an important step for BOF feature vectors as the number of local features extracted from video sequences varies widely. The practical advantage is that it avoids numerical difficulties during the calculation. Kernel values usually depend on the inner product of feature vectors, i.e. the linear kernel and polynomial kernel, and large attribute values may result in numerical overflow issue.

There are various way to achieve normalization and the optimal technique is typically determined heuristically. The different methods of normalization, as listed below, can be used to normalize the feature vector for classification:

- Min-Max : $\overline{\mathbf{x}} = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})}$
- $l^2 - norm$: $\overline{\mathbf{x}} = \sqrt{\sum_i \mathbf{x}_i^2}$

3.4 Experimental Evaluation

In this section, we present the experiments carried out in order to determine the optimal configuration for the baseline action recognition method. The experimental work can be categorised into three parts and structured in the

following sub-sections. Section 3.4.1 presents a comprehensive evaluation of a various processing components of the framework and their effect on the overall system performance. In particular, we evaluate four state-of-the-art local descriptors, four different dictionary sizes in three prominent action datasets. Section 3.4.3 evaluates the local feature descriptors and their possible combinations in the presence of varying degrees of occlusion. The experimental work is focused on *static* occlusion and the objective is to understand how the missing local descriptors, i.e. due to occlusion, affect the action recognition performance. Section 3.4.2 focuses on the evaluation of the learning step in the pipeline. In particular, we examine popular kernel methods along with the different feature normalisation techniques with a standard SVM classifier.

3.4.1 Evaluation of Local Features and Codebook Generation

3.4.1.1 Evaluation Framework

This section presents the experimental work focused on the feature description element in the recognition pipeline and aims to provide a guidelines on the effective type of feature in the presence in the presence of occlusion. In particular, we choose four widely used local descriptors, namely HOG, HOG, HOF, MBH and TD that discussed in detail in the previous section. For feature detector, we use the dense trajectories method. The dense trajectory (DT) detector source code from the author's website was used [2]. We choose the default parameter settings for the feature detector. Regarding codebook sizes, a subset of 250,000 descriptors sampled from the training video, where the codebook dictionaries with different size are formed using k-means. In the experiment, we use the mini-batch K-Means algorithm proposed by Sculley [80]. Then each descriptor type is assigned to its nearest cluster centroid using the Euclidean distance. The co-occurrence histogram with a dimension of k is constructed for each type of features to represent the BoF. For evaluating the combination of the different local descriptors, the co-occurrence

[2] http://lear.inrialpes.fr/people/wang/dense_trajectories

histograms of the feature types are concatenated to form a single feature vector. Finally, since the number of extracted trajectories may change depending on the given video, the magnitude of the combined feature histogram undergoes normalisation. This is achieved as $F = \frac{F'}{|F'|}$ where F is the normalized feature vector, F' is the vector before the normalization, and $|F'|$ is the l^2 norm of the vector. The normalised feature vector represents actions performed in the videos.

3.4.1.2 Classification Technique

A support vector machine (SVM) with a χ^2 kernel is used as a classifier to evaluate the performance of different local descriptors and codebook sizes. The χ^2 kernel function is a standard kernel method that is widely used in many recognition tasks. For multi-class classification task, a *one-against-the-rest* approach is used during the during training process. Next, we present the experimental results for various datasets with different descriptor and dictionary size.

3.4.1.3 KTH Dataset

Table 3.2 presents the results for the KTH dataset with different descriptor and dictionary size (k) combinations. The table shows average accuracy on a total of 6 action class of the dataset. For dictionary size, there is a positive correlation with the accuracy. The best result (94 %) is achieved with $k = 3000$ which is less than dictionary size $k = 4000$ that many works [97] [57] assumed to be the optimum value for BoF based action recognition systems. The explanation can be related with the simplicity of KTH dataset with a relatively small number of action class. Thus, a low level of partitions of descriptor space can result in a sufficiently discriminative dictionary model. Regarding descriptor, MBH and HOF consistently perform well in comparison with TRAJ and HOG. This suggests that motion is the effective cue in action recognition. In particular, MBH captures the higher order motion components that are resistant to the camera motion [101]. However, the top performance is achieved by combining all descriptor that results in a combined

description of the local region.

3.4.1.4 UCF-Sports

Table 3.3 presents the results for UCF-Sports dataset. Similar to the result with KTH dataset, a positive trend is observed for dictionary size with average accuracy. However, a dictionary size $k = 4000$ gives the best result performance. The UCF-Sports dataset contains a total of ten action classes compared to six for KTH. This might have influenced the optimal dictionary size as a higher dimensional dictionary is required to discriminatively identify between the increased number of classes. In terms of descriptor, again, MBH and HOF outperform the rest. The descriptor combination strategy improves the result further in comparison with the individual descriptor.

3.4.1.5 HOHA

The experimental result for HOHA dataset is presented in Table 3.4. The table shows mean-average-precision (MAP) on a total of 8 action class where the action segments are collected from original Hollywood movies [57]. The HOHA is considered to be a highly complex dataset [101] [74] [17] as it contains significant shot boundaries and intra-class variations. As expected, the combined descriptor produces the best result followed by MBH and HOF. In terms of dictionary size k, MAP increases along with k and it reaches its peak at $k = 4000$. Similar to the result with UCF-Sports, a higher value (k) performs well with HOHA. It demonstrates that a higher dimensional dictionary space is suited to deal with a video database that contains complex scenes.

3.4.1.6 Discussion

In summary, for effective performance, a dictionary size in range of $k = 3000$ to $k = 4000$ is a good choice. It is observed from the experiments that (i) the complexity of a video database (camera motion, shot boundary, etc.) and (ii) the diversity in

Vocabulary	TRAJ	HOG	HOF	MBHx	MBHy	Combined
1000	71.8%	72.4%	78.4%	71.3%	82.1%	82.6%
1500	65.6%	71.4%	80.8%	70.8%	81.1%	92.4%
2000	73.9%	72.2%	80.7%	70.1%	84.9%	94.0%
2500	68.4%	70.6%	80.0%	70.6%	84.5%	93.3%
3000	66.5%	71.4%	81.9%	71.5%	83.0%	94.0%
3500	74.7%	72.1%	81.1%	71.0%	82.8%	94.0%
4000	66.0%	71.4%	81.5%	70.5%	84.0%	92.7%
5000	65.6%	71.4%	81.1%	70.5%	83.0%	92.4%
6000	65.6%	70.6%	80.0%	70.1%	82.8%	91.6%

Table 3.2: The mean average accuracy (MAP) performance on the KTH dataset

Vocabulary	TRAJ	HOG	HOF	MBHx	MBHy	Combined
1000	46.6%	50.0%	60.6%	49.6%	62.0%	71.2%
1500	48.4%	52.3%	62.4%	52.2%	63.3%	72.6%
2000	49.1%	50.7%	61.3%	54.0%	62.8%	71.2%
2500	47.7%	50.2%	60.7%	52.3%	65.0%	73.6%
3000	47.3%	52.3%	61.0%	52.7%	64.8%	74.0%
3500	51.6%	51.6%	61.4%	50.3%	62.7%	74.0%
4000	52.0%	49.3%	60.9%	50.7%	62.8%	73.5%
5000	51.6%	49.3%	60.7%	50.3%	62.7%	72.9%
6000	47.7%	48.5%	59.5%	49.6%	62.0%	71.2%

Table 3.3: The mean average accuracy (MAP) performance on the UCF-Sports dataset

the action classes to be identified should be considered for choosing an optimum value k. Also, it is found that the choice of descriptor is crucial for accurate action classification. A poor selection can result in a considerable loss in accuracy. For instance, for the KTH dataset, the average accuracy performance with HOG is less than 30% in comparison with MBHx. The motion-based descriptor (HOF and MBH) is proved to be the most efficient and consistently performs best in all dataset. Finally, the descriptor combining strategy (concatenation) is found to be a simple and effective way to increase overall performance.

3.4.2 Evaluation of different kernel functions

This subsection presents the evaluation of the effect of different kernel method using the SVM classifier model. The performance of the SVM classifier strongly influenced

Vocabulary	TRAJ	HOG	HOF	MBHx	MBHy	Combined
1000	13.9%	11.0%	14.6%	11.7%	18.1%	32.9%
1500	14.8%	11.9%	15.3%	12.1%	19.1%	31.6%
2000	15.4%	12.4%	17.6%	13.6%	20.3%	33.0%
2500	14.2%	11.3%	17.8%	13.0%	22.4%	32.7%
3000	12.5%	12.5%	22.3%	11.2%	23.6%	34.9%
3500	12.1%	11.2%	22.9%	12.0%	25.0%	35.8%
4000	11.6%	11.5%	24.0%	12.4%	25.9%	36.3%
5000	11.6%	11.2%	24.0%	12.4%	25.5%	36.1%
6000	10.9%	10.9%	17.8%	11.2%	22.4%	35.5%

Table 3.4: The mean average accuracy (MAP) performance on the HOHA dataset

Kernel Type	TRAJ	HOG	HOF	MBHx	MBHy	Combined	Average
Jenson	15.3%	19.2%	22.4%	19.6%	32.9%	38.5%	**24.6%**
Intersection	14.2%	15.1%	21.6%	16.9%	28.1%	36.8%	22.1%
Chi-Square	14.8%	17.5%	23.5%	19.6%	30.5%	37.8%	**23.9%**
Linear	10.1%	11.9%	18.9%	15.5%	28.3%	**37.0%**	20.3%

Table 3.5: The kernel function performance with different descriptor on the HOHA dataset

by the type of kernel function utilised in the algorithm. In particular, four different kernel functions (Table 3.1) are studied with the baseline evaluation framework mentioned in Section 3.4.1. In the experiment, the HOHA dataset is used. The dataset is characterised by videos collected from different sources of Hollywood Movies and has huge variation of people and actions. The evaluate the kernel function on this dataset, we use the evaluation splits provided by the authors and report the results in MAP (mean average precision) over the action classes.

3.4.2.1 Results

The results are presented in both Table 3.5. On average, *Jenson*(23.9%) kernel outperforms followed by χ^2 (24.6%), *Intersection* (22.1 %) and *Linear* (20.3 %) as shown in Table 3.5. Interestingly, the many previous works [50] [101] have assumed that χ^2 kernel is well suited and used as the default kernel. Another useful observation is that the *Linear* kernel outperforms *Intersection* and obtains results close to that of χ^2 using the combined descriptor (the concatenated vector of

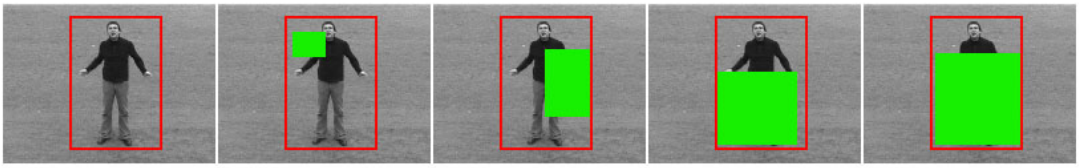


Figure 3.5: The sample shots where the different degree of random occlusion is applied into KTH video sequence. The red boundary is manually drawn in order to set an action boundary for each action performer. The green rectangles are occlusion regions randomly selected with 4 different occlusion sizes: 10%, 25%, 50% and 75% of the active region

TRAJ, HOG, HOF and MBH).

3.4.3 Evaluation of local descriptors in the presence of occlusion

This section examines the impact that the choice of the local descriptor has on action classification performance in the presence of static occlusion. This question is important when selecting a suitable descriptor for a local-feature based recognition pipeline for *uncontrolled* video data that is noisy, complex and incomplete. In an *uncontrolled* environment, it is natural that a human can be occluded by an object while carrying out different actions. However, it is unclear how the performance of the descriptors are affected by the associated loss of information. This subsection evaluates and compares the classification accuracy of different local descriptors in the presence of varying degrees of static occlusion. We consider the same local descriptors investigated previously, namely Trajectory (TRAJ), Histogram of Orientation Gradient (HOG), Histogram of Orientation Flow (HOF) and Motion Boundary Histogram (MBH). In the experiment, the descriptors are combined with a standard bag-of-features (BoF) representation and a SVM classifier for action recognition. We investigate the performance of these descriptors and their possible combinations on varying amounts of artificial occlusion in the KTH action dataset.

3.4.3.1 Synthetic Occlusion

Occlusion may occur due to static and dynamic occluding objects. For example, if an action performer is occluded by a moving object like a moving car or a person, it is considered as dynamic occlusion. On the other hand, the occluding object may be static like a building or a table in which case an occlusion represents static occlusion.

In our experiment, we focus our attention on *static occlusion*. Our objective is to understand how the missing action features, i.e. because of *static occlusion*, affect action classification performance. To model static occlusion, we occlude human action regions with rectangular shaped uniform colour objects, so that the action descriptors are not extracted within those regions. The uniform colour ensures no interest points are detected.

Since the KTH action dataset does not contain any occlusion, we have integrated random static occlusion only for the test set sequences. First, action boundaries are manually selected in each test sequence as a bounding box as shown in red boundary in Figure 3.5 . The action boundary (AB) should be selected with a specific height H_{AB}, width W_{AB}, position (x_{AB}, y_{AB}), in order to accommodate the region of video where the action is performed. Once we label the action boundaries for all test video sequences, occlusion bounding box (OB) is automatically generated within the action boundary region specified by H_{AB}, W_{AB},x_{AB}, y_{AB} with varying sizes of occlusion area $A(OB)$. The occlusion position is randomly generated and remained static for each test sequence. In our experiment, we have chosen the occlusion areas $A(OB)$ to be 10%, 25%, 50% and 75% of the action boundary area $A(AB)$ as shown in Figure 3.5. In a given action boundary AB and occlusion percentage $Occ\%$, the parameters H_{OB}, W_{OB}, x_{OB}, y_{OB} of the occlusion bounding box OB are randomly selected as follows:

$$H_{OB} \in [H_{AB} - (1 - Occ\%) \times H_{AB}, H_{AB}] \tag{3.3}$$

$$W_{OB} \in [W_{AB} - (1 - Occ\%) \times W_{AB}, W_{AB}] \tag{3.4}$$

$$x_{OB} \in [x_{AB}, x_{AB} + (W_{AB} - W_{OB})] \tag{3.5}$$

$$y_{OB} \in [y_{AB}, y_{AB} + (H_{AB} - H_{OB})] \tag{3.6}$$

where $Occ\% = \frac{A(OB)}{A(AB)}$ and H_{AB}, W_{AB}, (x_{AB}, y_{AB}) is height, width and top-left corner coordinate of the action boundary box, AB, whereas H_{OB}, W_{OB}, (x_{OB}, y_{OB}) is height, width and top-left corner coordinate of the occlusion boundary box, OB, and $H_{OB}, W_{OB}, x_{OB}, y_{OB} \in \mathbb{N}$.

3.4.3.2 Experimental Results

Table 3.7 shows the ranking of different combinations of descriptors in the partial occlusion case based on F-Score. The F-Score measure is the harmonic mean of precision and recall to balance their trade-off and defined as $F = 2 \times \frac{precision \times recall}{precision + recall}$. The best three combinations are TRAJ+MBH (90.1 %), TRAJ+HOF+MBH (88.9%) and HOF+MBH (88.4%). The worst performance is with HOG and HOF features. HOG descriptor obtained 64.3%, and HOF descriptor obtained 83.3%, and their combination is 83.5%.

The heavy occlusion ranking is presented in Table 3.8. TRAJ (67.7%), TRAJ+MBH (66.9%), TRAJ+HOF+MBH (64.7%) combinations perform best. The HOG, HOF and their combination perform poorly. The best descriptors are TRAJ, MBH and their combination. They consistently outperform any other combination for different scales of occlusion area in our experiments.

We now present experimental results for various descriptor combinations. We use multi-class classification where we apply the one-against-rest approach and compare the performance based on precision, recall and F-score. The scores are reported as an average of the 6 action classes. To measure impact of occlusion, we compute the above-mentioned scores for four different cases of occlusion: 10%, 25%, 50% and 75% occlusion of the action area. We also group the cases into partial occlusion (10% - 25% occluded) and heavy occlusion (50% - 75% occluded). The classifier is trained with non-occluded training data. Therefore all occlusion cases are classified

Descriptor Combination				Recall					Precision				
				No	Partial Occ		Heavy Occ		No	Partial Occ		Heavy Occ	
TRAJ	HOG	HOF	MBH	Occ.	10%	25%	50%	75%	Occ.	10%	25%	50%	75%
			✓	91.2%	89.1%	87.3%	71.8%	49.1%	91.6%	89.8%	88.2%	77.5%	68.3%
		✓		87.0%	87.2%	79.7%	68.5%	45.8%	88.6%	88.0%	81.4%	73.9%	63.6%
		✓	✓	91.2%	89.1%	87.7%	76.9%	50.0%	91.8%	89.8%	88.5%	81.7%	67.6%
	✓			74.5%	69.4%	62.5%	46.8%	26.9%	82.0%	80.8%	74.2%	65.3%	60.4%
	✓		✓	89.8%	88.7%	84.0%	70.4%	46.8%	90.5%	89.6%	85.8%	76.7%	73.9%
	✓	✓		88.4%	87.2%	81.1%	72.2%	44.4%	89.8%	88.7%	83.2%	77.3%	69.1%
	✓	✓	✓	89.8%	89.6%	84.9%	74.1%	49.1%	90.7%	90.6%	86.4%	79.6%	74.4%
✓				87.4%	84.9%	81.5%	79.6%	57.0%	88.7%	86.6%	84.1%	84.3%	73.1%
✓			✓	92.1%	93.4%	86.7%	76.9%	56.5%	92.5%	93.7%	88.1%	82.3%	75.0%
✓		✓		91.2%	88.2%	82.9%	75.5%	52.8%	91.8%	89.0%	84.8%	80.5%	72.0%
✓		✓	✓	92.6%	91.5%	86.2%	76.9%	52.3%	92.9%	92.0%	87.6%	81.2%	70.9%
✓	✓			89.8%	87.8%	81.5%	74.1%	51.9%	90.9%	89.7%	84.9%	80.8%	72.3%
✓	✓		✓	91.6%	90.1%	84.8%	73.6%	51.9%	92.1%	90.9%	86.7%	80.2%	75.9%

Table 3.6: The precision and recall rate for different combination of descriptors. Here, the precision is defined as $P\% = (\frac{TP}{TP+FP}) \times 100$,where TP is true positive, FP is false positive. The Recall (i.e. detection rate) is defined as $R\% = (\frac{TP}{TP+FN}) \times 100$, where TP is true positive and FN is false negative. In this table, all of the measures must be high for a method to show that it can provide sufficient discrimination and classification.

	Descriptor Combination				No	Partial Occlusion		
Rank	TRAJ	HOG	HOF	MBH	Occ.	10%	25%	**Avg.**
1	✓			✓	92.0%	93.4%	86.7%	**90.1%**
2	✓		✓	✓	92.5%	91.5%	86.2%	**88.9%**
3			✓	✓	91.1%	89.1%	87.7%	**88.4%**
4				✓	91.1%	89.0%	87.2%	**88.1%**
5	✓	✓	✓	✓	91.5%	90.5%	84.8%	**87.7%**
6	✓	✓		✓	91.6%	90.1%	84.9%	**87.5%**
7		✓	✓	✓	89.6%	89.5%	84.9%	**87.2%**
8		✓		✓	89.6%	88.5%	84.0%	**86.2%**
9	✓		✓		91.1%	88.4%	83.1%	**85.7%**
10	✓	✓	✓		90.7%	88.8%	82.2%	**85.5%**
11	✓	✓			89.8%	87.8%	81.6%	**84.7%**
12		✓	✓		88.3%	87.3%	81.1%	**84.2%**
13			✓		86.9%	87.3%	79.8%	**83.5%**
14	✓				87.3%	85.0%	81.7%	83.3%
15		✓			74.0%	68.6%	59.4%	64.0%

Table 3.7: The ranking is computed on the F-Score measure. The F-score is a measure of accuracy that considers precision and recall rates to compute the score as follows: $F\% = 2 \times \frac{Precision \times Recall}{Precision + Recall}$. This table shows the ordered list of descriptor combination in terms their F-Score measure in the partial occlusion case. A higher value indicates better performance

with the same trained classifier.

