Human Action Recognition using Differential Motion

By

GAURAV KUMAR YADAV

under guidance of

DR. AMIT SETHI

AND

DR. SRINIVASAN KRISHNASWAMY

Department of Electronics and Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

A dissertation submitted to the Indian Institute of Technology Guwahati in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY.

AUGUST 2018

I dedicate this thesis to my parents, teachers and friends who have always inspired me and stood by my side during tough times.

AUTHOR'S DECLARATION

I, Gaurav Kumar Yadav, declare that this thesis titled,"Human Action Recognition using Differential Motion" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has not been submitted previously for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

SIGNED: ..... DATE: .....

CERTIFICATE

This is to certify that the thesis entitled "**Human Action Recognition using Differential Motion**" submitted by **Gaurav Kumar Yadav (11610229)**, a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in my opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

SIGNED: ..... DATE: .....

Supervisor: Dr. Srinivasan Krishnaswamy
Assistant Professor,
Department of Electronics and Electrical Engineering,
Indian Institute of Technology Guwahati,
Guwahati-781039, Assam, India.

SIGNED: ..... DATE: .....

Co-Supervisor: Dr. Amit Sethi
Assistant Professor,
Department of Electrical Engineering,
Indian Institute of Technology Bombay,
Mumbai-400076, Maharashtra, India.

ACKNOWLEDGEMENT

I feel great privilege in expressing my deepest and most sincere gratitude to my supervisor Dr. Amit Sethi, for his excellent guidance throughout the course of this work. He was always supportive and helping hand. His kindness, dedication, friendly accessibility and attention to detail have been a great inspiration to me. My heartfelt thanks to him for the unlimited support and patience shown to me. I would particularly like to thank for all his help in patiently and carefully correcting all my manuscripts, reports and other scientific writing. I could not have imagined having a better advisor and mentor for my PhD training.

I would like to thank my doctoral commitee members Prof. Prabin Kumar Bora, Dr. M. K. Bhuyan, Prof. Rohit Sinha and Dr. V. Vijaya Saradhi for sparing their precious time to evaluate the progress my research work. Their suggestions have been valuable. I would also like to thank other faculty members for their kind help in my academic studies. I am also thankful to the technical and administrative staff of the department.

I would like to thank all my bachelor students, specially, Rahul Nallamothu, Prakhar Shukla, Rakhil Immidisetti, Vignesh Kothapalli and others for working with me. I also extend my gratitude and thanks to all my beloved friends at the IPCV laboratory. They have been around to provide useful suggestions, companionship and created a good research environment. I thank Samdarshi, Anurag Singh, Santosh Yadav, Sunil Kumar for their valuable suggestions to my research work. I would also like to thank Anmol Srivastava, Subir Dey, Venkatesh Varla, Nilakshi Yein, Himakshi Chodhury and others for their friendly support during my stay at IIT Guwahati.

Last but not the least, I would like to thank my family: my parents and to my sisters for supporting me throughout writing this thesis and my life in general.

Gaurav Kumar Yadav

Abstract

Analyzing video and getting information from it is a growing field in computer-vision. The differential feature of a video compared to an image is motion. A video is a very high dimensional data and contains information of the environment, which is important for applications such as navigation, surveillance and video indexing. Therefore, extracting a compact representation of a video is required which can be used for various applications. There are various methods for estimating the motion from a video. One of the popular methods for estimating motion is optical flow.

The main theme of this work is to separate motions of different objects, such as background and foreground, by computing higher order derivatives of the motion vector. We proposed a method to compute differential motion on optical flow, which captures the motion of moving objects with respect to their potentially non-stationary backgrounds. Two methods based on curl and divergence for computing differential motion are proposed. The curl of optical flow is used to determine the interest points in video and extract features around it. We demonstrate the robustness of the proposed curl-based detector under common video transformations. We also proposed a descriptor that captures the location of the interest point with respect to neighboring interest points, which contains important information that state-of-the-art descriptors do not capture. The combination of the proposed detector and descriptor in a bag-of-features action recognition framework tested competitively with state-of-the-art methods.

In the second method, divergence is used for computing differential motion maps. We project differential motion maps on Cartesian planes and interpolate them to a fixed size. Even after this projection, the dimension of the feature vector was very high.

Based on the insight that contributing features of multiple moving objects in a video are likely to amalgamate additively on our representation, we show that semi-supervised non-negative matrix factorization performs better than other techniques to reduce the feature dimension. We demonstrate that optical flow, which captures absolute motion, is inadequate to capture discriminating features due to its inability to capture relative motion between an object and its background when the camera is in motion. The method to capture differential motion proposed by us is more effective than optical flow.

Recently, Convolutional Neural Networks (CNN) have been producing state-of-the-art results in classifying images of objects, complex events, and scenes. We have also proposed a method to detect abnormal events for human group activities using CNN. Our main contribution is to develop a strategy that learns with very few videos by isolating the action and by using supervised learning. First, we subtract the background of each frame by modeling each pixel as a mixture of Gaussians(MoG) to concatenate the higher order learning only on the foreground. Next, features are extracted from each frame using a convolutional neural network (CNN) that is trained to classify between normal and abnormal frames. These feature vectors are fed into long short term memory (LSTM) network to learn the long-term dependencies between frames. The LSTM is also trained to classify abnormal frames, while extracting the temporal features of the frames. Finally, we classify the frames as abnormal or normal depending on the output of a linear SVM, whose inputs are the features computed by the LSTM.