Table 3.6 shows the recall and precision scores for all combinations of the descriptors we evaluated. The recall is calculated for partial and heavy occlusion scenarios. In partial occlusion, MBH and its combination with other descriptors performed significantly better than other combinations. Especially the combination of TRAJ + MBH outperforms the without-occlusion case by 2%. This can be explained by the fact that occlusion also acts like a noise filtering. It increases the discriminative power of the representation. Regarding heavy occlusion, the best performance is shown with all four combinations of trajectory descriptor. It makes the trajectory descriptor particularly suitable for scenarios with large occlusions. For example, with 75% occluded area, TRAJ individually obtained 57% recall rate which is the highest score compared to any other combination where most barely reached 50%.

Regarding precision, the same trend is observed in both occlusion scenarios. The

	Descriptor Combination				No	Heavy Occlusion		
Rank	TRAJ	HOG	HOF	MBH	Occ	50%	75%	**Avg.**
1	✓				87.3%	79.2%	56.1%	**67.7%**
2	✓			✓	92.0%	76.7%	57.2%	**66.9%**
3	✓		✓	✓	92.5%	76.6%	52.7%	**64.7%**
4	✓		✓		91.1%	74.9%	52.8%	**63.8%**
5	✓	✓		✓	91.6%	73.7%	53.0%	**63.3%**
6			✓	✓	91.1%	77.0%	49.7%	**63.3%**
7	✓	✓			89.8%	74.0%	51.5%	**62.8%**
8		✓	✓	✓	89.6%	74.3%	50.8%	**62.5%**
9	✓	✓	✓	✓	91.6%	73.8%	50.2%	**62.0%**
10				✓	91.1%	72.0%	50.1%	**61.1%**
11	✓	✓	✓		90.7%	74.3%	47.8%	**61.1%**
12		✓		✓	89.6%	70.8%	48.5%	**59.7%**
13		✓	✓		88.3%	72.3%	45.3%	**58.8%**
14			✓		86.9%	68.3%	45.8%	**57.0%**
15		✓			74.0%	42.2%	22.5%	**32.3%**

Table 3.8: Here shows the F-Score based ranking in heavy occlusion case for local action descriptors and their possible combinations.

partial occlusion is predominantly handled significantly better than others when there is a combination of MBH descriptors. For heavy occlusion, TRAJ + MBH descriptors topped the precision rank.

The poorest performance is exhibited by HOG and its combination with other descriptors. In both partial and heavy occlusion cases, the HOG descriptor obtained the worst precision and recall rate. Therefore it is unsuitable to use HOG even with other occlusion tolerant features like MBH or TRAJ as it significantly decreases the performance.

3.4.3.3 Discussion

The experimental results confirm that the motion based descriptors (TRAJ, HOF and MBH) are more discriminative when recognising human actions in an occluded scene. Among the motion based descriptors, MBH and TRAJ descriptors significantly outperform other descriptors. In the partial occlusion case, MBH is the best choice, whereas the TRAJ descriptor is good for heavy occlusion. Texture or appearance based descriptors (HOG) performed poorly in the presence of

occlusion because the object's shape undergoes significant changes. We observed that combining MBH and TRAJ descriptors outperform other possible combinations in case of both partial and heavy occlusion. The performance under very heavy occlusion, in particular, is surprising. While showing a significant decrease in performance compared with no occlusion, average precision over the six actions of greater than 60% is still achieved. We speculate that this is due to the extremely simplified nature of the KTH dataset, a facet noted in a review of datasets for human action recognition [23].

3.5 Summary

This chapter provides a comprehensive evaluation based on the Bag-of-Features (BoF) representation. Many researchers have performed an evaluation of Bag-of-Features based action recognition. However, the main focus is typically on the evaluation of individual components in the framework or the framework within a limited experimental setting. In this chapter, this problem has been addressed with a comprehensive evaluation including several key elements in the pipeline, i.e., codebook dictionary construction, analysing the effect of kernel functions and the choice of local descriptors. This allows us to provide conclusion and suggestion regarding the optimal strategy and settings. Also, the performance of the local descriptors under different degrees of static occlusion was analysed. Nevertheless, the BoF-based framework still suffers the lack of spatio-temporal information. In the next chapter, we propose a graph-based approach to solving this problem.

Chapter 4

Action recognition based on a video graph

4.1 Overview

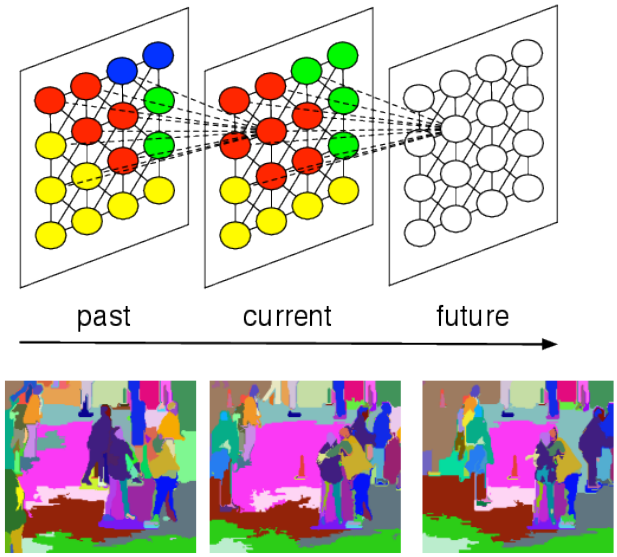
This chapter presents a new action recognition framework based on a video graph. The proposed approach aims to overcome the underlying limitation of the Bag-of-Features (BoF) model by introducing a video graph representation and a graph processing recognition pipeline. The video graph efficiently captures the spatial and temporal relationship between the local features while maintaining its discriminative power. We formulate action recognition as an energy minimization problem on the constructed video graph and perform the graph-cut optimisation to identify the action class for the corresponding video. The performance of the proposed approach is investigated using datasets of varying complexity and benchmarked against the state-of-the-art.

4.2 Introduction

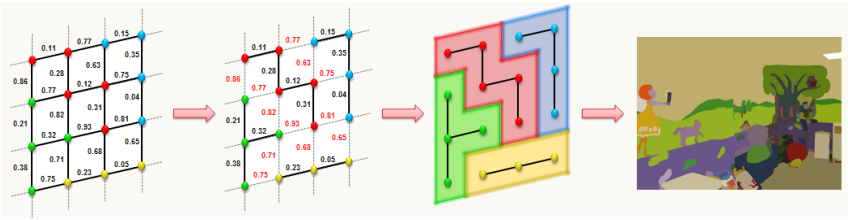
An important question in action recognition is how to efficiently and effectively represent a video scene while maintaining the discriminative appearance, motion and contextual cues of the scene. In recent years, Bag-of-Features (BoF) based methods [101] [50] [32] [56] have demonstrated excellent results in action recognition. However, as identified in a number of works [114] [90] [67], such

approaches typically ignore the spatial and temporal cue of the local features in the recognition process, limiting fine-grained analysis of the video. Many authors noted [74] [24] that capturing the spatiotemporal patterns in an action recognition framework can improve system performance. In this context, this chapter introduces an action recognition framework based on graph-structured local features to explicitly exploit their spatial and temporal connections. The graph representation is a simple and powerful method for a modelling interaction between a system's individual elements. It is a well-established data abstraction approach in computer vision, in particular for 2D image understanding, due to its intuitive representation of capturing the underlying image spatial structure. For instance, in many successful object segmentation methods [15] [16], they interpreted the image parts as the graph nodes and their pairwise spatial relationship in terms of edges (See Figure 4.1). Recently, this representation has attracted researchers [25] [74] [95] from the action recognition community to focus on adding structural cues into effective video representation. Many of them have exploited the graph representation to capture the spatial-temporal relationship between the local features extracted from video data and thus increase the discriminative power of recognition.

This chapter presents two contributions. First, we propose to extend the popular *dbscan* clustering algorithm [35] towards a graph-based video representation. In this video graph, each cluster c_i of local features forms a node u_i and its connectivity (edge) is determined by proximity in space and time (see Figure 4.2). Each node, v_i, is associated with a discriminative score indicating the degree of support for an action class based on the corresponding set of a local descriptor. Second, this chapter explores the application of the graph-cut optimisation method from 2D image segmentation to 3D spatio-temporal volume analysis to investigate its effectiveness for action recognition in video. Graph-cut based methods have achieved impressive performance for object segmentation, even on difficult image datasets [27]. It is interesting to study how successful



(a) The video segmentation process that utilizes the graph structure to impose the spatio-temporal relationship between the video parts [38].



(b) The image is interpreted as a graph for the object segmentation problem [121].

Figure 4.1: The application of graph structure in computer vision problems

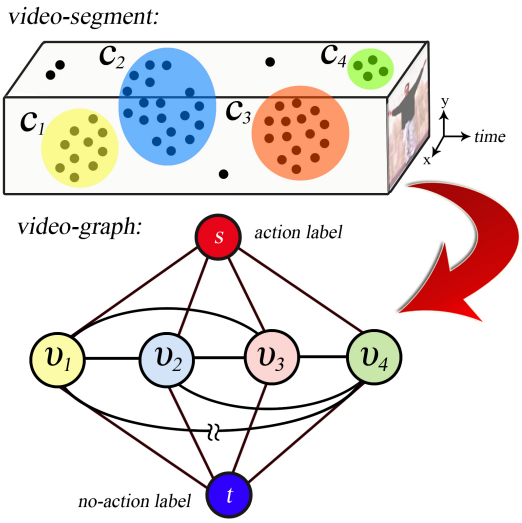


Figure 4.2: Action recognition is formulated as a graph cut optimization problem over a constructed video graph.

approaches could be extended to the action recognition problem. The proposed approach has several important properties. First, the method accommodates a variety of features and classifiers, making it flexible as a general action recognition tool. To illustrate, we have used four descriptors effectively for action classification. Second, as Chen et al [24] highlighted, the graphical representation is equivalent to that of an exhaustive sliding window search, yet requires orders of magnitude less search time. Finally, the graph-based representation is sufficiently generic that one can directly apply any graphical probabilistic inference methods to gain insight into the video data.

The remainder of the chapter is structured as follows - Section 4.3 describes the action recognition framework; Section 4.4 presents an extension to the *dbscan* algorithm and video graph construction and Section 4.5 introduces an action recognition process based on a video graph followed by the result (Section 4.6) and the chapter summary (Section 4.7).

4.3 Framework Overview

In the proposed recognition framework (Figure 4.3), the training stage adopted a Bag-of-Features (BOF) model to train a classifier model that is used to assign discriminative node score during the graph construction. For graph construction, first, the local feature extraction is performed while maintaining its spatio-temporal location information. Next, based on spatio-temporal location and feature similarity, a video graph is constructed. In this video graph, each node indicates a set of local features and the associated discriminative score is assigned by a learned model during the training stage. Finally, graph-cut optimisation is performed to solve the optimum labelling problem by minimising the objective energy function associated with the video graph. We assign the most frequent node label in the graph as a class label for the corresponding video.

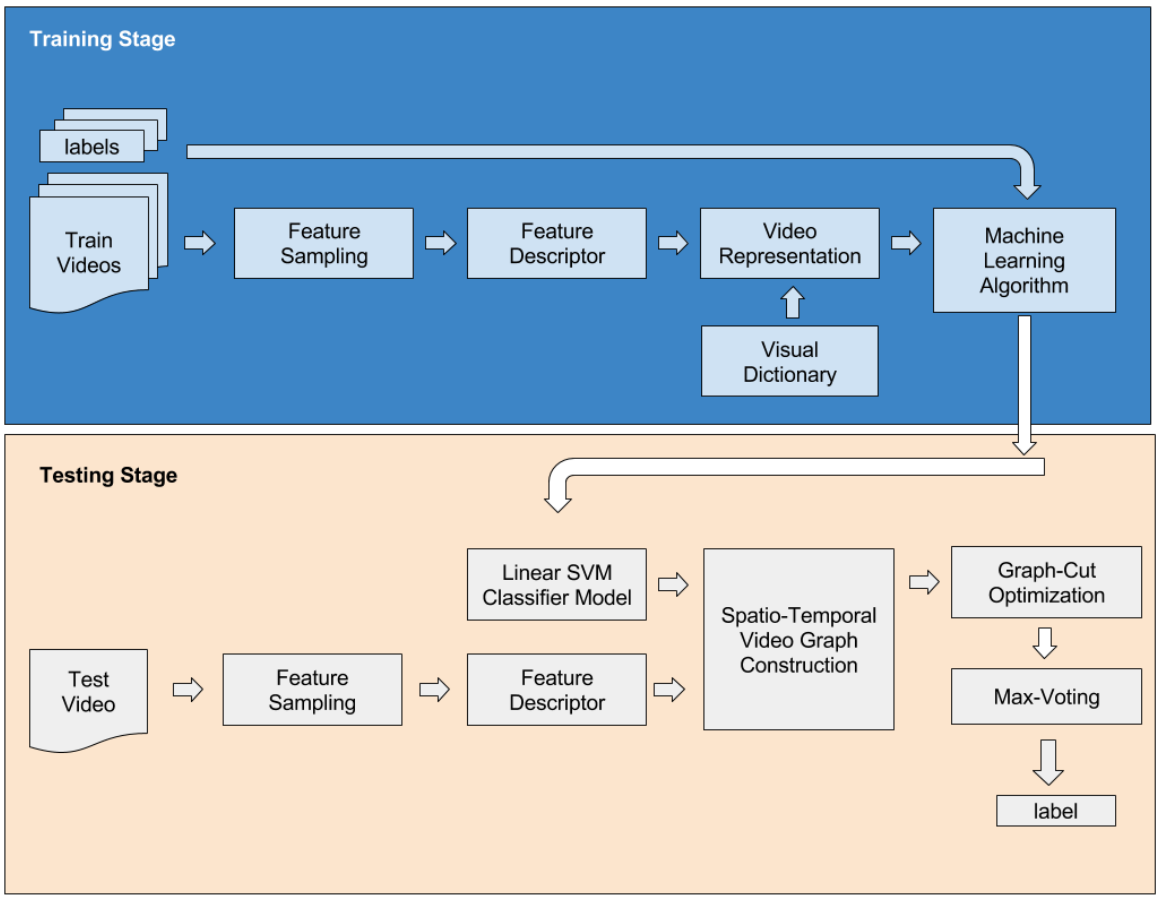


Figure 4.3: The recognition pipeline of our proposed system. During training, the standard BOF model is applied. However, at prediction stage, we have exploited the graph structure to embed spatio-temporal cue for video representation which is a major weakness in BOF model.

4.4 Video Graph Construction

This section describes how to construct a video graph for a video sequence. In the case of a 2D image, the graph construction can be straightforward by considering individual pixels into a node. In practice, the image segmentation algorithm [15] typically use this strategy to interpret an image in the graph representation. However, let us say that the input is 360×240 dimensional video with 100 frames, the total number of nodes in the resulting graph will exceed 8 million. The amount of memory space and the computational power required for processing such a graph results in slower processing speed or it requires special purpose processing tools and hardware such as graphical processing unit (GPU). The graph construction technique should produce a sparse as well as a discriminative graph structure. We propose to exploit the popular dense clustering method (dbscan) for the development of the video graph.

4.4.1 Local features

The proposed technique can be used with any spatio-temporal local features. However, we adopt the same approach described in the previous chapter. The feature extractor (DT) is used extract volumetric motion trajectories. The size of the volume is $N \times N$ pixels and L frames, with $N = 32$ and $L = 15$ used in the experiments. For each trajectory volume, the different descriptor is calculated to capture the local video region properties. Similar to the baseline framework (Chapter 3), we compute HOG and HOF [57] to capture the local appearance and motion around the trajectories. Also, MBH [30] and TD [101] are computed in order to represent the relative motion and trajectory shape. The feature descriptor dimensions of HOG, HOF, MBH and TD are respectively 96, 108, 192 and 30.

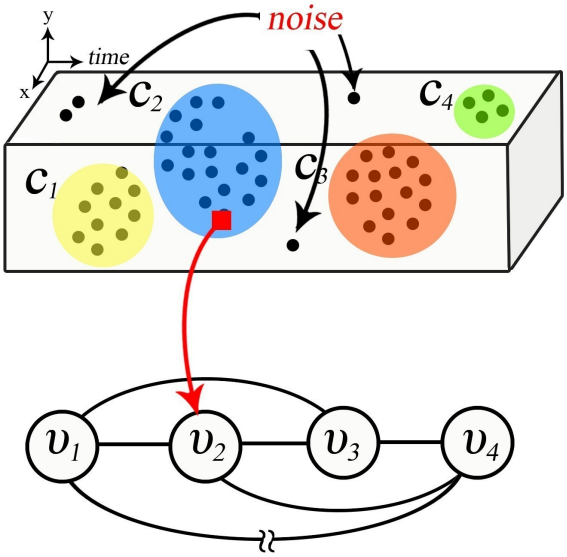


Figure 4.4: The extended dbscan algorithm does not suffer from restriction of a pre-defined grid boundary and the resulting graph is sparse and intuitive.

4.4.2 Video Graph Construction

This section describes the video graph $G(V, E)$ for the input video sequence, where V is a set of nodes and E is a set of edges. Each node in the graph is the abstraction of a set of local features extracted within a spatio-temporal neighbourhood. The smallest possible node is a single feature point, and the largest possible one would be the full test sequence, i.e., all features from all frames. The factors to be considered for choosing the scale is the granularity of detection and the computational complexity. Note that nodes with a larger number of feature points are favourable not only for computational efficiency, but also their aggregated descriptor statistics have better discriminative power. The node structure and edge formation strategy are as follows:

- ***Node Structure***: For constructing the graph node, we propose a feature-point clustering method inspired by the density-based clustering method, particularly the *dbscan* algorithm. The density-based clustering approach does not require one to specify the number of clusters in the data as a prior and can find arbitrarily shaped clusters by tuning only two parameters, a maximum search radius ϵ and the minimum number of points

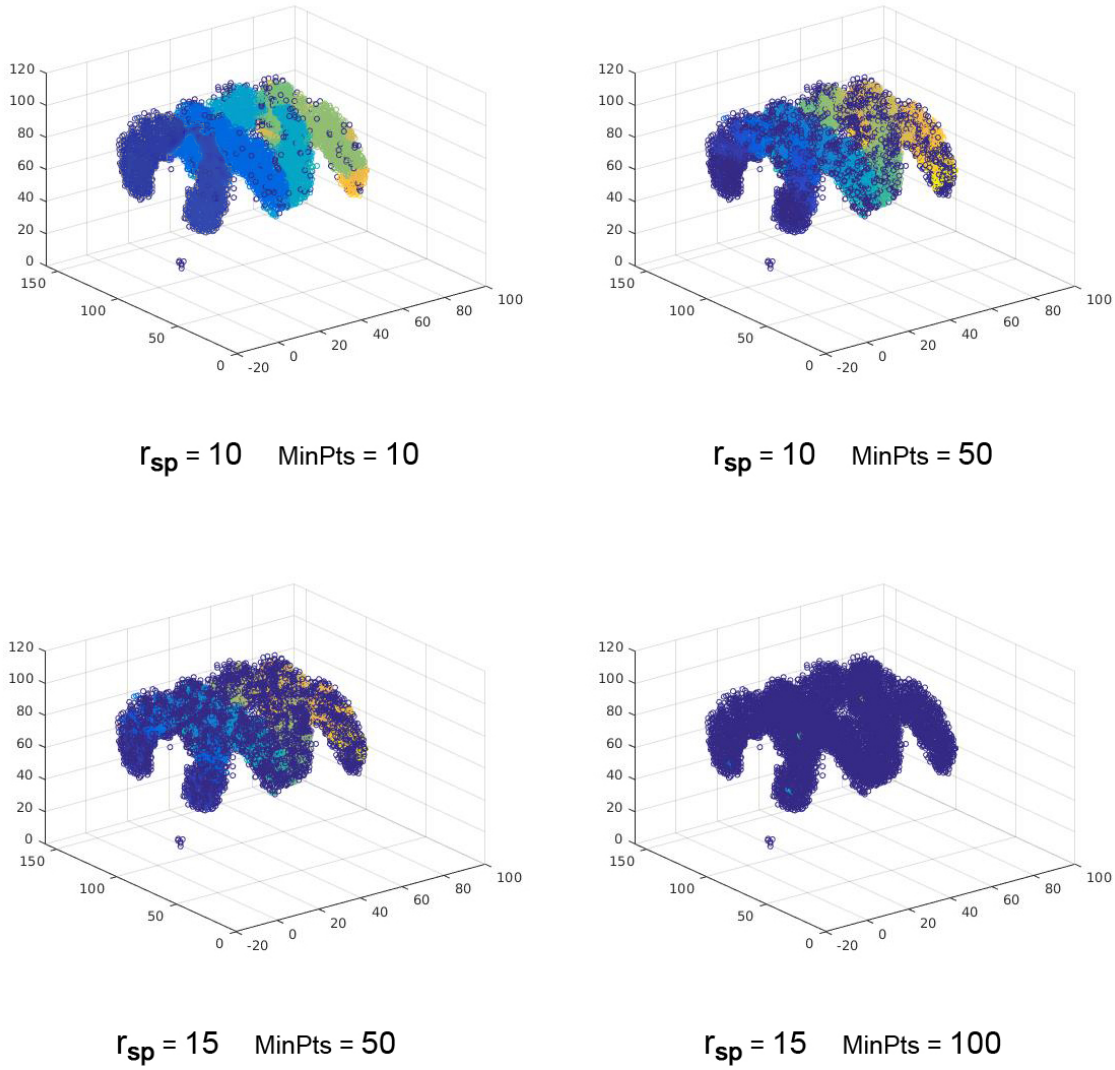


Figure 4.5: The node structure variation with varying parameter values of spatial radius and minPts where the dark blue colour represents the disconnected node (noise).