Evaluation of proposed methods is done on the popular benchmark datasets for action recognition task.

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

CHAPTER 1

INTRODUCTION

In this chapter, we introduce the topic of this Ph.D. thesis, action recognition in videos (Section 1.1). We present the motivation of our work (Section 1.2). Then, we describe research challenges related to action recognition (Section 1.3), and we present our main contributions in this topic (Section 1.4). Finally, we close this chapter with the thesis structure (Section 1.5) and list of publications (Section 1.6).

1.1 Introduction

Nowadays video capturing has become very easy due to advancements in camera technology. Miniaturization of cameras has started a new trend of capturing video from mobile phones, laptops, tablets and other hand-held devices, leading to an exponential increase in video data over the last ten years. According to recent YouTube® statistics, 300 hours of videos are uploaded every minute.[1] Automated video analysis will help in better utilizing large video databases for search and retrieval. Currently, only text tags in video meta-data are used for finding a video of interest in large video databases, which makes the search difficult in situations where the text tag is too broad or too narrow.

[1] http://www.statisticbrain.com/youtube-statistics

Since it is very tedious to annotate videos, automated classification and annotation techniques are required to manage and use large video databases. Human action classes are of primary interest in organizing and searching these databases. Therefore, action recognition has become a popular research topic over the last decade because of its application in video retrieval, video surveillance, healthcare, human-computer interaction and entertainment industry.

1.1.1 Video Retrieval

With the huge amount of video data uploaded on the Internet with every second, and with such widespread popularity of watching videos and movies on the Internet, video understanding and action recognition systems have become extremely important and necessary for relevant video retrieval. There are many videos which capture the same scene or even but are uploaded with some modifications in format and compression ratio by different users, which makes direct pixel matching infeasible to detect copies. Examples of such videos on YouTube include the same movies or sports clips uploaded by different users. For example, Batman movie is very popular. This single movie has been uploaded by different users to cover 71000 hours [2]. This increases the storage required as well as the need to detect its copies. Similarly, a single sports event covered by multiple reporters can lead to a lot of redundant videos. Therefore, human action recognition is very much useful in case of video retrieval.

1.1.2 Video Surveillance

Video surveillance cameras are a part of our lives. Video surveillance is the process of monitoring people and objects of interest using video cameras. With each passing day, a growing number of people have daily contact with video surveillance systems. They have become normal and widely accepted by society. Video surveillance cameras are used almost everywhere– not only at airports, subways, train stations, bus terminals, shopping malls, banks, post offices, casinos, swimming pools, cinemas, parking lots, but also in

[2] http://youtube-trends.blogspot.in/2013/

the streets for traffic monitoring, and in a great many buildings, for intrusion detection and analysis of human behaviour. The number of possible examples of applications is enormous. Analysis of large datasets of surveillance footage or several live streams requires automated classification of human action.

1.1.3 Health Care

In the field of healthcare, it is difficult to watch patient through CCTV or in person. Many times a person has to sit with the patient to monitor. Through CCTV camera some of the issues can be addressed and work-load can be reduced, while also addressing privacy concerns of being constantly watched by a human observer. Therefore, human action recognition can provide solutions to these problems such patient falling or behaving abnormal and report to the authorities.

1.1.4 Human-Computer Interaction

Nowadays, video cameras are used not only for video surveillance, and not only in mobile phones and laptops, but they are also used for human-computer interaction and in the entertainment industry. Video cameras provide a natural and intuitive way of human communication with a device. They have become so cheap that with each passing day, a rapidly growing number of people have daily contact with them. An example of a popular video camera is the Microsoft Kinect sensor, which is available in millions of homes around the world. Kinect is a motion sensing input device that allows users to control and interact with it through a natural user interface, using gestures and spoken commands. Therefore, one of the most important aspects of this sensor is the recognition of gestures and short actions.

1.2 Motivation

For human action recognition in videos, the most interesting part is to find a compact feature from the video which can describe an event. Structures in images and videos

are not restricted to constant motion or appearance with respect to time. On the other hand, many interesting events in the video are characterized by strong variations in the pixel intensities or color along both spatial and temporal dimensions. Feature extraction can be done using the local approach or holistic approach. The local approach includes interest point detector and holistic approach takes whole frame or video as input.

In the spatial domain, the locations of the points with a significant local variation of image intensities are referred as "interest point". The same notion can be extended for the videos. Points with significant local variation in image intensities and motion, both in the spatial and temporal domain can be referred as interest points in videos. A descriptor is extracted around that interest point which captures the pattern of an event. Interest points are attractive due to their high information contents and relative stability with respect to perspective transformations of the data.