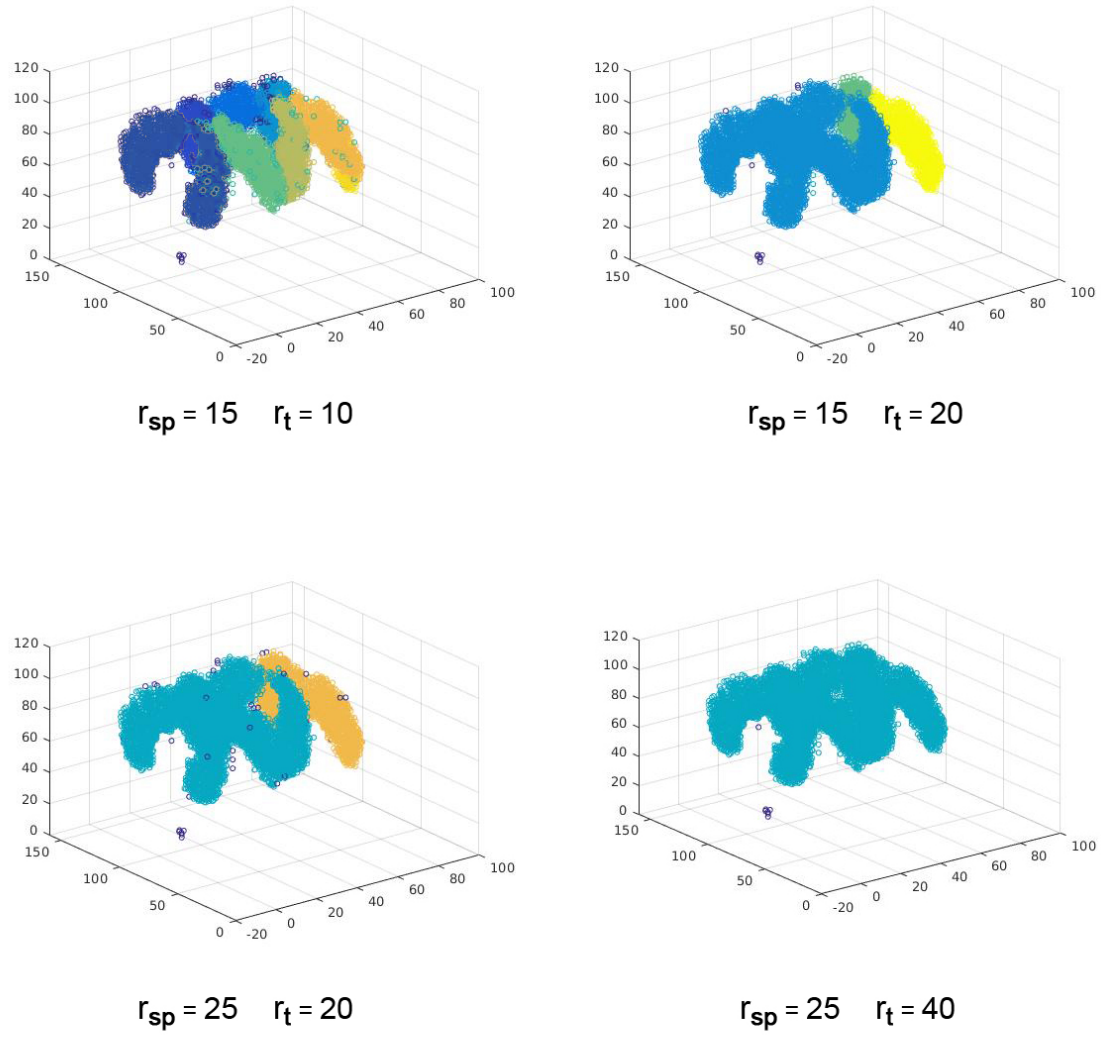


Figure 4.6: The node structure variation with varying parameter values of spatial radius and temporal radius. Each colour represents a different cluster group.

$minPts$. As shown in Figure 4.4, the algorithm groups only feature points that are densely inter-located. If the density of the feature point's spatio-temporal neighbourhood is less than the threshold value, $minPts$, such a feature point is considered as noise and does not contribute towards an action class. Sometimes, the feature sampling technique or the context of video may result in densely distributed local features. The *dbscan* algorithm can not handle dense data points. It merges each data point to produce a single giant cluster, which is not ideal.

Therefore we extended the *dbscan* algorithm designed for clustering the feature points to take into account not only their location $[x, y, t]$ but also the local descriptor characteristic. In addition, the maximum search radius parameter, ϵ is split into two components: spatial radius r_{sp} and temporal radius t_{tmp}. This allows us to reduce the pairwise distance calculation space by bounding using t_{tmp} radius and independently treating spatial and temporal dimensions, respectively measured in $pixels$ and $video\text{-}frames$. To account for the trajectory shape, we calculate a lower dimensional *"trajectory code"* ($vCat$) over randomly sampled trajectory descriptors. The trajectory descriptor is a sequence of displacement vectors, for scale invariance, scaled by the sum of the magnitudes. Therefore, the extended *dbscan* algorithm operates on 4-dimensional data points, $(x, y, t, vCat)$, where x, y, t is a mean coordinate of the extracted trajectory and $vCat$ is the nearest codeword associated with this trajectory. A cluster is formed if its neighborhood contains enough points ($minPts$) with the same *"trajectory code"* ($vCat \in \{1, .., 64\}$). This ensures similar trajectories between the feature points. This technique may occasionally produce a disconnected node if the local feature density within the spatio-temporal neigbhourhood is less than the threshold $minPts$. It could create problem by resulting in a significant number of disconnected nodes if the parameter value for $minPts$ is not set properly (See Figure 4.5).

- ***Linking Strategy***: The connectivity between nodes also affects both the shape of the graph structure and the cost of graph-cut optimisation. We adopt a distance based linking strategy for connecting the constructed nodes from the extended *dbscan* clustering algorithm. We have chosen the distance-based strategy as it is a simple and effective measure that is able to add spatio-temporal proximity among the constructed nodes. Additionally, the edge has no direction as we use graph cut optimisation that operates only on non-directed graphs. The connectivity between nodes v_i and v_j is determined by the distance between their corresponding group g_i and g_j. If the resulting node corresponds to hla non-spherical 3D region in the video, the group centre is calculated as the mean coordinate of the local features in the region. In the experiment, we use Euclidean distance measure. However, any distance metric can be utilized such as *minimum distance* which is more suitable for non-spherical nodes. For example, if the distance between g_1 and g_2 is greater than a pre-defined threshold value, then an edge between node v_1 and v_2 will not be formed.

4.5 Recognition problem formulation over the video graph

Given a video sequence represented as a graph of clustered feature nodes, we now seek to determine regions where there is significant label agreement. The 3D graph cut algorithm solves the labeling problem by minimizing the following energy function defined using a video graph G:

$$\mathcal{E}(L) = \sum_{r \in V} -\mathcal{E}_1(l_r) + \lambda \sum_{(r,s) \in V} -\mathcal{E}_2(l_r, l_s) \tag{4.1}$$

where l_r is the action label of node r, and $L = (l_r : \forall r)$. The first term $\mathcal{E}_1$ (likelihood) measures the conformity of the local features extracted in the region

Algorithm 1 The pseudo code for our extended *dbscan* algorithm. It clusters local features based on their spatio-temporal location.

D, R_{sp}, R_{tmp}, $MinPts$, $vCat$ Cluster $\mathcal{G}$ for D data points

Initialization

for each unvisited point P in dataset D **do**
  mark P as visited
  $NeighPts$ = REGIONQUERY(P, R_{sp}, R_{tmp}, $vCat$)
  sizeof($NeighPts$) < $MinPts$
  mark P as NOISE
  C = next cluster
  EXPANDCLUSTER(P, $NeighPts$, C, R_{sp}, R_{tmp}, $MinPts$, $vCat$)
  function EXPANDCLUSTER(P, $NeighPts$, C, R_{sp}, R_{tmp}, $MinPts$, $vCat$)
    mark P as visited

    for each point P' in $NeighPts$ **do**

      if P' is not visited **then**
        mark P' as visited
        $NeighPts'$ = REGIONQUERY(P', R_{sp}, R_{tmp}, $vCat$)

        if SIZEOF($NeighPts'$) > $MinPts$ **then**
          $NeighPts = NeighPts \cup NeighPts'$

          if P' is not yet member of any cluster **then**
            add P' to cluster C

    function REGIONQUERY($P, Rsp, Rtmp, vCat$)
      return all points within P's temporal, R_{tmp}, and spatial, R_{sp}, neighborhood, with the same visual category $vCat(P)$
    end function

r to the action class label. The second term $\mathcal{E}_2$ measures the agreement between two adjacent nodes. An important property of this formulation is that it can be theoretically justified regarding maximum posterior estimation of a Markov Random Field (MRF).

4.5.1 Energy Minimization Methods

In the domain of 2D image understanding, energy minimization approaches have had a renaissance, due to its simple and powerful representation of structural data and availability of effective optimisation algorithms such as graph cuts [15] and Loopy Belief Propagation (LBP). For instance in stereo vision research, according to the widely-used Middlebury stereo matching benchmarks, the majority of top-performing methods rely on energy minimization approaches. In addition, the range of applications of labelling problems has successfully expanded, starting from early applications such as image restoration [31], texture modelling [54], image labelling [16] and interactive object segmentation [15].

This family of methods aims to assign to each site a label which can be any quantity (e.g disparity, object and foreground/background) in such a way as to minimise the objective energy function $\mathcal{E}$. The energy function $\mathcal{E}$, which can also be viewed as the log of the posterior probability of a Markov Random Field that consists of a data energy $\mathcal{E}_d$ and smoothness energy $\mathcal{E}_s$. In the MRF framework, energy term $\mathcal{E}_d$ comes from the negative log likelihood of the measurement noise and $\mathcal{E}_s$ is related to the negative log of the prior probability.

There are a number of optimisation algorithms; that finds an exact or approximate solution of the energy function $\mathcal{E}$, such as ICM, Graph Cuts, LBP and ICM. Relatively little attention paid to the relative performance of the various optimisation algorithm. Among the few survey papers [92], Szeliski at el [91] have performed a comprehensive evaluation and highlighted Graph Cuts and LBP regarding accuracy and efficiency among other methods. In this chapter, we have used Graph-Cut based optimisation in our experimental work.

4.5.1.1 Graph Cuts Optimization

Energy minimization problems can be reduced to instances of the maximum flow problem in the graph. Graph cut based algorithms are known to rapidly compute a local minimum. We have used the Graph Cuts algorithm proposed by Boykov et al [14] and the implementation is available online[1]. This method efficiently calculates the approximate solution in the multi-label scenario.

The Graph Cuts algorithm finds a local minimum by making local improvements. The most popular graph cuts algorithms are called "swap-move" and "expansion-move". The "swap" algorithm performs local improvements by selecting two possible states of α and β, then finds those nodes, labelled as α, that must be changed to β or vice-versa in order to minimise the total energy. However, based on the min-cut and max-flow technique, the optimal swap for the whole graph can be efficiently calculated. The "expansion move" algorithm obtains a local minimum state when there is no expansion move further improves the total energy level. For expansion moves, the criteria for a local minimum is strong compared to standard moves that results in less minima.

4.5.2 Estimation of Likelihood ($\mathcal{E}_1$) and Prior ($\mathcal{E}_2$)

To measure node likelihood, the discriminative classifier should satisfy two properties. First, it must be able to score an arbitrarily shaped set of feature points. Second, it must be defined such that features computed within local space-time regions can be combined additively to obtain the cumulative classification for a larger region. Suitable additive classifiers include linear support vector machines (SVM), boosted classifiers, or Naive Bayes classifiers. In our experiments, we use a linear SVM with histograms (bags) of quantized space-time descriptors. We consider BoF's computed over several types of local descriptors discussed in Section 3.1.

We compute a vocabulary of K visual words by quantizing a subset of randomly

[1]http://vision.ucla.edu/ brian/gcmex.html

sampled features from the training videos. A training video subvolume with N local features is initially described by the set $\mathcal{S} = \{(\boldsymbol{x}_i, v_i)\}_{i=1}^{N}$, where each $\boldsymbol{x}_i = (x_i, y_i, t_i)$ refers to the 3D feature position in space and time, and v_i is the associated local descriptor. Then the volume is converted to a K-dimensional BoW histogram $h(\mathcal{S})$ by mapping each v_i to its respective visual word c_i ,and tallying the word counts over all N features.

We use the training instances to learn a linear SVM for each action label, which means the resulting scoring function has the form: $f(S) = \beta + \sum_i \alpha_i < h(\mathcal{S}), h(\mathcal{S}_i) >$ where i indexes the training examples, and α, β denote the learned weights and bias. This can be rewritten as a sum over the contributions of each feature. Let $h^j(\mathcal{S})$ denote the j-th bin count for histogram $h(\mathcal{S})$. The j-th word is associated with a weight $w^j = \sum_i \alpha h^j(\mathcal{S}_i)$, for $j = 1, ..., K$. Thus the classifier response for any subvolume $\mathcal{S}$ is:

$$f(\mathcal{S}) = \beta + \sum_{j=1}^{K} w^j h^j(\mathcal{S}) = \beta + \sum_{i=1}^{N} w^{c_i} \tag{4.2}$$

where c_i is the index of the visual word that feature v_i maps to, $c_i \in [1, K]$. By writing the score of a video region as the sum of its N features' "word weights", we now have a way to associate each local descriptor occurrence with a single weight based on its contribution to the classifier score.

This same property of linear SVMs is used in [24] to enable efficient subgraph search for action detection.

Likelihood, $\mathcal{E}_1$, is defined as:

$$\mathcal{E}_1(l_r) = \sum_{\boldsymbol{x}_j \in r} w^{c_j} \tag{4.3}$$

where $\boldsymbol{x}_j$ is the 3D coordinate of the j-th local descriptor falling within node $r \in V$, and c_j is its quantized feature index. Note that $\boldsymbol{x}_j$ is the feature point position of the low-level descriptors. Consequently, nodes with high positive weights indicate that the action presence in the video region where node corresponding while

negative value means the absence.

Prior energy, $\mathcal{E}_2$, simply measures the label agreement between adjacent nodes, defined as:

$$\mathcal{E}_2(l_r, l_s) = \begin{cases} 1, & \text{if } l_r = l_s \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

The prior energy definition can be extended to include additional information such as the similarity between different action classes.

The objective function of Equation (1) can be globally minimised by an efficient graph cut algorithm [15] and the resulting most frequent labels over graph nodes will determine an action label for the test sequence, that can be found by a simple voting strategy. The default parameter is empirically fixed to $\lambda = 0.85$ in all our experiments. The λ value is calculated based the validation set of the KTH dataset.

4.6 Experimental Evaluation

This section presents the evaluation of the proposed approach. For the purpose of comparative analysis, the identical training stage with the baseline approach is adopted. The evaluation framework uses the same local descriptors (i.e., TRAJ, HOG, HOF and MBH) investigated previously (Chapter 3). In the experiment, the co-occurrence BoF histograms of the above descriptors are concatenated to form a single descriptor vector. The linear SVM model is learned using the combined descriptors. For the recognition stage, first, the video graph is constructed where each node's discriminative score is assigned by a SVM learned model. Then graph-cut optimisation is performed to solve the optimum labelling problem by minimising the objective energy function defined in Section 4.5. Finally, the most frequent node label in the graph is chosen as a class label for the corresponding video.

The experimental work is split into two sections. Section 4.6.1 presents the comparative analysis between the baseline and the proposed approach. In this

Action class	BoF baseline	Proposed	Difference
brush_hair	**66.7%**	53.6%	**-13%**
catch	**20.0%**	11.7%	-8%
clap	23.1%	**31.5%**	8%
climb_stairs	**66.7%**	61.2%	-6%
golf	83.3%	**91.5%**	8%
jump	0.0%	0.0%	
kick_ball	**20.0%**	2.9%	**-17%**
pick	8.3%	21.4%	13%
pour	68.8%	75.9%	7%
pullup	68.8%	**85.5%**	**17%**
push	50.0%	**71.5%**	**22%**
run	18.2%	**26.3%**	8%
shoot_ball	11.1%	35.2%	**24%**
shoot_bow	26.7%	39.8%	13%
shoot_gun	12.5%	24.3%	12%
sit	60.0%	**81.4%**	**21%**
stand	45.5%	26.2%	**-19%**
swing_baseball	20.0%	**76.0%**	**56%**
throw	0.0%	**12.5%**	13%
walk	**75.0%**	56.1%	**-19%**
wave	8.3%	**15.8%**	8%
MAP	**35.9%**	**42.9%**	

Table 4.1: The performance by action class for J-HMDB dataset

context, we perform the experiment with the J-HMDB dataset. This dataset is chosen since (i) the dataset contains a large number of different action categories (a total of 21 classes) and (ii) the setting where the video sequence is captured is a good representation of an *uncontrolled* environment. Also, the dataset is provided with meta-data information that enables to analyse the effect of different environment factors. In the section 4.6.2, we present the evaluation of the proposed approach using several datasets and a comparative analysis to the respective state-of-the-art.

Action class	Camera motion	Video quality	Camera view-point				Visible body part			
			back	front	left	right	full	head	lower	upper
brush_hair	21%	98%	41%	41%	15%	2%		10%		90%
catch	76%	45%	19%	33%	25%	23%	75%			25%
clap	32%	51%		93%	2%	5%	7%			93%
climb_stairs	79%	49%	75%	20%		5%	68%		33%	
golf	59%	68%		19%	7%	74%	100%			
jump	59%	5%	15%	26%	31%	28%	100%			
kick_ball	45%	44%	67%	22%	6%	6%	89%		8%	3%
pick	73%	53%	15%	33%	23%	30%	90%		3%	8%
pour	32%	65%		89%		11%				100%
pullup	56%	60%	51%	36%	2%	11%	60%			40%
push	89%	12%	21%	10%	29%	40%	95%			5%
run	66%	50%	35%	28%	20%	18%	93%		3%	5%
shoot_ball	30%	50%	80%		8%	13%	65%			35%
shoot_bow	9%	50%	9%	85%		6%	17%			83%
shoot_gun	32%	54%	13%	13%	27%	47%	27%			73%
sit	61%	54%		51%	28%	21%	64%			36%
stand	67%	43%		64%	19%	17%	61%			39%
swing_baseball	44%	40%	17%	81%	2%		81%			19%
throw	24%	48%	57%	11%	17%	15%	37%			63%
walk	72%	49%	32%	39%	10%	20%	80%			20%
wave	46%	55%	2%	86%	2%	10%	19%	7%		74%

Table 4.2: The meta-label statistics by action class for the J-HMDB (Split 1) dataset. The column "Camera Motion" represents the percentage of the videos for a certain class ,e.g golf, that has the camera motion whereas the column "video quality" represents the average video quality where the good quality being '1'. The column "camera view-point" represents the percentage of the video by shot camera angle likewise the "Visible body part" column by four type.

Variables	BoF baseline	Proposed
Camera view-point variation	-0.06	-0.22
Camera motion presence	-0.19	-0.07
Visible body part variation	-0.05	-0.02
Video quality	0.42	0.30
Action region size	0.13	0.08

Table 4.3: The environment variable correlation with the classification performance

4.6.1 Comparative analysis result with the baseline

4.6.1.1 J-HMDB dataset

Table4.1 shows the result on the J-HMDB dataset. The result is shown with performance metric of average precision (AP). The metric is recommended by the dataset author, and thus we adopt this metric. In terms of average performance over all action classes, the proposed approach (42.9%) significantly outperforms the baseline (35.9%) by 7 percent in MAP (mean AP). Interestingly, as you can notice in the 4th column of the same table, the performance increment is not uniform and varies highly among the action classes. In particular, we found several classes i.e. *brush_hair*, *kick_ball*, *stand* and *walk* for which the baseline is outperforming. To explain the result in a justified manner, we perform a statistical analysis on the meta data provided with the dataset with an assumption that environmental factors might have influenced on the performance. Table 6.2 shows each class and the environmental settings where the corresponding videos are captured. We have selected "camera Motion", "video quality", "camera view-point" and "visible body part" factors. The "camera motion" represents the percentage of the videos for a certain class that have camera motion whereas the "video quality" represents the average video quality where the good quality is '1' , medium '0.5' and bad '0'. The "camera view-point" represents the percentage of the videos shot from different camera positions (*back, front, left, right*). The "visible body part" represents the percentage of the videos that have a certain type of visibility (*full, head, lower, upper*) of an actor/action. Let us examine the top performing classes from each method. For instance, the action class *brush_hair* has an excellent result with the baseline. The class has only 28% videos where the camera motion is present and furthermore 90% of the videos dominated by a specific body part (upper body). In contrast, the action class *push* has obtained 22% improvement by the proposed approach in comparison with the baseline. However, the portion of the corresponding videos that have camera motion present

Methods	(Avg.Acc)
Laptev et al. [57]	91.8%
Kovashka et al. [51]	94.5%
Gilbert et al. [39]	95.7%
Le et al. [62]	93.9%
Wang et al. [101]	94.2%
Baseline	94.0%
Proposed	98.1%

Table 4.4: The state-of-art comparison for KTH dataset

is significantly higher (89 %) compared to *brush_hair*. We can test hypothesis that the baseline (BoF) is suitable for the relatively simple scene and environment, i.e.,. fixed camera, fewer variations in the scene. The proposed approach shows robustness under the different variations in the scene. To generalise the claim, the statistical correlation between the environmental variable and the performance of the baseline and the proposed approach is calculated (Table 4.3). As expected, the variations in the view-point, visible body part and the camera motion have negative correlation with the performance of both approaches. In particular, the camera motion and body part variation have a higher influence on the baseline. However, the camera view-point variation has a significant impact on the proposed approach. We hypothesis that this is due to the smoothness assumption of the proposed approach as it builds the spatio-temporal connection between the video frames. The video quality and action region size has the positive correlation with the performance. In particular, unexpectedly, the video quality has the highest impact on the performance among all factors for both approaches.

4.6.1.2 KTH dataset

4.6.2 Comparison to the state of the art

The KTH dataset [13] is the most popular dataset used in the evaluation of action recognition methodologies. We follow the original experimental setup of the dataset publisher [13]. The average accuracy is a commonly accepted performance

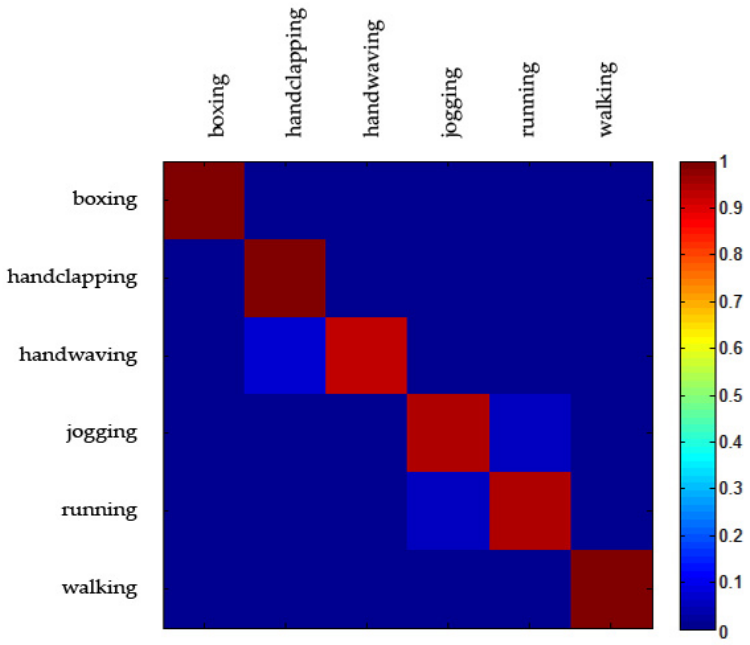


Figure 4.7: The confusion matrix for the KTH dataset.

measurement for this dataset. Table 4.4 shows the comparison of methods applied to the KTH dataset. It is observed that the proposed approach obtains an excellent result (98.1 %) that improves the current state of the art. Figure 4.7 shows the confusion matrix of the action classification. The *boxing, handclapping, walking* classes have an excellent accuracy of 1 whereas *handwaving, jogging, running* have minor miss-classification cases.

4.6.2.1 HOHA dataset

The HOHA dataset contains 430 videos with eight different actions. This dataset is extremely challenging due to significant camera motion, rapid scene changes and occasionally significant clutter. We followed the experimental setting previously proposed in [55]. As compared with the state-of-art methods in HOHA dataset (Table 4.5), our method is less accurate with MAP (mean average precision) of 35.2 %. This can be attributed to the use of the simple linear SVM classifier in our method, while the latter methods [114] [74] use flexible learning techniques such as multi-instance based learning and non-linear kernel methods. Also, our

Methods	MAP
Raptis et al. [18]	40.1 %
Yeffet et al. [36]	36.8 %
Laptev et al. [81]	38.4 %
Matikanien et al. [6]	22.8 %
Shandong et al. [103]	47.6 %
BoF Baseline	36.3 %
Proposed	35.2 %

Table 4.5: The state-of-art comparison for HOHA dataset

classifier is learned from the training set where only the temporal extent of the depicted action is unknown. For instance, Raptis et al [74] work, that is similar to our approach, performed with MAP of 40.1 % however, the authors used the manual annotated spatio-temporal bounding box for each training sequence to learn the model. Interestingly, the baseline outperforms the proposed approach. We hypothesise that the this is due: (i) long duration and significant motion boundaries associated with the HOHA compared to the other datasets (ii) the prior assumption in the graph model that enforces the adjacent nodes to have the same class label might have a negative impact on these shot boundary regions.

4.6.2.2 UCF-Sports dataset

For the UCF-Sports dataset, we used the experimental protocol proposed by Lan et al [55]. The dataset is split into 103 training and 47 test samples. The performance confusion matrix is shown in Figure 4.8. As one can see in Table 4.6, our method improves the state-of-art performance by 5% in terms of average accuracy. We associate this good performance with the following points. First, the characteristic of the UCF-Sports dataset is rather simple compared to the HOHA, and the average action duration is relatively short. Thus, this facilitates learning a cleaner classifier (noise free). Secondly, we believe the graph-structure has a significant impact on the system's performance, given that we notice a significant improvement over the baseline performance (the same classifier applied for the

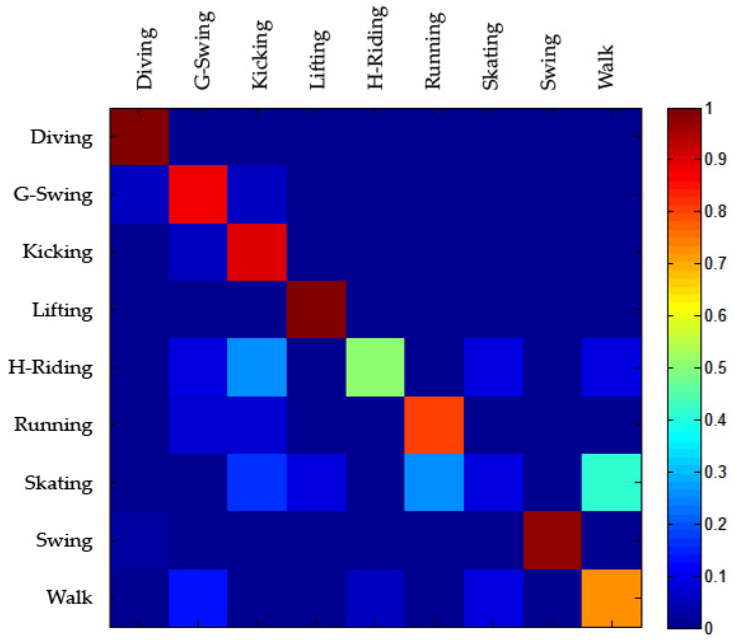


Figure 4.8: The confusion matrix of the proposed approach for the UCF dataset.

test set using BoF representation).

4.6.2.3 Localisation results on the J-HMDB dataset

This section presents the localisation results on the J-HMDB dataset. The dataset contains about 900 videos of 21 different actions and the mAP (mean average precision) metric is used in the evaluation. The IOU (intersection over union) parameter is set to $\sigma = 50\%$ which is used by other methods [111] [40]. We have used the coordinates associated with the nodes to determine the location of the

Methods	(Avg.Acc)
Ma et al. [65]	81.7 %
Lan at al. [55]	73.1 %
SDPM [37]	75.2 %
Xu et al. [117]	78.8 %
Raptis et al. [119]	79.4 %
BoF Baseline	73.5 %
Proposed	86.7 %

Table 4.6: The state-of-art comparison for UCF-Sports dataset

J-HMDB ($\sigma = 0.5$)	
Action Tube [40]	53.3 %
STMH [111]	60.7 %
Our method	22.1 %

Table 4.7: The state-of-art comparison for J-HMDB localisation dataset

recognized action in space and time. Table 4.7 shows the performance result. The application of the proposed method in the localisation problem showed poor performance in comparison to the localisation-dedicated state-of-the-art algorithms. However, our approach only relied on the local descriptor without using a high-level descriptor such as regional [111] and object-level descriptors [24].

4.6.3 Discussion

This section presents a discussion of the experimental evaluation performed in Section 4.6.1 and 4.6.2. The experimental works have shown promising result across the various datasets. Also, a statistical analysis is performed to investigate the robustness under the different environmental factors. It is found that the approach effectively deals with the various environmental changes i.e. camera motion, the visible body variation as long as there are smooth changes in the scene. However, in a scenario where there is a discontinuity in the spatio-temporal relationship (shot boundaries, view-point variation changes, etc.), the video graph model fails to capture the scene accurately and consequently it causes a drop in the performance.

4.7 Summary

This chapter proposes an action recognition framework based on a video graph. For the video graph construction, we introduced an effective strategy based on the extension of the density clustering method *dbscan*. In addition, a new action recognition formulation in terms of an energy function that operates on the nodes

and edges of the video graph node is presented. For the energy optimisation, we apply the effective graph-cut technique in the application of action classification problem. The experimental results prove the effectiveness of the proposed approach. The approach consistently outperforms the Bag-of-Features (BoF) baseline and achieved state-of-the-art result with KTH and UCF-Sports datasets. In the next chapter, we focus on the graph construction technique and perform the comprehensive evaluation.

Chapter 5

Evaluation of different video graph construction techniques

5.1 Overview

The previous chapter introduced an action recognition framework that operates on a graph representation of video data. In practice, there are numerous ways to obtain a graph-based representation of a video volume that encodes its spatial-temporal relationship among the local regions of a video. This chapter introduces the various techniques that can be used for this process and investigates the impact of the video graph construction strategies on action recognition. We examine three techniques as representatives of the identified taxonomy of approaches and perform a comparative analysis among them. The experimental setup is designed to evaluate such techniques in terms of recognition accuracy, parameter sensitivity and computational complexity.

5.2 Introduction

The graph representation is a well-established data abstraction approach used in many computer vision algorithms. Examples include image segmentation [15], shadow/light detection [122] and object detection [100]. Recently, it has attracted many researchers [25] [74] [95] from the action recognition domain due to (i) the availability of a wide range of sophisticated graphical inference and learning tools

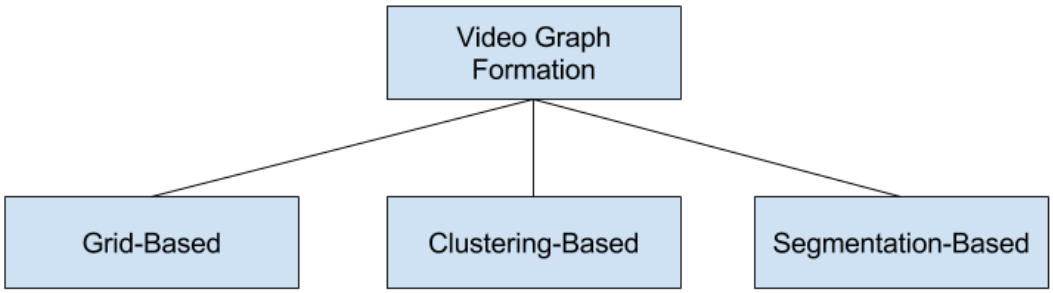


Figure 5.1: A possible taxomony of the different graph construction techniques for video-data representation

(ii) its intuitive representation for capturing the underlying spatial and temporal patterns among the local regions of the video or the local features. The typical processing pipeline for this class of approaches consists of graph interpretation of a video volume followed by a model building step that involves learning or inference. The majority of the related work focuses on building a model that is well suited for a specific graph or not tested with diverse variations in the graph structure. To the best of our knowledge, there is no work in action recognition that studies the impact of the video graph construction process. However, it is an important question as to what degree the performance of recognition system depends on structural variations of video graph representation. In this context, this chapter investigates the different video graph construction techniques and how they affect the overall recognition performance in terms of parameter sensitivity and computational complexity. In particular, we perform a comparative analysis on three graph construction techniques, namely *fixed-grid* (FG), *adaptive-region* (AR) and *supervoxel-based* (SV). The rest of the chapter is organised as follows. Section 5.3 presents the detailed description of the evaluated graph construction techniques, and Section 5.4 presents the comparative analysis results regarding their impact on the recognition accuracy, parameter sensitivity and computational complexity. Three different benchmarking action datasets (KTH, J-HMDB and

UCF-Sports) are used in the experimental works.

The graph-based action recognition methodologies [95] [25] [88] [75] can be classified into three categories (Figure 5.1) according to how the input video is translated into a graph representation: *proximity-based*, *clustering-based* and *segmentation-based*. The *proximity based* methods [95] [24] consider only spatio-temporal proximity for setting a video graph structure. The typical process is to subdivide the video volume into a non-overlapping regular grid of cell regions, with each video graph node corresponding to individual cells. For instance, Chen et al [25] use a set of contiguous voxels to create nodes in the $ST - Subgraph$ and two different linking strategies for edge connection between the nodes. In contrast, the *clustering-based* [8] [75] methods use a clustering algorithm to group the local features based on their spatio-temporal distribution. The clustering process takes into account not only the spatio-temporal proximity but also the local feature characteristics. In the work of [74], a greedy agglomerative hierarchical procedure is used to produce the group of local trajectories. The resulting groups constitute the graph element. The work in [104] uses the hierarchical clustering to iteratively merge similar trajectories (proximity and shape) to obtain an *action component graph*. The *segmentation-based* approaches [44] [88] rely on the segmentation method that produces a 3D local region that forms a node in the video graph. For instance, Soomro et al [88] generates so-called *context graph* by using a video segmentation technique to produce the supervoxel region that corresponds to its node. This class of approach obtains the graph nodes where action boundaries are well preserved. However, a segmentation process on video data results in a higher memory and computational cost.

5.3 Graph construction techniques

This section describes the evaluated video graph construction techniques. We denote:

Figure 5.2: Different types of cell connectivity in 3D space

- a video volume with N local features as $\mathcal{S} = \{(\mathbf{x_i}, u_i)\}_{i=1}^{N}$, where $\mathbf{x_i} = (x_i, y_i, t_i)$ is a spatio-temporal location, u_i is a local descriptor extracted from a video volume.

- the corresponding video graph as $G = (V, E)$, where V, E is a set of nodes and edges respectively

In a nutshell, the video graph construction $\mathcal{S} \mapsto G(V, E)$ can be understood as a partitioning process $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, ..., \mathcal{S}_M\}$ whereby the resulting set $\mathcal{S}_i$ (a local 3D region of the video) forms a node $u_i \in V$ and its pairwise spatio-temporal relationship with S_j determines an edge $e_{i,j} \in E$. The partitioning strategy varies according to the technique involved in the process. However, the common aim is to obtain a graph node that represents a semantically homogeneous region (a human body excluding the background) or a part of the region (a limb of the human body).

Once a node region $\mathcal{S}_i$ is identified, the discriminative score for the node u_i is assigned similar to the re-formulated linear SVM scoring function introduced in Section 5.2 of Chapter 4:

$$score(u_i) = \beta_l + \sum_{j=1}^{K} w_l^j h^j(\mathcal{S}(r)) = \beta_l + \sum_{i \in r_i} w_l^{c_i} \tag{5.1}$$

where the discriminative score is a measure of a support that action l is performed within the region S_i of node u_i and w_l, β_c are the learned bias and support vector of the learned SVM classifier for action label l.

The next subsections present a detailed description of the different graph

construction techniques studied.

5.3.1 Fixed-Grid (FG) approach

This approach is the most simple and fastest technique to achieve a video graph representation in comparison with AR and SV methods. The implementation of this technique is based on the work of Chen et al. [24]. In the constructed graph, nodes correspond to the non-overlapping contiguous cell regions in the video (Figure 5.3). In principle, the smallest cell can be a pixel and the largest possible cell can be a full video volume. In practice, it is observed that a larger cell scope is preferred. Because it reduces computational efficiency (a sparse number of nodes), but also increase the discriminative power of the individual nodes (accumulation of local features statistics). Next, we discuss node structure and their linking strategy in detail.

5.3.1.1 Node Structure

The fixed-grid approach subdivides the video volume into a regular grid of cells with a size of $\delta t \times \delta x \times \delta y$ as shown in Figure 5.3. The local features extracted within each cell constitute a video graph node $u_i \in V$. In the experiment, we set $\delta t = 5$ and 10, and $\delta x, \delta y = \frac{1}{3}$ and $\frac{1}{5}$ of the video frame spatial dimension. After defining the node scope in the video volume, the discriminative score for each node v_i is calculated based on the corresponding local features according to Equation 5.1.