Interest points in video depend upon both spatial and temporal information. Detecting interest points in the video is an open problem. A few methods are present in literature survey [6], sufficient work has not been done yet. Traditional approach uses motion analysis to find interest point in a videos by computing optical flow [39, 116]. Other methods are there which computes gradients in both spatial and temporal direction to detect spatio-temporal variations [21, 56]. Our aim is to find the interest points in video which capture variation in both spatio-temporal directions. It should be scale, rotation and illumination invariant and also robust to noise.

In case of holistic approach, features are extracted globally. Based on color and shape information of the whole object or person features are formed.

Video interest points and descriptors can be used in surveillance, video indexing, video alignment etc. As the video data are increasing on the internet, finding similar kind of video is a big challenge. Video interest point can help in retrieving the video of similar kind. In surveillance field, the operator has to sit and continuously watch the video to find out the unusual event. Interesting event detection and human activity analysis can be done by finding interest point in videos. There are many more applications like pedestrian detection, pose estimation, tracking an object, driving autonomous vehicle

etc. in which video interest point can be used.

Motion information based human action recognition is done in state-of-the-art methods. Differential motion has not been explored much. Therefore, we explore differential motion information both locally and globally, to find out spatio-temporal interest points and compute a compact feature representation for human action recognition application.

1.3 Research challenges

There are various challenges in automated video analysis such as variations in environments, viewpoints and actor movement. Variations in environments are caused by changes in lighting conditions, moving background, occlusion, moving camera, and sensor noise. Videos of the same action class taken from different viewpoints have high intra-class variation. In an inappropriate feature space, such intra-class variation is comparable to inter-class variation. The problem is further intensified for classes of similar actions such as walking, jogging and running, which differ mostly in speed and stride length. To achieve real-time performance a system must be robust to such intra-class variations while picking up on inter-class variations of interest. Also, videos usually are very high dimensional data which causes computational as well as storage problems.

One way to meet these challenges is to detect a sparse set of pixel locations in a video whose local pixel intensity patterns are unique in their neighborhood. Use of such points, known as interest points, is very common in image analysis [69] for applications ranging from stereo matching to object recognition. The idea is to find matching pairs of points or their aggregate characteristics (such as bag-of-words [57]) between images using a local descriptor of intensity patterns centered at each interest point. A successful interest point detector must yield locally unique points that are robust to some of the unwanted scene variations described above. These points are characterized by discontinuities in pixel intensities as we move in any direction from their location, for example at corners, small circles, or textured points. Additionally, a successful descriptor must yield an invariant feature at corresponding interest points between two scenes, incorporating

qualities such as scale invariance.

With the advent of deep learning, the use of image interest points is fast dwindling, especially for object recognition. In a recent paper, Shi et al. [91] proposed a sequential deep trajectory descriptor with three streams CNN. However, deep learning requires specialized hardware, long training times, and a large number of labeled examples. The latter may not be available for action recognition in videos. Video interest points along with their descriptors can also be used for applications that require explicit point matching such as video alignment and structure-from-motion for which deep learning by itself may not give a complete solution.

Not many interest point detectors have been proposed for videos as compared to images. Most of the proposed video interest point detectors are 3-dimensional (3-D) extensions of their 2-D (image) counterparts [15, 21, 56]. Such detectors yield interest points that are too sparse because of lack of sharp temporal discontinuities due to the persistence of objects. Additionally, these detectors do not filter static background which appears moving due to camera motion. Concentrating only on the moving foreground is important for action recognition.

For video representation using interest points, the so called bag-of-features model is popular due to its ability to yield a fixed dimensional representation for videos of different frame sizes and lengths [76]. Such a representation is built around a normalized histogram of descriptor prototypes for various interest points detected in a video. Obviously, such a representation lacks information about the relative locations of the interest points detected in a video.

1.4 Main Contributions

The salient contributions of the thesis are as follows:

- Examines the special characteristics of the temporal dimension of videos.
- Proposes a more appropriate treatment of the temporal dimension for detecting video interest points by using vector calculus on optical flow.

- Demonstrates that the proposed method is more robust to common video transformations in a video copy search setting-scale and compression ratio changes- when compared to other methods.

- Demonstrates that the proposed interest point detector gives improved action recognition performance on a popular benchmark dataset by combining it with local descriptors in a bag-of-features framework.

- Proposes a representation of video based on the differential motion maps for classification of actions that implicitly models a video as a non-negative mixture of various activities.

- Proposes that using a semi-supervised and linear mixture modeling based dimension reduction method is better suited to capture discriminative features in the differential motion maps. We test this by comparing different popular dimension reduction methods.

- Proposes that a classifier that is based on collaborative filtering such as l_2-regularized collaborative classifier (LRCC) is better suited for differential motion maps, which we demonstrate by comparing it with different popular classifiers. Our classification accuracy on KTH [89] and UCF11 [64] datasets is competitive with the state-of-the-art methods.

- Proposes a method to detect abnormal events for human group activities using the Convolutional neural network(CNN) and Long-short term memory(LSTM). Our main contribution is to develop a strategy that learns with very few videos by isolating the action and by using supervised learning.