5.3.1.2 Linking Strategy

In image processing, pixel connectivity is the well-known technique that relates a pixel to its neighbours. The node structure (Figure 5.3) generated by the FG approach can be viewed as an array of the image pixels in the 3D space. Therefore we apply a pixel connectivity strategy to create a link between the graph nodes. In the experiments, we explore three types of connectivity (Figure 5.2). In *6-connected neighbourhood*, cell q_i are linked to q_j that are adjacent along the

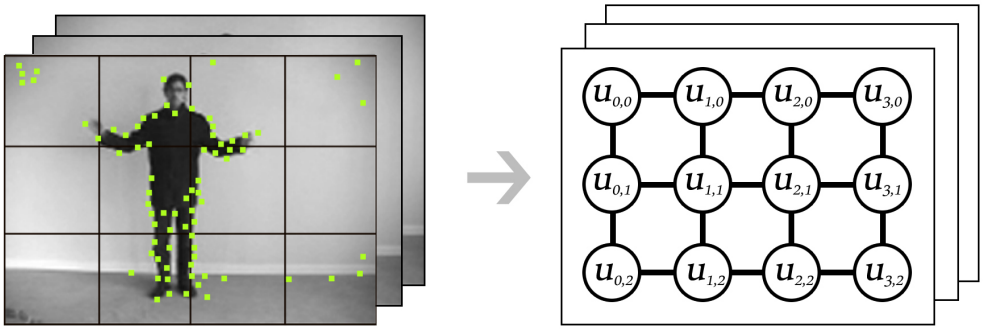


Figure 5.3: The *fixed-grid* (FG) approach divides a video volume into a fixed grid of $\delta t \times \delta x \times \delta y$ and the feature points inside the space-time volume constitutes node u_i in the video-graph.

primary axes. In the case of *18-connected neighbourhood*, cell q_i forms edges between cells that are connected along either one of two of the primary axes. In *26-connected neighbourhood*, the cells are connected along either one, two or all three of the primary axis.

5.3.2 Adaptive-Region (AR) approach

The Adaptive-Region (AR) approach belongs to the class of *clustering-based* methodologies. In particular, AR is based on the extension of *dbscan* [35]. In this technique, graph node is determined using a density clustering procedure using the spatio-temporal density of local trajectories. The method is proposed in the previous chapter and the implementation details can be found in Section 4.4.2 of Chapter 4. However next we briefly discuss on how node and edge are formed with this method.

5.3.2.1 Node Structure

In comparison to similar methods [74] [8], The AR approach does not require one to specify the number of clusters as a prior and can find arbitrarily shaped clusters. The method uses an extended dbscan algorithm (Section 4.4.2 of Chapter 4) to cluster the local features into groups G. The algorithm takes account of local descriptor

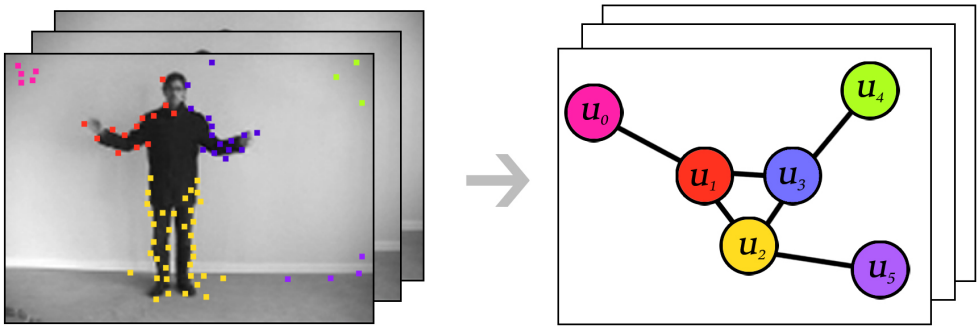


Figure 5.4: The *adaptive – region* (AR) approaches do not suffer from restriction of a pre-defined grid dimension and the resulting graph is sparse and more intuitive compared to *fixed-grid* (FG)

location information but also descriptor similarity. After the clustering step, each group g_i forms node u_i in the resulting video graph as shown in Figure 5.4.

5.3.2.2 Linking Strategy

The strategy is based on the Euclidean distance between nodes. Edge $e_{i,j}$ between nodes v_i and v_j is determined by the distance between their corresponding cluster group g_i and g_j. For instance, let us say that the distance between g_i and g_j is less than a pre-defined threshold value, then edge $e_{i,j}$ is formed and otherwise vice versa.

5.3.3 Supervoxel-based (SV) approach

This subsection describes a video graph construction technique called *supervoxel-based* (SV). This method implementation is inspired by the work [44] and it is based on the video segmentation process that extracts local 3D regions where the object boundaries are well preserved by utilising a rich set of cues such as spatio-temporal proximity, appearance and motion. However, the main drawback is that it requires loading all or a part of the video, which significantly increases memory and computational cost. The recent advances in the parallel programming field (using a GPU accelerator) has a potential to improve the computational speed associate with the segmentation step. In the experiment, the *graph-based* (GB)

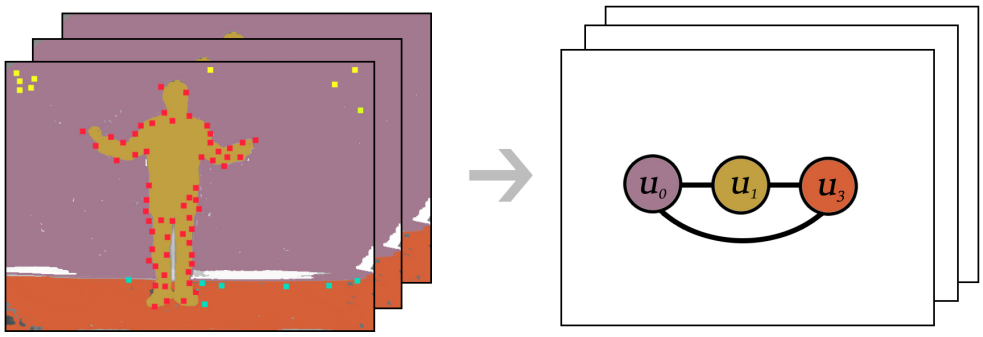


Figure 5.5: The *Supervoxel-based* (SV) uses a video segmentation method in the pre-processing step which results in a superior region where the object boundaries are well preserved. The colour-code indicates the different segmented regions.

[38] segmentation algorithm is used due its efficiency in terms of computational complexity and accuracy trade-off compared to its hierarchical counterpart [38]. The algorithm uses the colour and motion information to determine the supervoxel regions in the video. Therefore it should be noted that the colour information is used only in the graph building process of the SV approach but not in any other part of the SV approach. Further investigation is required to measure the performance contribution from the colour information in the SV method. Next, we describe the SV technique in detail in terms of how it constructs node and edge of the video graph.

5.3.3.1 Node Structure

The video graph node structure (Figure 5.5) is straightforward as it is based on the pre-segmented video regions. For a given input video volume, the segmentation algorithm partitions it into M non-overlapping local 3D regions, called supervoxels, denoted as $\mathcal{C} = \{\mathcal{C}_0, \mathcal{C}_1, ..., \mathcal{C}_{N-1}\}$ and each $\mathcal{C}_i$ is comprised of arbitrary shape and sized pixel points $\mathbf{x}_i = \{\mathbf{x}_i^0, \mathbf{x}_i^1, ..., \mathbf{x}_i^P\}$ in space and time. A supervoxel region $\mathbf{C}_i$ describes node $v_i \in V$ in the video graph $G(V, E)$. The cardinality of $|V|$ is equal to a total number of the segmented regions. A set of local descriptors $\mathcal{S}_i$ are extracted from a supervoxel region $\mathcal{C}_i$ and determine the discriminative score for

node v_i according to Equation 5.1.

The practical challenge is to represent the supervoxel region $mathcalC_i$ efficiently without compromising on memory. The supervoxel is represented as a set of pixel points where each point is a three-dimensional vector $p_i = (x_i, y_i, t_i)$. For instance, a supervoxel comprises of $N = 1500$ pixel points will occupy an array memory of 1500×3 that would have been reduced by approximately 600 times [1] if a 3D bounding box is fitted to represent such a supervoxel. We apply a simple solution to divide the video frame into regular $m \times m$ sized cells and represent supervoxel in terms of such cell region rather than a pixel. It reduces the memory load by m^2.

5.3.3.2 Linking Strategy

We employ the distance-based edge formation technique similar to *adaptive-region* (AR) method. The center point $p_i = (x_i, y_i, t_i)$ is calculated for each supervoxel $\mathcal{C}_i$ region. The Euclidean pairwise distance $d_{i,j}$ is used to determine edge linking between node u_i and u_j. If the distance exceeds a certain threshold value, an edge is formed.

5.4 Evaluation of graph construction techniques

This section presents the evaluation framework and the comparative analysis results on the selected graph construction techniques: *adaptive-region* (AR), *fixed-grid* (FG) and *supervoxel-based* (SV).

5.4.1 Evaluation framework and results

The recognition framework introduced in Chapter 4 has also been used to investigate the different video-graph construction techniques. As highlighted in Figure 5.6, "Spatio-Temporal Video Graph Construction" step is replaced with the corresponding technique discussed in the previous section. Similar to the previous

[1] The vector with a total length of 4500 is needed to store a supervoxel with 1500 pixel points whereas 8 for the case of 3D bounding box and the ratio will be 562.

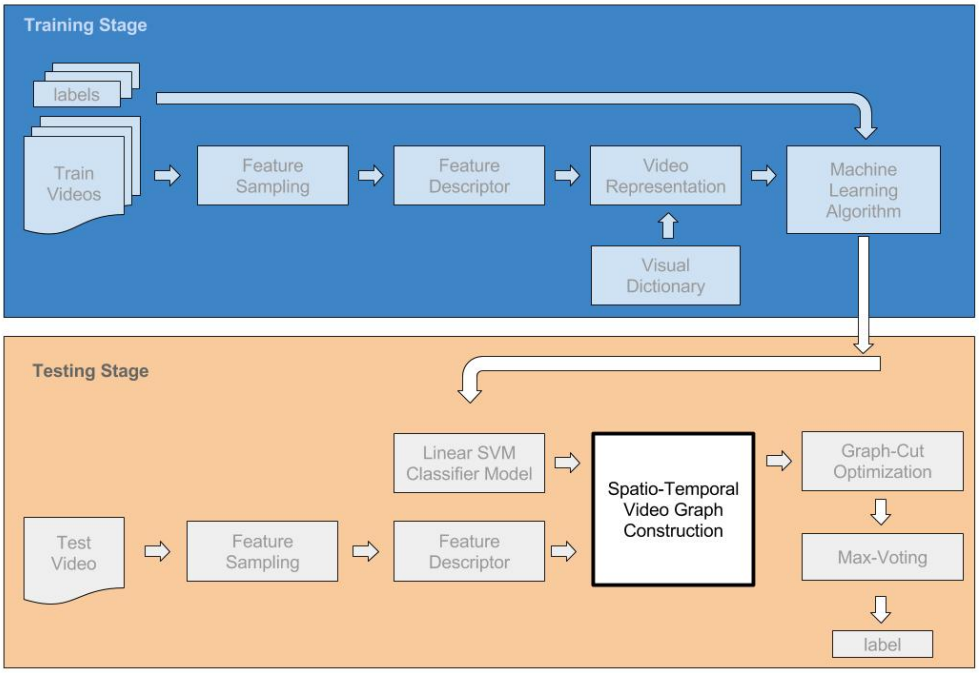
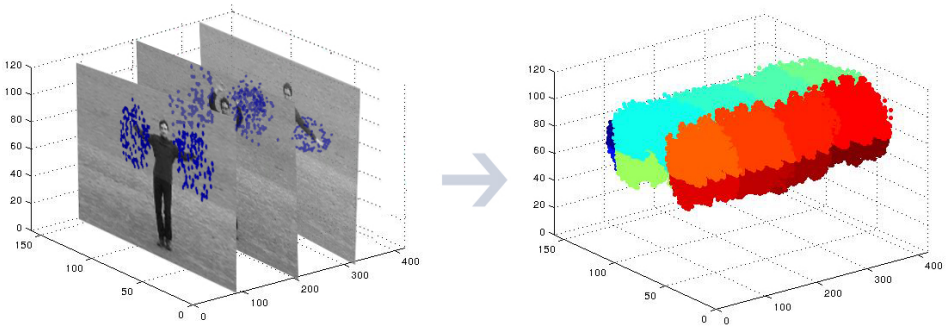


Figure 5.6: The evaluation framework.

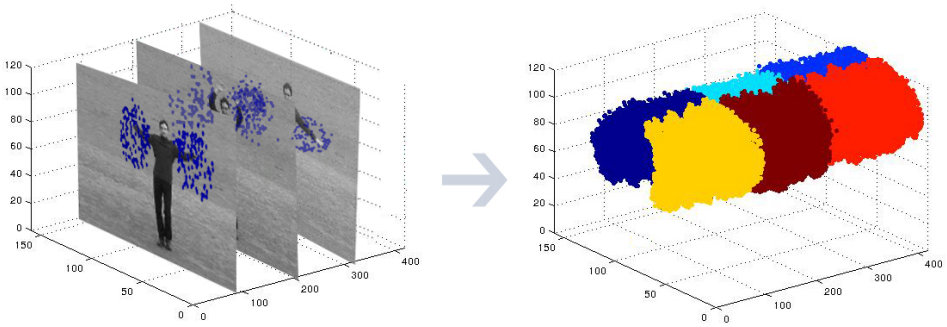
experiments, we have adopted the same feature detector (dense trajectory) and also computed TRAJ, HOG, HOF and MBH descriptors. To build the SVM classifier, we use a standard Bag-of-Features (BoF) model using the computed descriptors. In the recognition stage, the energy function formulated in Section 4.5 of Chapter 4 is used to calculate the optimum label configuration. The voting scheme is applied to find the action label for a test video. In the experiment, the value of scalar parameter λ is varied among the different technique, and it is chosen empirically to give the best performance for each technique. The remainder of this section presents the results on the KTH, UCF-Sports and J-HMDB datasets followed by an investigation of the parameter sensitivity and the computational complexity of each approach.

5.4.1.1 KTH dataset

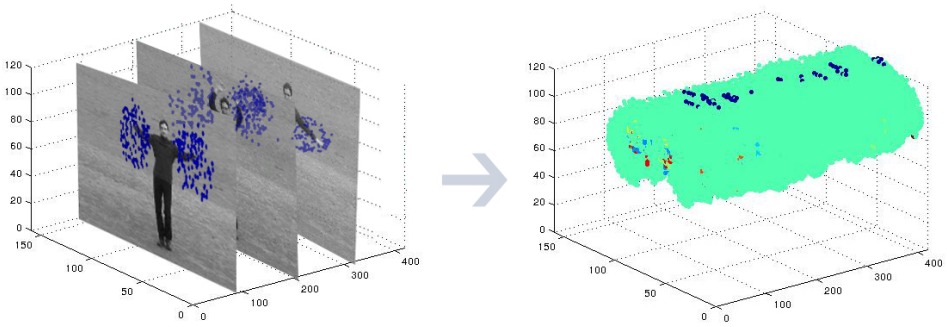
KTH actions is the most commonly used dataset in evaluations of action recognition system. Table 5.1 shows the evaluation result for the different graph



(a) FG technique



(b) AR technique



(c) SV technique

Figure 5.7: Visualization of different graph construction techniques where each node is color coded.

construction methods and the descriptor types. Note that the standard BoF approach is used as a baseline. All techniques outperform the baseline approach. This shows the effectiveness of graph-based techniques in general. In particular, the best result is obtained for AR (98.1%) followed by SV (97.7 %) and FG (96.21 %). The KTH dataset has a homogeneous and stationary background. This might help for the clustering-based approach (AR) to obtain a noise-free and intuitive graph. In contrast, the segmentation-based approach (SV) does not improve the discriminate power as the video sequences are already implicitly segmented (the dataset contains a single actor in the scene). In terms of individual descriptors, the motion descriptor (MBH) significantly outperforms the others as seen many times in the previous experiments. Combining the different descriptors is also proven to be more effective than using individual features.

	TRAJ	HOG	HOF	MBHx	MBHy	Combined
AR	71.54%	72.52%	81.20%	74.61%	86.13%	98.09%
FG	76.65%	73.51%	81.89%	71.52%	82.65%	96.21%
SV	69.12%	71.54%	82.55%	73.85%	85.41%	97.75%
Baseline	66.00%	71.39%	81.50%	70.46%	84.04%	92.75%

Table 5.1: Average accuracy for graph construction/descriptor combinations for the KTH dataset

5.4.1.2 UCF-Sports dataset

We now present experimental results for UCF-Sports dataset. The experimental results are shown in Table 5.2. In comparison with the KTH result, we notice a slight change in the performance order. The best performance is obtained with SV (88.4%) followed by AR (86.4%), FG (78.2 %). The hypothesis is that the segmentation-based approach (SV) can capture the action region efficiently by relying on extra cues, i.e., colour. In particular, the UCF-Sports dataset contains a complex scene with camera motion and dynamic background. In terms of the descriptors, the same trend is observed i.e. that the combined descriptors give the best result followed by MBH and HOF.

	TRAJ	HOG	HOF	MBHx	MBHy	Combined
AR	56.1%	53.5%	69.3%	57.0%	74.3%	86.4%
FG	53.0%	49.2%	65.9%	52.7%	65.5%	78.2%
SV	59.4%	57.5%	71.2%	58.4%	72.5%	88.4%
Baseline	52.1%	49.3%	60.9%	50.7%	62.8%	73.5%

Table 5.2: Average accuracy for graph construction/descriptor combinations for the UCF-Sports Dataset

Action Categories	AR	FG	SV	Baseline
brush_hair	53.56%	59.89%	49.15%	**66.67%**
catch	11.65%	18.81%	14.50%	**20.00%**
clap	**31.46%**	21.16%	28.56%	23.08%
climb_stairs	61.16%	64.98%	63.46%	**66.67%**
golf	91.45%	84.17%	**92.01%**	83.33%
jump	0.00%	0.00%	0.00%	0.00%
kick_ball	2.94%	15.11%	5.09%	**20.00%**
pick	**21.39%**	11.85%	19.98%	8.33%
pour	**75.94%**	69.55%	72.85%	68.75%
pullup	85.54%	65.56%	**88.22%**	68.75%
push	71.45%	50.01%	**75.21%**	50.00%
run	**26.32%**	19.99%	25.12%	18.18%
shoot_ball	35.15%	15.06%	**47.19%**	11.11%
shoot_bow	**39.80%**	25.15%	36.89%	26.67%
shoot_gun	24.32%	13.32%	26.16%	12.50%
sit	81.35%	65.46%	**86.32%**	60.00%
stand	26.19%	**46.89%**	29.02%	45.45%
swing_baseball	76.00%	26.95%	**81.20%**	20.00%
throw	12.50%	5.70%	**15.10%**	0.00%
walk	56.10%	71.00%	55.96%	**75.00%**
wave	15.80%	9.60%	13.00%	8.33%
MAP	**42.86%**	36.20%	**44.05%**	35.85%

Table 5.3: The recognition accuracy for each action in the J-HMDB Dataset (Split 1)

Action	Camera motion		Visible Part		Camera View-Point			
	pre.	abs.	full	upp.	ba	fr	le	ri
action group where baseline performs best								
brush_hair	20%	80%	2%	98%	48%	36%	9%	7%
catch	76%	24%	73%	27%	11%	43%	24%	22%
stand	73%	27%	54%	36%	0%	77%	8%	15%
Avg.	**56%**	44%	**57%**	43%	20%	52%	14%	15%
action group where graph-based approach performs best								
pick	68%	32%	90%	10%	16%	45%	10%	29%
pullup	62%	38%	64%	36%	48%	40%	2%	10%
sit	54%	46%	63%	37%	0%	63%	5%	32%
Avg.	**61%**	39%	**28%**	72%	21%	49%	6%	24%

Table 5.4: Two group action categories that performed best for the baseline and graph-based approaches

5.4.1.3 J-HMDB dataset

Table 5.3 shows the experimental result for the different graph construction methods for the J-HMDB dataset. Note that the combined descriptor type (TRAJ + HOG + HOF + MBH) is used. The dataset contains different varieties of single person action performed in the *uncontrolled* scenario. In terms of MAP (Mean Average Precision), all three techniques (AR: 42.8%, FG: 36.2% and SV: 44.0%) outperform the baseline (35.8 %). In particular, SV and AR techniques have a significant margin of improvement. This result demonstrates the effectiveness of SV and AR techniques in *uncontrolled* scenes. Also, it confirms that the graph construction process has a significant effect on the overall performance (9 % (MAP) improvement over the baseline). However, interestingly, there is a few action classes that the baseline outperforms: *brush_hair*, *catch*, *climb_stairs* and *walk*. The J-HMDB dataset provides the meta-data of each action category, i.e camera motion, performer visible part, camera view-point and video-quality. Table 5.4 shows the two groups of top outperforming action categories for both the baseline and graph-based approach (AR, FG, SV) represented according to the meta-data. The camera motion and visible-part cue have the most significant difference between the two groups of actions. From this observation, the standard

Dataset	Resolution	AR	FG	SV
KTH	160×120	36 fps	262 fps	2.5 fps
UCF-Sports	720×480	21 fps	136 fps	0.8 fps

Table 5.5: The average frame rate at runtime for the two datasets

BoF (model) is well suited for the scenario where there is no or less camera-motion. This claim can be supported by the results on the simple KTH dataset where there is the small margin between the baseline and our graph-based approach. In contrast, in complex situations such as view-point changes and camera motion, the proposed graph-based approach is more suitable.

5.4.1.4 Computational complexity considerations

This subsection compares the graph construction techniques based on their computational speed. The comparison is performed on a set of videos from KTH and UCF-Sports dataset with the resolutions of 160×120 and 720×480 respectively. The run time is measured on a machine with 64-bit Ubuntu 14.04 OS, Intel Core i5-4690K CPU (3.50 GHz $\times 4$) with 24GB RAM. Note that the entire construction process, i.e., the video segmentation for SV, is included in the run-time. Among the techniques, FG is the fastest (KTH: 262 fps and UCF: 136 fps) followed by AR (KTH: 36 fps and UCF: 21 fps). The slowest technique is SV (KTH: 2.5 fps and UCF: 0.8 fps). Obviously, the video segmentation step for SV is the most resource intensive operation. However, note that the computational cost can be improved by adapting the video segmentation code using a parallel programming model based on either multi-core processor or GPU (graphical processing unit). Further investigation is needed to validate the speed improvement for the SV method with the optimised processing pipeline.

5.4.1.5 Parameter sensitivity evaluation

We further investigate the performance of the various graph construction techniques in terms of different parameter settings with KTH and UCF-Sports datasets. This

Parameters	KTH				UCF-Sports			
	16	32	64	128	16	32	64	128
$r_{sp} = 5$, $t_{tmp} = 5$	92.5%	94.0%	94.0%	91.5%	73.8%	78.2%	75.5%	70.1%
$r_{sp} = 5$, $t_{tmp} = 10$	94.3%	**97.2%**	**97.2%**	92.6%	74.6%	82.6%	**84.2%**	72.2%
$r_{sp} = 10$, $t_{tmp} = 5$	**95.7%**	**97.2%**	**95.2%**	93.3%	78.7%	**86.4%**	**85.7%**	73.8%
$r_{sp} = 10$,$t_{tmp} = 10$	95.8%	93.3%	95.5%	91.6%	75.5%	79.4%	78.2%	73.8%

Table 5.6: The parameter sensitivity evaluation for AR technique

analysis helps us to understand the robustness of the graph construction techniques. We have studied the following parameter setup:

- **Adaptive Region (AR):** The technique is based on the extension density-clustering and the algorithm depends on the following parameters: the minimum number of points ($minPts$) and maximum search spatial radius (r_{sp}) and temporal radius (t_{tmp}). We evaluate two values of r_{sp} and t_{tmp} and four different $minPts$ values.

- **Fixed-Grids (FG)**: In this method, there are three parameters (i.e. δt, δx and δy) that control the subvolume size. In the experimental work, we set $\delta t = \{5, 10\}$ and δx, $\delta y = \{\frac{1}{3}, \frac{1}{5}\}$ of the video frame dimension. For connection, we study all possible linking in 3D space, i.e 6-neighourhood, 18-neigbhourhood and 26-neigbhorhood.

- **Supervoxel-based (SV):** This technique uses a video segmentation algorithm called Graph-based (GB) to generate supervoxels. There are two main parameters that control the size of the resultant supervoxel regions, i.e merging threshold C and minimum segment area Min. We also examine the effect of cell size m and the total 12 combinations of different parameter values are evaluated in the experiment.

Table 5.6 shows the results for the AR method. The parameter $minPts$ has the highest performance gain at values of $minPts = 32, 64$ for both datasets. This parameter can be understood as density control and a lower value tends to form a larger node in the video graph or smaller node at larger values. We speculate this

Approach parameter	KTH			UCF-Sports		
	$N = 6$	$N = 18$	$N = 26$	$N = 6$	$N = 18$	$N = 26$
$5 \times 1/5 \times 1/3$	93.3%	94.0%	94.0%	72.2%	72.2%	73.6%
$10 \times 1/5 \times 1/3$	94.0%	95.2%	95.2%	74.4%	75.2%	74.6%
$5 \times 1/3 \times 1/5$	**95.7%**	**96.2%**	94.0%	73.3%	**76.4%**	**76.4%**
$10 \times 1/3 \times 1/5$	**95.8%**	95.0%	**95.5%**	74.5%	**76.4%**	**75.5%**

Table 5.7: The parameter evaluation for FG technique

parameter is dependent on the type of feature detector used because the feature point density in the video volume varies significantly among different detectors. In our case, for both datasets, the same detector is utilised, and the same performance gain is observed at the same value interval. Regarding the maximum search parameters, the top performance results are obtained at $r_{sp} = 10$ and $t_{tmp} = 5$ and these parameters are also responsible for the resulting node size and the optimum values should be tuned accordingly. We assume that these parameter values can be set constant for different datasets as long as the framework is using the same feature detector.

Table 5.7 shows the performance result which has been obtained by changing the parameter settings for the FG technique. We have observed no significant improvement for different sub-volume scales. However, it can be seen that smaller sub-volume size does not increase the performance. For KTH and UCF-Sports, the highest performance is measured when sub-volume parameter values are $\delta t = 10$ and δx, $\delta y = \frac{1}{3}$. Regarding neighbourhood type, for KTH, the simple connection achieved the top performance gain. In essence, sparse edge connectivity implies that the action label for a sub-volume region is determined more independently i.e. without undue influence from its surrounding regions. In the case of a simple dataset KTH, the discriminative power of local region alone is sufficiently accurate to achieve good performance. In contrast, in UCF-Sports which is the more challenging dataset, dense neighbourhood connection has a positive impact on the performance. The top gain (76.4%) is observed when the neighbourhood parameter $N = 26$.

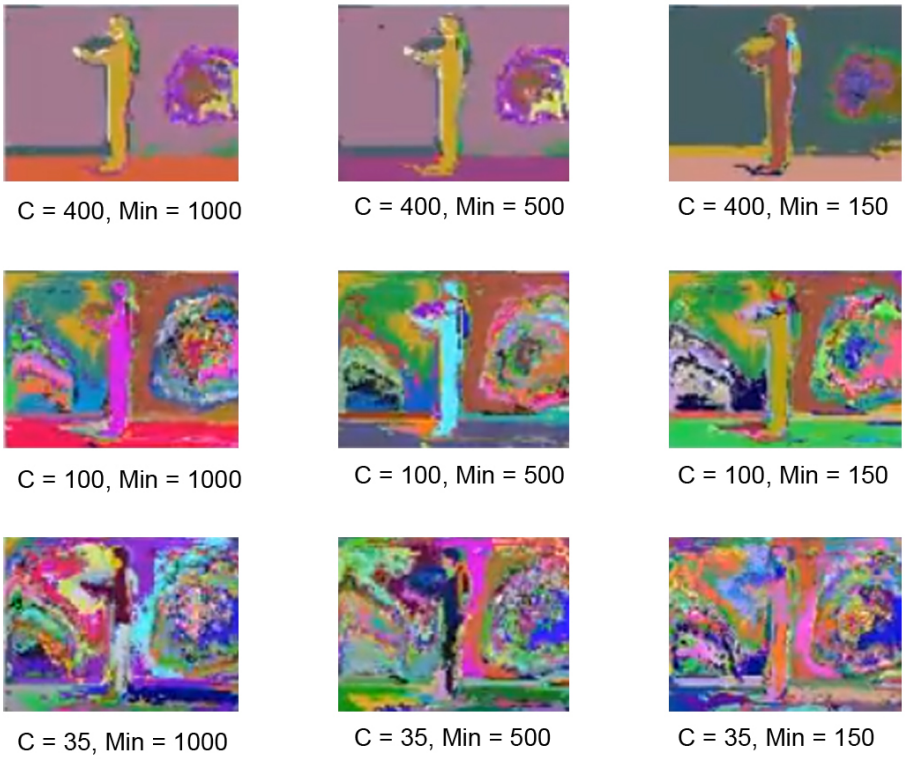


Figure 5.8: Video frames showing the effect of merging threshold C and minimum segment area Min on the number of resulting supervoxels using a LIBSVX video segmentation tool (each supervoxel is color-coded)

Approach parameter	KTH			UCF-Sports		
	$m = 4$	$m = 8$	$m = 16$	$m = 4$	$m = 8$	$m = 16$
$C = 400$, $Min = 500$	**97.8%**	**97.8%**	91.6%	**88.4%**	85.7%	78.7%
$C = 400$, $Min = 1000$	**95.9%**	**94.2%**	89.2%	**87.6%**	**85.7%**	77.9%
$C = 100$, $Min = 5000$	89.2%	91.6%	88.7%	79.4%	79.4%	75.5%
$C = 100$, $Min = 1000$	88.7%	89.2%	87.5%	81.7%	79.4%	76.9%

Table 5.8: The parameter sensitivity evaluation for SV technique

In the supervoxel-based SV graph construction technique, the evaluation result is shown in Table 5.8. As you can see, the best result has obtained for KTH (97.8 %) and UCF-Sports (88.4%) with $C = 400$ and $Min = 500$. These parameters control the size of the individual supervoxel region and higher values result in a number of supervoxel cells as shown in Figure 5.8. However, the discriminative power of video graph node constructed on top of the smaller cells is weakened when associated with fewer local descriptors. The same trend is also observed with the FG technique where a larger sub-volume results in better accuracy. The worst performance is measured with $C = 100$ and $Min = 1000$. In comparison with other techniques (AR and FG), SV is the most sensitive to the different parameter values. In both datasets, SV's best performance is higher than other two techniques. Interestingly, the worst performance is also less than AR and FG worst performance. Therefore, when using the *segmentation-based* technique, one should be careful about tuning the optimal parameter values.

5.5 Summary

This chapter presents various classes of approaches for video graph construction and their comparative analysis. The objective is to perform a comprehensive evaluation of the different strategies considering their impact on the overall performance of the recognition system as well as considering the associated computational complexity and parameter sensitivity. In most of the tests, SV obtains the best results, closely followed by AR. This confirms the robustness and the effectiveness of the supervoxel-based technique, specifically in the case of the complex scene. However, from a computational perspective, SV is computationally expensive in comparison to the other evaluated techniques. Overall, based on the trade-off between computational cost and performance for the action classification task, the graph-based recognition framework with a video graph configuration using the AV method gives the best performance across the different datasets. Finally, the chapter demonstrates that

the graph construction technique has a significant impact on action recognition performance.

Chapter 6

Spatial-localization of human action in video

6.1 Overview

This chapter presents a graph-based action localisation framework as an extension of the proposed approach discussed in Chapter 4. Action localisation is a challenging task as it requires the classified action to be spatially and temporally localised. In this chapter, we develop a video graph that accommodates not only local features but also region-based features to facilitate the localisation. The action localisation is performed by maximising the score associated with nodes and edges in the video graph. The effectiveness of the proposed approach is investigated and benchmarked to the state-of-the-art using various localisation datasets.

6.2 Introduction

Recent methods [101] for action recognition mostly focus on action classification rather than action localisation. Mostly the top-performing classification approaches in the action modelling process [56, 73, 97, 101] explicitly or implicitly use the background information, i.e., the region where the action is not performed. This significantly contributes to the classification performance [21, 89] but prevents the identification of the region where the action is taking place. However, the action localisation task requires the classified action to be localised both spatially

and temporally. The temporal localisation can be efficiently [88] detected using the typical classification method coupled with the sliding window technique. However, the spatial localisation is complicated for the classification-based methods due to the above mentioned use of the background. This chapter aims to address the localisation problem emphasising the spatial localisation of the action. In particular, we propose an action localisation framework based on a directed video-graph and present two major contributions as follows:

First, we propose a new directed video graph suited for the localisation task. In the graph, the node describes a candidate action region and its connectivity (edge) describes the similarity with the adjacent region regarding cues such as region colour, motion and region geometry. The discriminative score of each node is calculated using a late fusion technique based on the corresponding local and regional features. The late fusion provides a means to integrate a variety of features of different type and dimensions (local, global and regional). Also, it makes the video graph representation sufficiently flexible to combine a richer set of features that has a potential to increase the performance. In Section 4.6.2 of Chapter 4, we noted that local descriptors are not sufficiently discriminative to tackle the challenging localisation task. Hence the regional descriptor is added in the pipeline to complement the local descriptor. Secondly, this chapter presents the application of the maximum-path finding algorithm to identify the localised action. This method has been successfully adopted in an object tracking problem [12] where it showed its effectiveness. We propose that action localisation can be understood as semantic concept tracking over time. Therefore we investigate whether this approach can be extended to the challenge of action localisation. In comparison to the undirected graph strategies introduced in the previous chapter, the directed graph is chosen to incorporate the sequential element in the representation.

The proposed approach is evaluated using two benchmark action datasets, namely J-HMDB and UCF-Sports. In the literature [55, 94, 111], these datasets

have been extensively used for action localization. The remainder of this chapter is organized as follows: Section 6.4 presents a overview of the proposed framework followed the video graph construction (Section 6.4.1) and the maximum-path (MPF) formulation on the video graph (Section 6.5) and the experimental result (Section 6.6).

6.3 Related Work

Action localisation is becoming crucial for effective analysis of the *uncontrolled* video capture scenario that consists of videos captured in complex settings that have significant background clutter or contain multiple actors or actions. The earlier works [94] propose to directly use a classifier on the action localisation task using 3D sliding window or similar technique. The main advantage is that the mid-level representation may not be necessary. However, the sliding window approach substantially increases the computational complexity when an input video has a long duration or high-resolution.

In other approaches, the localisation is primarily based on the action proposal [40] inspired by the success of the region proposal methods for the object localisation task in 2D images. For instance, the *objectness* technique [4] for object localization is extended to video by [12] and *selective search* [96] is modified into spatio-temporal tubelets in [111]. This class of method overcomes the short-coming of classifier-based approach by investigating the selected part or region of the video rather than the entire video. The region proposal approach is computationally efficient and obtains good results for action localisation in comparison to the other methods. We adopt this strategy in the development of the video graph.

Recently the approaches based on features from convolutional neural networks [53] [60] have achieved significant progress in the object detection and image classification task. In particular, the approaches based on regional convolutional neural networks (RCNN) [76] are the state-of-the-art that have

produced best results with a high-margin of difference compared to competing approaches for the object localisation task. Gkioxari et al [40] first applied the region-based convolutional feature (R-CNN) in the action localisation task and achieved promising results. However, the action detection is frame-based and can not take into account the temporal dynamics of the action which is an important cue in any action recognition system. More recently, Weinzaepfela et al. [111] introduced a method to overcome this weakness by fusing the region-based feature (R-CNN) with a track descriptor, that is similar to the trajectory feature used in our approach, and achieved further improvement. This shows that combining the frame-level descriptor, such as R-CNN, with local temporal features (motion trajectories) that are complementary to each other and improves the performance. Our framework embeds both the region-based convolutional features (R-CNN) and the local trajectory features to obtain the discriminative graph model.

In an approach similar to our work, [99] introduces an action localisation framework based on action proposals from dense trajectories features. However our proposed framework differs in several key aspects: first, we develop the effective graph structure that is capable of integrating the different feature types i.e., local trajectory and RCNN features. Furthermore, the additional cues such as local, motion and region geometry are captured as the graph edges. Finally, the localisation is performed by maximising the path score (MSP) in a video graph.

6.4 Proposed Framework

In the localisation framework, given a video, we first apply a region proposal technique at the frame level. This step produces the candidate action regions that form the basis for constructing the video graph. In the video graph, the node represents the region along with its corresponding features and the edge describes the similarity with its adjacent region. To assign the discriminative node score, support vector machines (SVMs) classifier is built with training videos for each

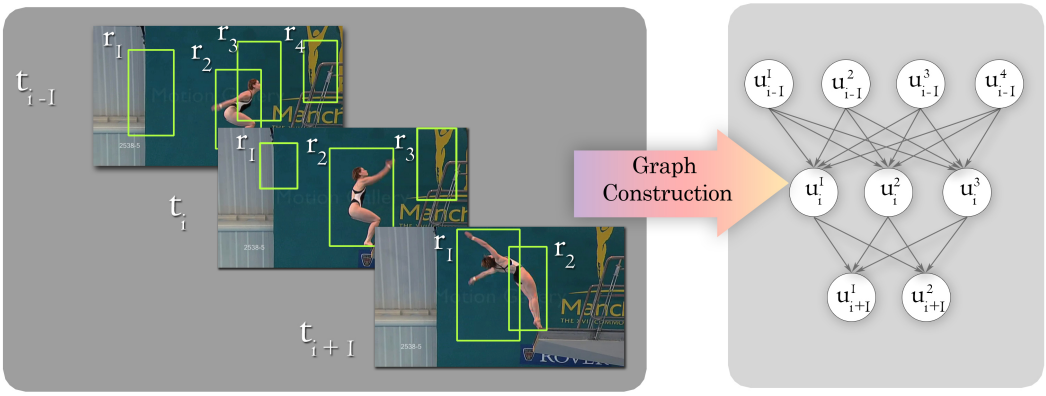


Figure 6.1: A directed video graph construction process

type of feature (local and regional) and integrated using a late fusion method (details in Section 6.4.1.3). Finally, the maximum-path finding (MPF) algorithm is used to find the maximum scoring path in the video graph of a test video. The regions associated with the maximum path is considered as the localised action. Next, we describe construction of the video graph.