1.5 Thesis Structure

The contents of the thesis are organized as follows: In **Chapter 1**, introduction to human action recognition and motivation has been discussed. In **Chapter 2**, datasets used

and state-of-the-art methods have been discussed. In **Chapter 3**, motion analysis in videos and prior work at treating temporal dimension differently has been discussed. An interest point detector based on differential motion is discussed for human action recognition in **Chapter 4**. We have also proposed a trajectory based descriptor and location based descriptor for the proposed interest point detector. In **Chapter 5**, we have proposed a compact video representation based on differential maps for action recognition. In **Chapter 6**, we have used CNN and LSTM for abnormal event detection. In **Chapter 7**, we summarize the thesis, its key contributions, limitations and scope for future work.

1.6 List of Publication

1.6.1 Referred Conferences

1. Vignesh, **Gaurav Kumar Yadav**, and Amit Sethi. "Abnormal Event Detection on BMTT-PETS 2017 Surveillance Challenge." In Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on (pp. 2161-2168). IEEE.

2. **Gaurav Kumar Yadav** and Amit Sethi. "Action Recognition using Spatio-Temporal Differential Motion." In Image Processing (ICIP), 2017 IEEE International Conference on (pp. 3415-34199). IEEE

3. **Gaurav Kumar Yadav**, Prakhar Shukla, and Amit Sethi. "Action Recognition using Interest Points Capturing Differential Motion Information." In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on (pp. 1881-1885). IEEE.

4. **Gaurav Kumar Yadav**, and Amit Sethi. "A Flow-based Interest Point Detector for Action Recognition in Videos." In Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing (p. 41). ACM.

5. Nallamothu, Vineeth, **Gaurav Kumar Yadav**, Amit Sethi, and T. Jacob. "Interest Point Detection in Videos Using Long Point Trajectories." In Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing (p. 70). ACM.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we summarize various human action recognition methods. A broad class of human action recognition techniques requires two-step i.e. representation and classification. Feature representation can be classified into three categories holistic approaches, local approaches and learning based approaches as described in Sections 2.2.1, 2.2.2 and 2.2.3. We have discussed various methods in this chapter for human action recognition. We first start with a brief overview of datasets used to evaluate human action recognition in Section 2.1.

2.1 Datasets

In this section, we review the most common datasets. We have also used these datasets for experiments.

2.1.1 Weizmann

The Weizmann dataset of human actions was proposed by Blank et al. [33], which is a simple and small dataset. It contains 10 different activities, namely walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack and

skip, each of which is performed by 9 or 10 subjects. The main characteristics of this dataset are the static and relatively uniform background, and the static position of the camera.

2.1.2 KTH

KTH is a simple action database which consists of 6 actions – boxing, hand-clapping, hand-waving, jogging, running and walking [89]. Each action was performed four times by 25 subjects in different scenarios such as outdoors and indoors, with different scale and clothing, resulting in 100 videos per action. Few samples are shown in Fig. 2.1.

Figure 2.1: Example frames of KTH dataset actions.

Figure 2.2: Example frames of UCF11 dataset actions.

2.1.3 UCF11

UCF11 is a complex action database which consists of 11 actions – basketball shooting, biking, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog [64]. This dataset has variations in camera motion, object appearance and pose, object scale, viewpoint, background clutter, illumination conditions, etc. For each class, the videos were grouped into 25 groups with at least 4 clips in each group. The video clips in the same group share some common features such as the same actor, background, or viewpoint. Few samples are shown in Fig. 2.2.

2.1.4 UCF101

UCF101 is an action recognition dataset of realistic action videos, collected from YouTube [96], having 101 action categories. This dataset is an extension of UCF50 dataset which has 50 action categories.

With 13,320 videos from 101 action categories, UCF101 gives the largest diversity in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc, it is the most challenging data set to date. As most of the available action recognition data sets are not realistic and are staged by actors, UCF101 aims to encourage further research into action recognition by learning and exploring new realistic action categories.

The videos in 101 action categories are grouped into 25 groups, where each group can consist of four to seven videos of an action. The videos from the same group may share some common features, such as similar background, similar viewpoint, etc.

The action categories for UCF101 data set are: Apply Eye Makeup, Apply Lipstick, Archery, Baby Crawling, Balance Beam, Band Marching, Baseball Pitch, Basketball Shooting, Basketball Dunk, Bench Press, Biking, Billiards Shot, Blow Dry Hair, Blowing Candles, Body Weight Squats, Bowling, Boxing Punching Bag, Boxing Speed Bag, Breaststroke, Brushing Teeth, Clean and Jerk, Cliff Diving, Cricket Bowling, Cricket Shot, Cutting In Kitchen, Diving, Drumming, Fencing, Field Hockey Penalty, Floor Gymnastics, Frisbee Catch, Front Crawl, Golf Swing, Haircut, Hammer Throw, Hammering, Handstand Pushups, Handstand Walking, Head Massage, High Jump, Horse Race, Horse Riding, Hula Hoop, Ice Dancing, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Knitting, Long Jump, Lunges, Military Parade, Mixing Batter, Mopping Floor, Nun chucks, Parallel Bars, Pizza Tossing, Playing Guitar, Playing Piano, Playing Tabla, Playing Violin, Playing Cello, Playing Daf, Playing Dhol, Playing Flute, Playing Sitar, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rafting, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Shaving Beard, Shotput, Skate Boarding, Skiing, Skijet, Sky Diving, Soccer Juggling, Soccer Penalty, Still Rings, Sumo Wrestling, Surfing, Swing, Table Tennis Shot, Tai Chi, Tennis Swing, Throw Discus, Trampoline

Figure 2.3: Example frames of UCF101 dataset actions.

Jumping, Typing, Uneven Bars, Volleyball Spiking, Walking with a dog, Wall Pushups, Writing On Board, Yo Yo. Few samples are shown in Fig. 2.3.

2.2 Representation

For video analysis feature extraction or representation is very important. The performance of the human action recognition methods mainly depends on the appropriate and efficient representation of data. Feature extraction or representation can be divided into two categories: Hand-crafted features and Learning-based approaches. The Hand-crafted

feature can be further divided into two categories: Holistic approach and Local approach. In Holistic approach, features are extracted based on global information present in the scene whereas in local approach various local features are extracted based on the region of interest and combined together to get a single feature. The main drawback of holistic approach is that they require more processing such as background subtraction or tracking to remove unwanted region also they are more sensitive to noise such as imperfect imaging techniques, variations in illumination, compression artifacts, etc. Local features are neither sensitive to noise nor they require extra preprocessing. The important features from the sequence of images and frames are extracted and feature descriptor is formed. Then classification is performed by training generic classifiers such as Support Vector Machine (SVM). Learning-based approaches have the capability to learn the feature automatically from the raw data, thus introducing the concept of end-to-end learning which means transformation from pixel level to action classification. Some of these approaches are based on dictionary learning while others employ deep learning-based models for action representation.

2.2.1 Holistic Approach

In this section, we discuss the approaches based on shape, appearance or motion for action recognition. The shape-based approaches [62, 100] use contour-based features for action representation and motion-based approaches [25, 28] use optical flow or its variants for action representations. There are also hybrid approaches which combine both shape and motion features. Now we will discuss each in detail.

2.2.1.1 Shape-based approaches

The shape-based approaches capture the local shape features from the human silhouette [111]. These methods first obtained the foreground silhouette from an image frame using foreground segmentation techniques. Then, they extract the features from the silhouette itself (positive space) or from the surrounding regions of the silhouette (negative space) between canvas and the human body [78]. Some of the important features

that can be extracted from the silhouette are contour points, region-based features, and geometric features. The region-based human action recognition method was proposed in [104]. This method divides the human silhouette into a fixed number of grids and cells for action representation and used a hybrid classifier support vector machine and nearest neighbor (SVM-NN) for action recognition. An action recognition method was proposed using Symbolic Aggregate approximation (SAX) shapes [46]. In this method, a silhouette was transformed into time-series and these time-series were converted into an SAX vector for action representation followed by a random forest algorithm for action recognition. In [8], a pose-based view-invariant human action recognition method was proposed based on the contour points with a sequence of the multi-view key poses for action representation. An extension of this method was proposed in [9]. This method uses the contour points of the human silhouette and radial scheme for action representation and support vector machine as a classifier. In [78], and [79] a region-based descriptor for human action representation was developed by extracting features from the surrounding regions (negative space) of the human silhouette. Another method used pose information for action recognition [12]. In this method, first, the scale invariant features were extracted from the silhouette, and then these features were clustered to build the key poses. Finally, the classification was performed using a weighted voting scheme.

2.2.1.2 Motion-based approaches

Motion-based action recognition approaches use the motion features for action representation followed by a generic classifier for action recognition. A novel motion descriptor was proposed in [16] for multi-view action representation. This motion descriptor is based on motion direction and histogram of motion intensity followed by the support vector machine for classification. Another method based on 2D motion templates using motion history images and histogram of oriented gradients was proposed in [72]. In [51], action recognition method was proposed based on the key elements of motion encoding and local changes in motion direction encoded with the bag-of-words technique.

2.2.1.3 Hybrid approach

These approaches combine shape-based and motion-based features for action representation. An optical flow and silhouette-based shape features were used for view invariant action recognition in [1] followed by principal component analysis (PCA) for reducing the dimensionality of the data. Some other methods based on shape and motion information were proposed for action recognition in [75, 105]. The coarse silhouette features, radial grid-based features and motion features were used for multi-view action recognition in [75]. Meanwhile, [45] used shape-motion prototype trees for human action recognition. The authors represented the action as a sequence of prototypes in shape-motion space and used distance measure for sequence matching. This method was tested on five public datasets and achieved state-of-the-art results. In [26], the authors proposed a method based on action key poses as a variant of Motion Energy Images (MEI), and Motion History Images (MHI) for action representation followed by the simple nearest-neighbor classifier for action recognition.

2.2.2 Local Approach

Local approaches are based on local patterns and structures. A number of local features are extracted and combined using a codebook or dictionary to get a single representation. This is divided into two-step: First is detector which detects the region of interest and second is descriptor which extracts the feature from the region of interest.

2.2.2.1 Local detector

Interest points in the spatial domain, i.e. images, have been extensively explored and several robust image interest point detectors are now available. In this section, we describe various image interest point detectors and their 3-D extensions to videos. Then we will discuss why the 3-D extensions are not appropriate for videos, followed by surveying detectors that treat the temporal dimension differently.