6.4.1 Video Graph Construction

Given a video sequence $\mathbf{I} = \{I_1, I_2, .., I_n\}$, where I_t is a static frame at the time instance t, we construct the corresponding video graph $G(V, E)$. As shown in Figure 6.1, the node u_t^j describes the action candidate region regions defined by a rectangular region $r_t^j = (x_t^j, y_t^j, h_t^j, w_t^j)$ in the static frame I_t. There are various ways to acquire candidate regions such as dense sampling [24] that subdivides the frame into fixed grids at different scales. However, it has an implication to substantially increase the number of candidate regions whereby the computational complexity increases. Consequently, the alternative strategy is to use the region proposal method that efficiently identifies the likely object regions using only texture and edge information. Although any object proposal can be used in our framework, the selective-search method [96] is used in the experiment due to the availability of its implementation [1]. The region proposal is applied on the video

[1] http //koen.me/research/selectivesearch/

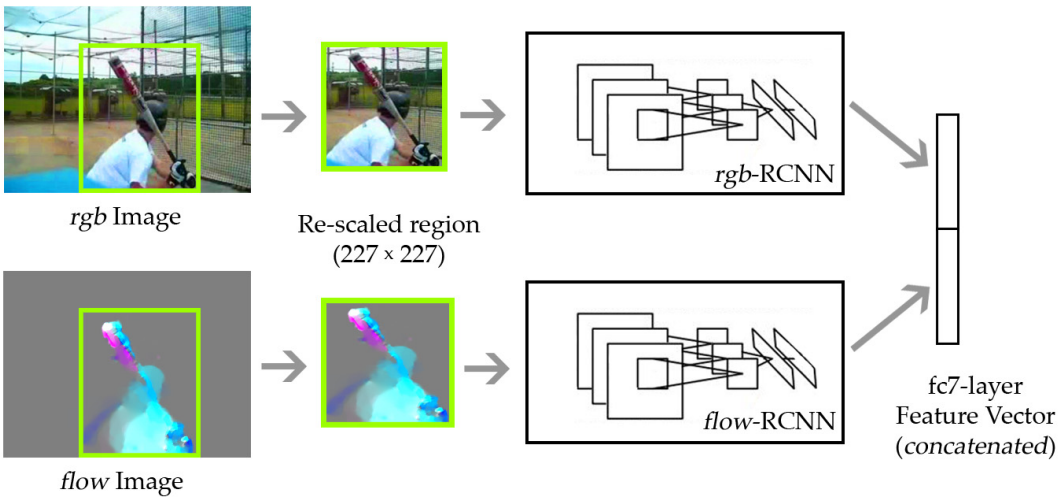


Figure 6.2: The regional feature calculation process

frames to generate approximately 2000 candidate action regions per frame. Furthermore, we filter the candidate regions where there is no significant motion according to the method [40]. This significantly reduces the number of a region by 85% with a loss of only 4% (action positive regions).

Once the node region is determined, the next important step is feature extraction process. The recently [40, 111] successfully used region-based neural network (RCNN) features for action localisation, are adopted for describing the node region. The RCCN is shown to be highly discriminative as well as able to describe the region with arbitrary size. However, it does not capture the temporal dynamics of the action beyond two consecutive frames. Thus, the local dense trajectory feature is extracted from the node region to complement the RCCN feature. Next, we discuss how the features are extracted in detail.

6.4.1.1 Regional feature

Gkioxari et al [40] introduced RCNN features that operate separately on the image and optical flow. We use the same set of RCNN features i.e., rgb-RCNN and $flow$-RCNN. Given a region re-scaled to the dimension of 227×227, the rgb-RCNN operates on a three-channel of the colour image. It captures the static appearance

of the actor/scene. For $flow$-CNN feature extraction, the flow image is first formed by transforming the dense optical flow into a 3-channel image (the flow x & y component and its amplitude) followed by the re-scaling and convolutional process. The $flow$-RCNN captures the motion pattern of the action. In the experiment, the pre-trained RCNN network[2] is used to compute the rgb-RCNN and the $flow$-RCNN features from the video frame region associated with the graph nodes as show in Figure 6.2. We use the concatenation of the $fc7 - layer$ (4096 dimension) features of rgb-RCNN and $flow$-RCNN network. We refer to the concatenated vector (9192 dimension) as v_i^f for the node u_i with the corresponding region r_i.

6.4.1.2 Local feature

Although the proposed approach is not constrained by the type of local features, we select the same feature/descriptor as in the previous chapters. In particular, we use the dense trajectory [101] that extracts the motion trajectories. In the experiment, the trajectory length is set short $L = 15$ frames to avoid the drifting trajectory problem. We apply the feature extraction for the entire video. Then the video graph node u_i is associated with the local features located in its region r_k. For each feature, four descriptors (TRAJ, HOG, HOF, MBH) are calculated and concatenated to form a single vector.

6.4.1.3 Classifier training and node discriminative score

Since we use two sets of features, two separate classifiers (regional and local) are trained. For the regional feature, we train SVM classifiers for each action class $c \in C$, where ground truth regions are considered as positive examples and regions that overlap by factor of less than 0.3 times the area with the ground truth as negative. During training, the hard-negative mining technique is used. This strategy has shown significant improvement compared to traditional training [77] in the object localisation task.

[2]https://github.com/gkioxari/ActionTubes

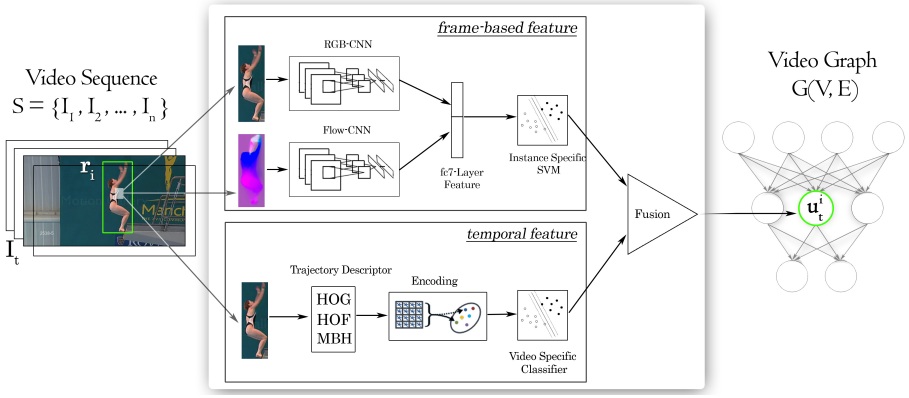


Figure 6.3: The procedure of calculating the node score of the action graph. We use two different feature types: *local* and *region-based*. Each feature is aggregated into final node score using late-fusion method.

For training a classifier for local features, we use the Bag-of-Features (BoF) model with the re-formulated scoring function introduced in Chapter 4. In the experiment, the one-against-rest strategy is used to produce a binary classifier for each action class l. Once the SVM classifier is learned, the discriminative score for node u_i is calculated as follows:

Regional Classifier: Given a region r_i of node u_i with the extracted regional feature vector v_i^f and the trained classifiers for action class. Each node u_i in the video graph is assigned with a discriminative score for action class l:

$$score_l^f(u_i) = \beta_l + w_l' \cdot v_i^f \tag{6.1}$$

where the discriminative score is the estimate of a likelihood that action l is performed within the region r_i of the node u_i and w_c, β_c are learned bias and support vector of the trained regional SVM classifier for action label l.

Local classifier: As we formulated the localisation as the maximum path, the discriminative score should be able to be combined additively to give the cumulative score for traversing the path in the video graph. The additivity requirement on the classifier property (discussed in Section 4.5.2 in Chapter 4) is also applicable here. Therefore, we use the linear (additive) SVM classifier for training. In particular,

for each training video, we compute the BoF encoding with K visual words. A training video with N local features is described by the set $S = \{(\boldsymbol{x}_i, v_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i = (x_i, y_i, t_i)$ refers to the local feature position in space and time, and v_i is the associated local descriptor. Let $h(S)$ be function maps feature set S into K-dimensional BoF coded vector.

The one-against-rest strategy is to learn a linear SVM for each action class $c \in C$. The resulting score function can be re-formulated as a sum over the contribution from each feature (according to the Section 4.5.2 in Chapter 4) and this formulation is used calculate the discriminative score for node u_i ,

$$score_l^t(u_i) = \beta_l + \sum_{j=1}^{K} w_c^j h^j(S(r)) = \beta_c + \sum_{i \in r_i} w_l^{l_i} \tag{6.2}$$

where $h^j(S)$ denotes the j-th bin count for histogram $h(S)$. The j-th word is associated with a weight $w^j = \sum_i \alpha h^j(S_i)$ and w_l, β_l are learned bias and support vector of the learned SVM classifier for action l.

Late-Fusion: To calculate the final discriminative score for a given node u_i, we use the fusion technique to combine the respective scores as follows:

$$score_l(u_i) = \alpha \cdot score_l^r(u_i) + (1 - \alpha) \cdot score_l^l(u_i) \tag{6.3}$$

where α is a scalar. In the experiment, we use this parameter to investigate the respective feature type contribution to localisation performance.

6.4.2 Edge weight

The edge $e(u_i, u_j)$ represents the similarity between given nodes u_i, u_j. In the proposed video graph, the edge is formed between temporally adjacent nodes as shown in Figure 6.4 and the edge direction is used to enforce the path to flow in time. The action localisation can be understood as semantic concept tracking over time. In tracking methods [7] [10], the authors use the color, motion cues for successful object tracking. A rich set of cues is crucial for the accurate registration

of the object over different frames. Therefore we propose to combine multiple cues (colour, descriptor and geometric) to determine the edge weight:

$$e(u_i, u_j) = \begin{cases} f_c(r_i, r_j) + f_g(r_i, r_j) + f_d(r_i, r_j), & \text{if } r_i \text{ and } r_j \text{ temporally adjacent} \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

where r_i, r_j are the corresponding region for the node u_i and u_j, respectively and the term f_c, f_g and f_d are defined as follows:

- **Color Similarity Term** (f_c): Many colour descriptors have been proposed in the literature. In the experiment, we use the region-based color descriptor proposed by Van et al. [98] due the availability of its implementation[3]. The color descriptor (108 dimension) is extracted from the region r_k for each color hannel (RGB) and concatenated to create the combined descriptor c_k. Then color similarity term $f_c(r_i, r_j)$ is defined as a cosine measure between $cosine(c_i, c_j)$ where c_k is concatenated color descriptor extracted from a region r_k of the node u_k. The cosine similarity is selected due to it has positive space, where the outcome is neatly bounded in a range of $[0, 1]$.

- **Descriptor Similarity Term** (f_d): This term is based on the assumption that the features extracted from the same actor/action should resemble similarity. In the experiment, we use the regional feature to determine the descriptor similarity term as follows: $f_c(r_i, r_j) = cosine(v_i, v_j)$ where v_j v_i is the regional features extracted at the region r_i and r_j respectively.

- **Geometric Similarity Term** (f_g): This term encourages the spatial coherence between the node regions. In other words, the term scores high if the spatial extent significantly overlaps. The geometric similarity is defined as intersection of over union measure, $f_g(r_i, r_j) = IOU(r_i, r_j)$ i.e the full overlap between the regions gives a score of 1.

[3] http://lear.inrialpes.fr/people/vandeweijer/color_descriptors.html

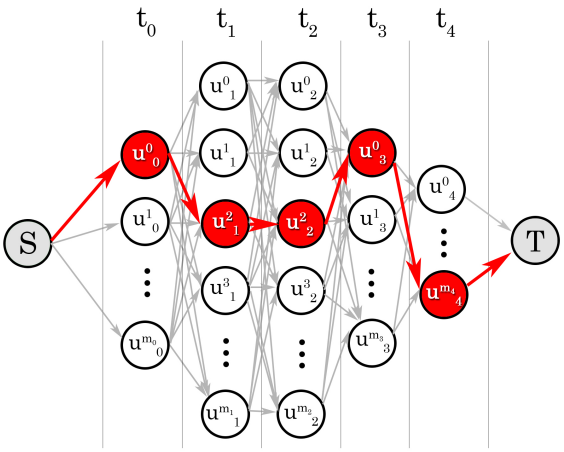


Figure 6.4: The maximum path in the graph considered to be the localized action. In the experiment, we use the Boykov-Kolmogorov method to calculate the maximum flow between node S, T .

6.5 Action localization in the video-graph

Assuming a video sequence is mapped into directed video-graph $V(G, E)$ as discussed in Section 6.4.1. We now describe how to localize action in the graph. Given a path p, the score $M_c(p)$ is defined as:

$$M_l(p) = \sum_{i \in p} score_l(u_i) + \lambda \sum_{(i,j) \in p} e(u_i, u_j) \tag{6.5}$$

where l is the action class and λ is a scalar. The edge weight $e(u_i, u_j)$ scores high if the corresponding node regions r_i, r_j overlap and agree in terms of color and regional feature. To localize the action, the problem becomes to find the optimal path $p*$ with highest accumulated score:

$$(p^*, l^*) = \arg\max_{l \in L} \arg \max_{p \in path(G)} M_l(p) \tag{6.6}$$

where $p^* = [u_1, u_2, ..., u_t]$ is the trajectory that maximizes the video graph with action class l^*. Finally the corresponding regions $[r_1, r_2, ..., r_t]$ will be considered as the localised action in the video sequence. The Maximum path problem is efficiently solved using dynamic programming. In the experiment, we have used Boykov-

Kolmogorov algorithm to find the maximum flow in the graph by adding zero-weighted source S and terminal T node as shown in Figure 6.4.

6.6 Evaluation

6.6.1 Datasets

We evaluate our approach on two widely used datasets, namely UCF Sports [63] and J-HMDB [45]. On UCF sports we compare against other techniques and show substantial improvement from state-of-the-art approaches. We present an ablation study of our CNN-based approach and show results on action classification using our action tubes on JHMDB, which is a substantially larger dataset than UCF Sports. The UCF Sports dataset consists of 150 videos with 10 different actions. There are on average 10.3 videos per action for training, and 4.7 for testing 1 . J-HMDB contains about 900 videos of 21 different actions. The videos are extracted from the larger HMDB dataset [24], consisting of 51 actions.To date, UCF Sports has been widely used by scientists for evaluation purposes.

6.6.2 Experimental Protocol

To quantify our results, we report AUC curves for the UCF-Sports dataset, a metric commonly used by other approaches. A number of recent methods have used AP metrics, and we have compared our method performance against these reported methods for both the J-HMDB and UCF-Sports dataset.

6.6.3 Results

6.6.3.1 UCF Sports

In Figure 6.5 we plot the average AUC (Area Under Curve) for different values of σ (IOU parameter). The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. We plot the curves as produced by the state-of-the-art approaches, Jain et al. [43] , Wang et

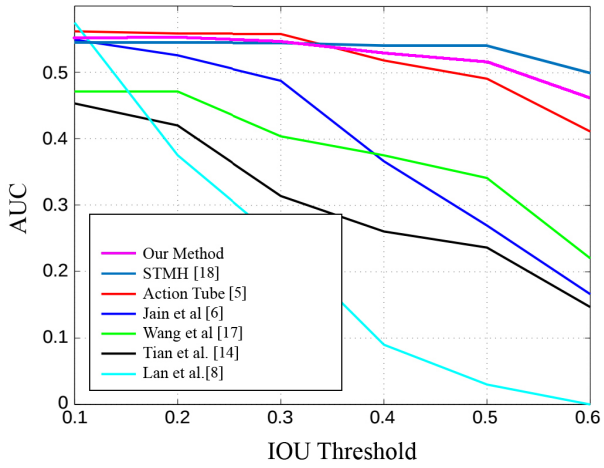


Figure 6.5: AUC for varying IoU thresholds for UCF-Sports Dataset

al. [103], Tian et al. [94], Lan et al. [55], Action Tube [40] and SMTH [111]. Our approach outperforms most of these techniques, showing the most improvement for high values of overlap. In particular, the proposed method achieves the competitive performance with the recent state-of-the-art work [111] only falling short by a slight margin. For comparison with the state-of-the-art methods, as shown in column 2 at Table 6.5, our method achieves competitive performance of MAP = 88.7 % with IOU parameter $\sigma = 0.5$.

6.6.3.2 J-HMDB dataset

First, we report the performance of the 21 actions of the J-HMDB dataset. Table 6.1 presents the result by the different combination of features used: local (TRAJ, HOG, HOF, MBH), regional (flow RCNN + RGB RCNN) and fused (local + regional). It is apparent that the fused approach consistently outperforms the individual features. The regional feature performs significantly better for almost all actions in comparison with the local counterpart. It proves the highly discriminative nature of the convolutional feature. However, interestingly, the local feature outperforms the regional only in one action class: *climb_stairs* which has an huge camera motion (79%) (Table 6.2). Regarding MAP, feature fusing (57.1%)

action class	local	region	combined
brush_hair	79.1%	59.8%	84.9%
catch	27.8%	11.6%	33.6%
clap	57.3%	20.1%	60.3%
climb_stairs	21.8%	23.0%	63.9%
golf	92.3%	29.7%	95.7%
jump	14.4%	9.0%	14.5%
kick_ball	14.5%	3.7%	15.5%
pick	42.4%	14.4%	53.9%
pour	92.3%	70.3%	97.1%
pullup	89.8%	52.1%	96.2%
push	63.3%	31.6%	55.8%
run	37.9%	13.7%	42.9%
shoot_ball	23.0%	19.9%	26.5%
shoot_bow	79.1%	31.0%	79.6%
shoot_gun	25.7%	19.6%	48.5%
sit	40.0%	30.5%	50.2%
stand	39.1%	32.3%	42.6%
swing_baseball	79.5%	9.6%	81.3%
throw	25.9%	11.1%	28.5%
walk	70.7%	33.7%	77.8%
wave	37.0%	23.9%	50.0%
MAP	26.2%	50.1%	57.1%

Table 6.1: The performance by action class for J-HMDB dataset (Split 1)

Action class	Camera motion	Video quality	Camera view-point				Visible body part			
			back	front	left	right	full	head	lower	upper
brush_hair	21%	98%	41%	41%	15%	2%		10%		90%
catch	76%	45%	19%	33%	25%	23%	75%			25%
clap	32%	51%		93%	2%	5%	7%			93%
climb_stairs	79%	49%	75%	20%		5%	68%		33%	
golf	59%	68%		19%	7%	74%	100%			
jump	59%	5%	15%	26%	31%	28%	100%			
kick_ball	45%	44%	67%	22%	6%	6%	89%		8%	3%
pick	73%	53%	15%	33%	23%	30%	90%		3%	8%
pour	32%	65%		89%		11%				100%
pullup	56%	60%	51%	36%	2%	11%	60%			40%
push	89%	12%	21%	10%	29%	40%	95%			5%
run	66%	50%	35%	28%	20%	18%	93%		3%	5%
shoot_ball	30%	50%	80%		8%	13%	65%			35%
shoot_bow	9%	50%	9%	85%		6%	17%			83%
shoot_gun	32%	54%	13%	13%	27%	47%	27%			73%
sit	61%	54%		51%	28%	21%	64%			36%
stand	67%	43%		64%	19%	17%	61%			39%
swing_baseball	44%	40%	17%	81%	2%		81%			19%
throw	24%	48%	57%	11%	17%	15%	37%			63%
walk	72%	49%	32%	39%	10%	20%	80%			20%
wave	46%	55%	2%	86%	2%	10%	19%	7%		74%

Table 6.2: The meta-label statistics by action class for the J-HMDB (Split 1) dataset. The column "Camera Motion" represents the percentage of the videos for a certain class,e.g golf, that has the camera motion whereas the column "video quality" represents the average video quality where the good quality being '1'. The column "camera view-point" represents the percentage of the video by shot camera angle likewise the "Visible body part" column by four type.

Video settings	region	local	fused
Camera view-point variation	-0.25	-0.14	-0.29
Camera motion presence	-0.21	-0.23	-0.17
Visible body part variation	-0.31	-0.06	-0.15
Video quality	0.43	0.55	0.54

Table 6.3: The different video settings correlation with the performance

shows the improvement of 7%, 31 % in comparison to using regional (50.1%) and local feature (26.2%) alone.

Next, we analysis the results using the meta-data provided with the J-HMDB dataset as shown in Table 6.2. We attempt to quantify the impact of the environmental settings, i.e., camera motion, actor/action visibility, on the action localisation performance. Based on Table 6.2, the following settings have been analysed using the statistical correlation technique against the system performance: camera view-point variation, camera motion presence, visible body variation and video quality. Table 6.3 shows the correlation result for the different combination of features used. Interestingly, the video quality has the highest correlation (positive) with the performance compared to the other settings. Although the regional (RCNN) feature is highly discriminative, it is influenced by view variation (-0.25) and body part variation (-0.31) in comparison with the local features (-0.14, 0.06). In particular, the body part variation has no influence on the local feature. It can be explained by that fact that local features are extracted on the specific point than the whole region. The fusing method is shown to be effective for suppressing the effect.