The principal idea behind image interest points is to detect locations with a large

variation in image intensity in all directions. Moravec developed the first signal-based interest point detector [70]. It is based on an auto-correlation function of the signal. The grey value differences between a window and windows shifted in several directions are measured. Four discrete shifts in directions parallel to the rows and columns of the image are used. An interest point is detected if the minimum of these four differences is superior to a threshold. Below, we review some important image interest point detectors in roughly chronological order.

1. **Hessian-based:** Beaudet [3] used the second derivatives of the signal for computing the measure $D = I_{xx}I_{yy} - I_{xy}^2$, where $I(x, y)$ is the intensity surface of the image. D is the determinant of the Hessian matrix and is related to the Gaussian curvature of the signal. Points, where this measure is maximal are defined as interest points. This interest point location is invariant to rotation. Kitchen and Rosenfeld [49] proposed an interest point detector which is based on the curvature of planar curves. They looked for curvature maxima on isophotes of the signal. However, due to image noise an isophote can have a high curvature without corresponding to an interest point, for example on a region with almost uniform grey values. Therefore, the curvature is multiplied by the gradient magnitude of the image where non-maximum suppression is applied to the gradient magnitude before multiplication. An interest point was detected if the measure $K = (I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_xI_y)/(I_x^2 + I_y^2)$ was above a threshold. Dreschler and Nagel [23] computed the locations of local extrema of the determinant of the Hessian D. A location of maximum positive D can be matched with a location of extreme negative D, if the directions of the principal curvatures that have opposite signs are approximately aligned. The interest point is located between these two points at the zero crossing of D.

2. **Auto-correlation-based:** Several interest point detectors [30, 31, 35, 101] are based on the auto-correlation function. For example, Harris and Stephens [35] improved the approach of Moravec [70] by using the auto-correlation matrix. A Gaussian is used to weigh the derivatives inside the window. Interest points are

detected if the auto-correlation matrix had two significant eigenvalues. Forstner [30] used the auto-correlation matrix to classify image pixels into contour and interest points. Interest points are further classified into junctions or circular features by analyzing the local gradient field. Local statistics allow a blind estimate of signal-dependent noise variance for automatic selection of thresholds and image restoration. Tomasi and Kanade [101] motivated their approach in the context of tracking. A good feature is defined as one that can be tracked well. They showed that such a feature is present if the eigenvalues of the auto-correlation matrix are significant.

3. **Miscellaneous approaches:** Heitger et al. [84] developed an approach inspired by experiments on the biological visual system. They extract 1-D directional characteristics by convolving the image with orientation-selective Gabor-like filters. In order to obtain 2-D characteristics, they computed the first and second derivatives of the 1-D characteristic. Cooper, Venkatesh and Kitchen [17] first measured the contour direction locally and then computed image differences along the contour direction. Knowledge of the noise characteristics is used to determine whether the image differences along the contour direction are sufficient to indicate an interest point. Early jump-out tests allowed a fast computation of the image differences. The detector of Reisfeld [82] used the concept of symmetry. They computed a symmetry map which shows "symmetry strength" for each pixel. This symmetry is computed locally by looking at the magnitude and the direction of the derivatives of neighboring points. Points with high symmetry are selected as interest points. Smith and Brady [94] compared the brightness of each pixel in a circular mask to the center pixel to define an area that has a similar brightness to the center. Two dimensional features can be detected from the size, centroid and second moment of this area. The approach proposed by Laganiere [55] is based on a variant of the morphological closing operator which successively applies dilation and erosion with different structuring elements. Two closing operators and four structuring elements are used. The first closing operator is sensitive to vertical/horizontal

L-corners and the second to diagonal L-corners. The SIFT detector was proposed by David Lowe [67]. SIFT is scale and rotation invariant. Difference-of-Gaussian is computed at different scales. An interest point is selected if it has maxima or minima values at neighboring scales, which is thought of as a stability criterion.

4. **Video interest point detector:** Compared to images, much fewer interest point detectors have been proposed for videos. Even among the ones proposed, a large majority are 3-D extensions of their 2-D (image) counterparts. For example, Cheung and H amarneh [15] proposed n-SIFT, which is an n-dimensional generalization of 2-D SIFT. Similarly, Laptev [56] extended 2-D Harris corner detectors to a 3-D Harris corner detector. However, it has been observed that spatio-temporal corners are relatively rare occurrences. This results in overly sparse features and poor performance for many real-world applications [56]. The method by Dollar et al. [21] (henceforth, "Dollar") improved upon Laptev's method for human action recognition by relaxing the constraint to detect a corner in the temporal domain to get better results. However, they stopped short of articulating the nature of the difference between the spatial and temporal dimensions of a video. Mo-SIFT [14] is different from other methods as it treats spatial and temporal dimension differently. It computes SIFT image interest points on each frame and uses temporal information by retaining only those SIFT points that have high flow magnitudes. This strategy works well to find points on moving objects captured using a static camera. However, it fails to differentiate between foreground and background using relative motion when the camera is moving. Our proposed interest point detector not only treats temporal dimension differently from the spatial ones, but is also able to isolated points associated with objects in motion relative to their backgrounds even if the background itself appears to be moving.

5. **Video descriptor:** A descriptor is computed for a window centered at an interest point to capture its distinctive local appearance so that it can be matched to a corresponding interest point in another video. Successful matching requires the

descriptor to be robust to transformations and invariant to viewpoints. In this section, we discuss some state-of-the-art descriptors.

Dollar et al. [21] proposed the cuboid descriptor for its detector which is based on intensity histogram of a cuboid around the interest point. Laptev [56] proposed a descriptor based on histograms of orientations of intensity gradient and optical flow (HOG/HOF) which characterizes the appearance and local motion around the interest points. The volume around an interest point is subdivided into a grid of cells. For each cell, a 4-bin histogram of gradient orientations (HOG) and a 5-bin histogram of optical flow (HOF) is computed. Normalized histograms are concatenated into HOG and HOF.

HOG3D descriptor was introduced by Klaser , Marszalek , and Schmid [50]. It is a 3-D extension of SIFT descriptor [67] for videos. Histograms of 3-D gradient orientations are computed using an integral video representation. The neighborhood volume is divided into cells and then gradient histograms are concatenated and normalized to form the descriptor. Willems, Tuytelaars , and Van Gool [112] extended the image SURF descriptor to videos to propose extended-SURF (ESURF) descriptor, in which the volume around the interest point is divided into cells, and each cell is represented by a vector of the weighted sum of the Haar-wavelets.

2.2.3 Learning based methods

The performance of action recognition system mainly depends on the appropriate and efficient representation of data. Learning based approaches have the capability to learn the feature automatically from raw data. Learning based approach involves learning the representation from the input data. This can be divided into two categories: Dictionary-based learning and Deep-learning. Dictionary-based methods try to learn the sparse representation of data from input data whereas deep-learning uses a neural network to learn the best feature from input data.

2.2.3.1 Dictionary based learning

Dictionary learning is generally based on the sparse representation of input data. The concept of dictionary learning is similar to Bag-of-words model because both are representation learned from the large number of feature vectors. One way to get the sparse representation of input data is to learn over-complete basis. Guha and Ward [34] has studied three over-complete dictionary learning frameworks, where each descriptor was represented by a linear combination of a small number of dictionary elements for compact representation. A supervised dictionary learning based method was proposed based on the hierarchical descriptor in [109]. In [123, 124], authors learned the view specific dictionaries where each dictionary corresponds to one camera view. Dictionary learning also uses unsupervised learning. Zhu et. al [125], proposed an unsupervised approach which does not require label information for action recognition.

2.2.3.2 Deep learning based

Taking in feature from rough data may be more helpful owing to the fact that there are no best hand-crafted features for all datasets. Deep learning ended up being a standard in the scope of machine learning which is a way to take in different levels of depiction and reflection that can make sense of data, for instance, speech, pictures, and videos. These procedures have automated the segmentation and feature extraction process for pictures/videos. These strategies use trainable component extractors and computational models with various layers for movement depiction and recognition. In perspective of the investigation over on deep learning presented in [20], we have assembled the deep learning models into two classes: (1) generative/unsupervised models (e.g., Deep Belief Networks (DBNs), Deep Boltzmann machines (DBMs), Restricted Boltzmann Machines (RBMs), and regularized auto-encoders); (2) Discriminative/Supervised models (e.g., Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), and Convolutional Neural Networks (CNNs).

Target class labels are not required for generative/unsupervised deep learning models during the learning process. These models are specifically useful when labeled data are

relatively less or unavailable. Deep learning models have been investigated since the 1960s [42] but researchers have not paid attention towards these models. This was mainly due to the success of shallow models such as SVMs [18], and unavailability of the huge amount of data, required for training the deep models.

A remarkable surge in the history of deep models was triggered out by the work of G. Hinton [37] where the highly efficient DBN and training algorithm was introduced followed by the feature reduction technique [38]. The DBN was trained layer by layer using RBMs [95], the parameters learned during this unsupervised pre-training phase were fine-tuned in a supervised manner using back propagation. Since the introduction of this efficient model, there has been a lot of interest in applying deep learning models to different applications such as speech recognition, image classification, object recognition, and human action recognition.

A method using unsupervised feature learning from video data was proposed in [58] for action recognition. The authors used independent subspace analysis algorithm for learning spatio-temporal features and combined with deep learning techniques such as convolutional and staking for action representation and recognition. Deep Belief Networks (DBNs) trained with RBMs were used for human action recognition in [29]. The proposed approach outperforms the handcrafted learning-based approaches on two standard datasets. Learning continuously from the streaming video without any labels is an important but challenging task. This issue was addressed in [36] by using unsupervised deep learning model. Most of the action datasets have been recorded under controlled environment; action recognition from unconstrained videos is a challenging task. A method for human action recognition from unconstrained video sequences was proposed in [2] using DBNs.