For comparison with the-state-of-art methods, recently two methods have evaluated their system MAP performance averaged over all three splits with IOU parameter $\sigma = 0.5$. As shown at column 1 of Table 6.5, our method achieves competitive MAP performance of 56.30 %.

	fusion parameter α						
	0.8	0.75	0.7	0.6	0.5	0.3	0.1
J-HMDB (Split 1)	51.13%	54.16%	56.42%	**58.07%**	57.84%	56.80%	55.15%
J-HMDB (Split 2)	46.93%	51.48%	54.15%	**56.01%**	55.96%	54.85%	53.52%
J-HMDB (Split 3)	47.49%	50.72%	52.35%	53.69%	**54.20%**	51.97%	51.12%
Average	48.52%	52.12%	54.31%	55.92%	**56.00%**	54.54%	53.26%

Table 6.4: The effect of feature fusing

J-HMDB ($\sigma = 0.5$)		UCF-Sports ($\sigma = 0.5$)	
Action Tube [40]	53.3 %	Action Tube [40]	75.8 %
STMH [111]	60.7 %	STMH [111]	90.5 %
Our method	56.3 %	*Our method*	88.7 %

Table 6.5: Comparison of the method with the state-of-the-art methods

6.7 Conclusion

We propose a novel video graph-based framework for human action localisation from video sequences. The proposed approach can effectively accommodate different types of feature using the late fusion method. Also, the additional cues such as colour, motion and the geometrical information are captured within the graph representation. We perform the action localisation by maximising the score associated with the node and the edge in the video graph. In summary, this chapter demonstrates that the video graph representation is a flexible framework for both action recognition but also action localization and achieves very competitive performance to works that tackle both of these tasks independently.

Chapter 7

Conclusion

7.1 Overview

This chapter summarises the work presented and the conclusions drawn. In this thesis, we propose and investigate the effectiveness of the video graph representation for both action recognition and localisation. The video graph has been experimentally shown to be a discriminative representation to encode the underlying spatial and temporal structures in video and improve the recognition performance. In addition, the recognition framework for processing the video graph is designed and validated with a number benchmarking action datasets. The following section summarises the chapter-wise contributions. In section 7.3, the research questions formulated in the opening chapter are re-visited and addressed from a post-experimental perspective. The final section discusses future research directions and the possible extension of this work.

7.2 Thesis Summary

Chapter 3 provides a comprehensive investigation of a baseline recognition framework based on a Bag-of-Features representation. Many researchers have evaluated Bag-of-Features based action recognition. However, their main focus is on the evaluation of individual components in the framework or the framework within a limited experimental setting. In this chapter, this problem has been addressed with a comprehensive evaluation including several key elements in the

pipeline, i.e codebook dictionary construction, analysing the effect of kernel functions and the choice of local descriptors. In addition, the performance of the local descriptors under different degrees of static occlusion was analysed. In a real world, *uncontrolled* scenario, there are many chances that a human can be occluded by an object. However, it is unclear how the performance of the local descriptors are affected by the associated loss of information. In summary, the following conclusions have been drawn:

(i) Each component in the recognition pipeline plays a significant role. The performance of the recognition system can be increased, i.e., up to 30% with suitable components in the pipeline.

(ii) The choice of local descriptors highly influences the overall accuracy. The motion-based descriptors such as HOF and MBH are proven to be more discriminative in comparison to other descriptors in many datasets. Combining different descriptors is shown to be a straightforward and efficient technique to boost the performance.

(iii) In the presence of occlusion, the choice of local descriptors is critical for good performance. Results show that the MBH and its combination with TRAJ achieve the best performance in the presence of both heavy and partial occlusion.

Chapter 4 proposes a spatio-temporal graph-based human action recognition framework. In this chapter, we propose an action recognition framework based on the video graph that explicitly exploits spatio-temporal cues for action recognition. The chapter presents two contributions. First, we propose to extend the popular *dbscan* clustering algorithm to construct a video graph representation. This method has been shown to be not only effective but also produces an intuitive and sparse video graph. Second, the chapter explores the application of the Graph-Cut optimisation method from 2D image segmentation to 3D spatio-temporal volume

analysis to investigate its effectiveness for action recognition in video. Experimental works demonstrate that the proposed framework is not only an intuitive representation but also provides sizeable improvement when applied to the KTH and UCF-Sports datasets.

Chapter 5 introduces different techniques involved in the video graph construction process. Recently, many researchers in the action recognition domain have exploited the graph representation to capture spatio-temporal relationship. However, to the best of our knowledge, we do not find any studies about the impact of various graph construction methods. In this chapter, we answer an important question - to what extent the performance of the recognition system depends on the structural variations of the video graph representation. We evaluate the following graph construction techniques: AR, FG and SV. The experimental work shows that the graph construction techniques are highly influential for action recognition and there is an improvement in the performance by a margin of 3 % for the KTH dataset and 5 % - 10 % for UCF-Sports and J-HMDB.

Chapter 6 proposes an action localisation framework. In this approach, the action localisation task is formulated using maximum-path finding optimisation in the directed video-graph. This method has several important characteristics including the ability to accommodate a variety of feature types. To demonstrate this, we have used two different feature sets: one operates at a frame-level and the other at a temporal-level. In addition, the graph-based representation is generic and can use any graphical inference method for further improvement of the system. The chapter explores the application of the maximum-path finding algorithm to identify the localised action. The proposed method is proven to be effective and achieved results on a with the state-of-art methods such as Action Track [111] and Action Tube [40].

7.3 Research questions addressed

In this section, the research questions are revisited from a post-experimental perspective:

RQ1. **Does incorporating spatio-temporal cues in the video representation stage increase action recognition performance?**

This question is explored in Chapter 3 and Chapter 4. First, we establish the baseline system with the Bag-of-Features (BoF) model that captures no spatial and temporal cues in the video action recognition process. In Chapter 4, the video graph based action recognition is proposed. The recognition problem is formulated as an energy function with two components: E_1 and E_2. The likelihood energy, E_1, is discriminative score calculated using the standard Bag-of-Features (BoF) model which ignores the spatio-temporal cue. In contrast, the prior energy E_2 models the spatial as well as a temporal relationship between local features concerning the neighbourhood agreement. To evaluate the effectiveness of the spatial-temporal cues, we adopt the identical learned SVM model for both baseline and the proposed approach. The evaluation is performed with several datasets of varying levels of complexity. The experimental works have shown promising results across the various datasets. Also, a statistical analysis is performed to investigate the robustness under different environmental factors. It is found that the approach effectively deals with the various environmental changes i.e. camera motion, the visible body variation as long as there are smooth changes in the scene. However, in a scenario where there is a discontinuity in the spatio-temporal relationship (shot boundaries, view-point variation changes, etc.), the video graph model fails to capture the scene accurately and consequently it causes a drop in the performance. In conclusion, spatio-temporal cues are a discriminative part for the improvement of the action recognition system.

RQ2. **Can a graph-based video representation provide an effective method for incorporating spatio-temporal cues?**

The investigation of this research question is reflected in contributions from several chapters of the thesis. In chapter 3, the effectiveness of the graph representation for encoding spatio-temporal cues is highlighted. In particular, the edge of the video graph captures the relationship between local features and graph-cut optimisation is used to exploit this structural information in the recognition process. The resulting structure showed promising performance with several datasets. In particular, the method achieves the state-of-the-art performance in KTH and UCF-Sports datasets. Chapter 5 shows the versatility of the video graph representation. In practice, there are numerous ways to construct a video graph, and appropriate selection has the potential to improve recognition accuracy. For instance, the segmentation-based technique (SV) improved performance by a margin of 2% - 10 % with various benchmarking action datasets. Finally, in Chapter 6, the video graph representation is proposed for action localisation problem. Also, the resulting spatio-temporal representation can accommodate a different type of features. In summary, the spatio-temporal graph representation is intuitive structure to capture spatio-temporal cue that is typically ignored by many popular methods. It is a generic representation, and any graphical inference methods can be used. Also, the spatio-temporal representation can be extended to not only address the recognition task but also localisation.

RQ3. **What is the most effective technique for constructing the video graph ?**

In literature, there are three types of graph construction methods: clustering, segmentation and grid-based. Chapter 5 evaluates the graph construction techniques belonging to each of these categories regarding

accuracy, parameter sensitivity and computational complexity. Regarding accuracy, the segmentation-based approach (SV) is best followed by clustering (AR) and grid-based (FG). However, the accuracy of this method is highly sensitive, and this method suffers due to computational inefficiency. Considering all the factors, the clustering method (AR) proves to be the best choice. From the experiments, the general observation is that the performance of a technique is influenced by the condition in which the video data is collected such as camera-motion, view-point changes and shot boundary. For instance, the grid-based approach (FG) is suitable for the stationary camera view-point whereas the segmentation-based method (SV) is good for handling scenarios where there are sudden changes. In summary, based on the three datasets, the clustering- and segmentation-based graph construction methods are most effective.

RQ4. **Can the video graph be further improved to address the challenging problem of action localisation?**

We address this research question in Chapter 6. In this chapter, we introduce a novel directed video graph suited for the localisation task. In this graph, the node accommodates the different types of feature (local and the region-based) and furthermore its edge encodes the additional cues such as region colour, motion and region geometry in comparison to the graph representation presented in Chapter 4. The action localisation is performed as a maximum path finding problem on this directed video graph. The effectiveness of the approach is evaluated with several benchmarking datasets. In particular, we perform the comprehensive analysis using the meta-data provided with the J-HMDB dataset. The key finding is that the node discriminative score calculation based on the complementary set of features prove to be an effective method and it improved the performance by 7-31 % regarding MAP (mean average precision). For comparison with the

state-of-the-art methods, the experimental results show that a graph representation is sufficiently discriminative to achieve performance comparable with the state-of-the-art techniques. In summary, the graph representation is highly versatile and it can be adapted to solve a particular domain problem efficiently.

7.4 Future Work

In this section, several different directions for potential future work that can be extended from this thesis are presented:

- **Exploring alternative graphical inference methods:** In this thesis, most of the focus has been given to efficient video graph construction methods. In particular, three different representation techniques have been evaluated. However, the important process of the proposed framework is the decision-making the stage and it was limited to the Graph-Cut based optimisation method. Future research can be carried out to explore other decision-making methods operating on the graph structure, which have the potential to boost the performance.

- **Feature Fusing:** In this thesis, we investigate human action recognition using local features based on the appearance and motion cues. The experimental results shown in Chapter 4 demonstrated that the efficient and simple method to improve the performance is the fusing method. Using the advantage of based-graph representation of the video, graph node weight can be modified to capture a richer representation of the video, e.g. incorporating audio.

- **Investigating Temporal Domain:** The research thesis objective was limited to the problem of classification and spatial-localisation. The framework is built on top of the flexible video graph containing spatial as

well as temporal information. Hence, the proposed framework has the potential to be applied to the action detection task, which involves not only classifying the action but localising it in time. Therefore, another future work direction can be an extension to address this problem.

Bibliography

[1] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

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

[3] M. A. R. Ahad, T. Ogata, J. K. Tan, H. Kim, and S. Ishikawa. Motion recognition approach to solve overwriting in complex actions. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*, pages 1–6. IEEE, 2008.

[4] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2189–2202, 2012.

[5] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):288–303, 2010.

[6] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *IEEE T-PAMI*, 32(2):288–303, 2010.

[7] J. G. Allen, R. Y. Xu, and J. S. Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, pages 3–7. Australian Computer Society, Inc., 2004.

[8] I. Atmosukarto, N. Ahuja, and B. Ghanem. Action recognition using discriminative structured trajectory groups. *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, (Figure 1):899–906, 2015.

[9] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris. Activities of daily living recognition using optimal trajectories from motion boundaries. *Journal of Ambient Intelligence and Smart Environments*, 7(6):817–834, 2015.

[10] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[11] L. Ballan, M. Bertini, A. Del Bimbo, L. Seidenari, and G. Serra. Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications*, 51(1):279–302, 2011.

[12] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1806–1819, 2011.

[13] 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*, volume 2, pages 1395–1402. IEEE, 2005.

[14] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[15] Boykov. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV 2001*.

[16] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[17] M. Bregonzio, S. Gong, and T. Xiang. Action recognition with cascaded feature selection and classification. In *Crime Detection and Prevention (ICDP 2009), 3rd International Conference on*, pages 1–6. IET, 2009.

[18] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *IEEE CVPR*, pages 1948–1955, 2009.

[19] W. Brendel and S. Todorovic. Activities as time series of human postures. In *European conference on computer vision*, pages 721–734. Springer, 2010.

[20] L. Campbell and A. Bobick. Recognition of human body motion using phase space constraints. In *IEEE ICCV*, pages 624–630, 1995.

[21] L. Cao, Z. Liu, and T. S. Huang. Cross-dataset action detection. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 1998–2005. IEEE, 2010.

[22] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[23] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 2013.

[24] C.-Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. In *CVPR 2012*.

[25] C. Y. Chen and K. Grauman. Efficient activity detection with max-subgraph search. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1274–1281, 2012.

[26] H.-S. Chen, H.-T. Chen, Y.-W. Chen, and S.-Y. Lee. Human action recognition using star skeleton. In *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pages 171–178. ACM, 2006.

[27] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *ICCV 2011*.

[28] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian. Towards fast, view-invariant human action recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[30] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *Computer Vision–ECCV 2006*, pages 428–441, 2006.

[31] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part i: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26(3):261–276, 2006.

[32] 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*, pages 65–72. IEEE, 2005.

[33] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[34] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 726–733. IEEE, 2003.

[35] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd 1996*.

[36] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *IEEE CVPR*, pages 1–8, 2008.

[37] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[38] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[39] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *IEEE T-PAMI*, 33(5):883–897, 2011.

[40] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768, 2015.

[41] A. Helal, D. J. Cook, and M. Schmalz. Smart home-based health platform for behavioral monitoring and alteration of diabetes patients. *Journal of diabetes science and technology*, 3(1):141–148, 2009.

[42] N. Ikizler and D. Forsyth. Searching video for complex activities with finite state models. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[43] M. Jain, J. Gemert, H. Jégou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 740–747, 2014.

[44] M. Jain, J. Van Gemert, H. Jegou, P. Bouthemy, and C. G. M. Snoek. Action localization with tubelets from motion. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 740–747, 2014.

[45] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3192–3199, 2013.

[46] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[47] T. Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, 19(4), 1999.

[48] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[49] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 166–173. IEEE, 2005.

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

[51] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE CVPR*, pages 2046–2053, 2010.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[54] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

[55] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV 2011*.

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

[57] 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 Computer Society Conference on*, pages 1–8, 2008.

[58] 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*, pages 1–8. IEEE, 2008.

[59] I. Laptev and P. Pérez. Retrieving actions in movies. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[60] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[61] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *Computer*

Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.

[62] Q. Le, W. Zou, S. Yeung, and A. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE CVPR*, pages 3361–3368, 2011.

[63] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *IEEE CVPR 2009*.

[64] J. Liu and M. Shah. Learning human actions via information maximization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[65] S. Ma. Action recognition and localization by hierarchical space-time segments. In *ICCV 2013*.

[66] J. Muncaster and Y. Ma. Activity recognition using dynamic bayesian networks with automatic state selection. In *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, pages 30–30. IEEE, 2007.

[67] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV 2010*.

[68] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pages 392–405. Springer, 2010.

[69] R. Polana and R. C. Nelson. Recognition of motion from temporal texture. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 129–134. IEEE, 1992.

[70] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[71] K. Prabhakar, S. Oh, P. Wang, G. D. Abowd, and J. M. Rehg. Temporal causality for the analysis of visual events. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1967–1974. IEEE, 2010.

[72] C. Rao and M. Shah. View-invariance in action recognition. In *IEEE CVPR*, volume 2, pages II–316, 2001.

[73] K. Rapantzikos, Y. Avrithis, and S. Kollias. Dense saliency-based spatiotemporal feature points for action recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1454–1461. IEEE, 2009.

[74] Raptis. Discovering discriminative action parts from mid-level video representations. In *CVPR 2012*.

[75] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1242–1249, 2012.

[76] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[77] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[78] N. Robertson and I. Reid. A general method for human activity recognition in video. *Computer Vision and Image Understanding*, 104(2):232–248, 2006.

[79] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360. ACM, 2007.

[80] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[81] H. Seo and P. Milanfar. Action recognition from one example. *IEEE T-PAMI*, 33(5):867–882, 2011.

[82] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 144–149. IEEE, 2005.

[83] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[84] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

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

[86] I. Sipiran and B. Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976, 2011.

[87] Y. Song, X. Feng, and P. Perona. Towards detection of human motion. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 810–817. IEEE, 2000.

[88] K. Soomro, H. Idrees, and M. Shah. Action Localization in Videos Through Context Walk. *Proceedings of the IEEE International Conference on Computer Vision*, pages 3280–3288, 2015.

[89] W. Sultani and I. Saleemi. Human action recognition across datasets by foreground-weighted histogram decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 764–771, 2014.

[90] J. Sun. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR 2009*.

[91] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *European conference on computer vision*, pages 16–29. Springer, 2006.

[92] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 900–906. IEEE, 2003.

[93] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. *Computer Vision–ECCV 2010*, pages 140–153, 2010.

[94] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2649, 2013.

[95] D. Tran, S. Member, J. Yuan, and D. Forsyth. Video Event Detection : From Subvolume Localization to Spatiotemporal Path Search. 36(2):404–416, 2014.

[96] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[97] M. Ullah, S. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, volume 10, pages 95–1, 2010.

[98] J. Van De Weijer and C. Schmid. Coloring local feature extraction. In *Computer Vision–ECCV 2006*, pages 334–348. Springer, 2006.

[99] J. van Gemert, M. Jain, E. Gati, and C. Snoek. Apt: Action localization proposals from dense trajectories. In *BMVC*, volume 2, page 4, 2015.

[100] S. Vijayanarasimhan and K. Grauman. Efficient region search for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1401–1408. IEEE, 2011.

[101] H. Wang, A. Klaser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *IEEE CVPR*, pages 3169–3176, 2011.

[102] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[103] H. Wang, C. Yuan, W. Hu, and C. Sun. Supervised class-specific dictionary learning for sparse modeling in action recognition. *Pattern Recognition*, 2012.

[104] L. Wang and H. Sahbi. Directed Acyclic Graph Kernels for Action Recognition. *The IEEE International Conference on Computer Vision (ICCV)*, pages 3168–3175, 2013.

[105] L. Wang and D. Suter. Informative shape representations for human action recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 2, pages 1266–1269. IEEE, 2006.

[106] Y. Wang, K. Huang, and T. Tan. Human activity recognition based on r transform. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[107] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323, 2011.

[108] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7. IEEE, 2007.

[109] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. *Computer Vision–ECCV 2010*, pages 635–648, 2010.

[110] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[111] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3164–3172, 2015.

[112] A. D. Wilson and A. F. Bobick. Parametric hidden markov models for gesture recognition. *IEEE transactions on pattern analysis and machine intelligence*, 21(9):884–900, 1999.

[113] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[114] S. Wu. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *ICCV 2011*.

[115] L. Xia and J. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2834–2841, 2013.

[116] G. Xu et al. Viewpoint insensitive action recognition using envelop shape. In *Computer Vision–ACCV 2007*, pages 477–486. Springer, 2007.

[117] R. Xu. Compositional structure learning for action understanding. *arXiv preprint arXiv:1410.5861*, 2014.

[118] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.

[119] X. Yang, C. Yi, L. Cao, and Y. Tian. Mediaccny at trecvid 2012: Surveillance event detection.

[120] A. Yilma and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *IEEE ICCV*, volume 1, pages 150–157, 2005.

[121] C.-P. Yu, D. Samaras, and G. J. Zelinsky. Modeling visual clutter perception using proto-object segmentation. *Journal of vision*, 14(7):4–4, 2014.

[122] Y. Yu and J. T. Chang. Shadow graphs and surface reconstruction. In *European Conference on Computer Vision*, pages 31–45. Springer, 2002.

[123] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.