Unsupervised learning played a pivotal role in reviving the interests of the researchers in deep learning. However, it has been overshadowed by the purely supervised learning since the major observing it rather being told the name of every object. The human and animal learning is mostly breakthrough in deep learning used CNNs for object recognition [54]. However, an important study by the pioneers of latest deep

learning models suggests that unsupervised learning is going unsupervised to be far more important than its supervised counterpart in the long run [59] since we discover the world by observing it rather being told the name of every object. The human and animal learning is mostly unsupervised.

Literatures [54, 120] report that the most frequently used type of deep learning model, which show excellent performance at tasks under the supervised category is Convolutional Neural Networks (CNN or Convnet) [60].This is a hierarchical learning model with multiple hidden layers to transform the input volume into output volume. Its architecture consists of three main types of layers: convolution layer, pooling layer, and fully-connected layer. Understanding the operation of different layers of CNN requires mapping these activities into pixel space. This is done with the help of Deconvolutional Networks (Deconvnets) [121]. The Deconvnets use the same process as CNN but in reverse order for mapping space to pixel space.

Most of the CNN are limited to 2D action recognition. This limitation was addressed by [44] by introducing both spatial and temporal dimensions at convolution layer to form a 3D CNN model. The application of this 3D CNN model displayed state-of-the-art result in airport video surveillance.

The supervised learning CNN model 2D or 3D can also be accompanied by some unsupervised endeavors. One of the unsupervised endeavors is slow feature analysis (SFA) [113], which extracts slowly varying features from the input signal in an unsupervised manner. Beside other recognition problems, it has proved to be effective for human action recognition as well [122]. In [97], two-layered SFA learning was combined with 3D CNN for automated action representation and recognition. This method achieved state-of-the-art results on three public datasets including KTH, UCF sports, and Hollywood2. Other types of supervised models include Recurrent Neural Networks (RNNs). A method using RNN was proposed in [24] for skeleton-based action recognition. The human skeleton was divided into five parts and then separately fed into five subnets. The results of these subnets were fused into the higher layers and final representation was fed into the single layer. For further detail regarding this model reader may refer to

[24]. The deep learning-based models for human action recognition require huge amount of video data for training. However, collecting and annotating huge amount of video data is immensely laborious and requires the huge computational resources. A remarkable success has been achieved in the domains of image classification, object recognition, speech recognition, and human action recognition using the standard 2D and 3D CNN models. However, still there exist some issues such as high computational complexity of training CNN kernels, and huge data requirements for training. To curtail these issues researchers have been working to come up with variations/adaptations of these models. In this direction, factorized spatio-temporal convolutional networks (FSTCN) was proposed in [98] for human action recognition. This network factorizes the standard 3D CNN model as a 2D spatial kernels at lower layers (spatial convolutional layers) based on the sequential learning process and 1D temporal kernels in the upper layers (temporal convolutional layers). This reduced the number of parameters to be learned by the network and thus reduced the computational complexity of the training CNN kernels. The detailed architecture is presented in [98]. Another approach using spatio-temporal features with a 3D convolutional network was proposed in [102] for human action recognition. The evaluation of this method on four public datasets confirmed three important findings: (1) 3D CNN is more suitable for spatio-temporal features than 2D CNN; (2) The CNN architecture with small $3 \times 3 \times 3$ kernels is the best choice for spatio-temporal features; (3) The proposed method with linear classifier outperforms the state-of-the-art methods.

2.3 Discussion

We have discussed the various datasets used in this work and presented a brief overview of state-of-the-art methods for human action recognition. We have divided these methods into three categories i.e. holistic approaches, local approaches and learning-based approaches. In this work, we have contributed to these three categories by proposing algorithms for human action recognition.

CHAPTER 3

ANALYSIS OF TEMPORAL DIMENSION

In this chapter, we show the special characteristics of temporal dimension both qualitatively and quantitatively. Additionally, we propose a trajectory based interest point detector which gives different treatments to temporal dimension and spatial dimension.

Figure 3.1: Example of tube like motion patterns in spatio-temporal domain [4].

3.1 Special characteristics of the temporal dimension

Video is a 3-dimensional data, which consists of height, width and time. Height and width are known as spatial dimensions and time is known as the temporal dimension.

Our hypothesis is that spatial and temporal dimensions are different from each other and should be treated differently. We show this both qualitatively and quantitatively.

3.1.1 Qualitative Analysis

In spatial dimensions objects have sharp boundaries due to their finite spatial shapes and sizes. However, objects mostly persist in the temporal dimension, forming tube-like shapes stretched in time. In other words, the temporal dimension is not equivalent to depth (i.e., t is unlike x, y, or even depth z) and it captures a continuous flow of the 2-D projection of an object in tube-shaped structures as shown in Fig. 3.1. The cross-section of the video object tube corresponds to its finite boundaries in each frame, while it persists in time for as long as it can be seen in the scene. The reason behind the poor performance of simple 3-D extensions of the 2-D interest point detectors in videos is most likely an inappropriate incorporation of temporal data in these detection methods. Because of object persistence, temporal discontinuities (sharp boundaries) are rare occurrences, and are usually associated with occlusion, revelation, entry, or exit of the object from the scene. Objects also move only to the extent allowed by laws of physics (e.g. finite velocity and acceleration). These properties of the temporal dimension automatically imply that temporal edges, and consequently spatio-temporal corners are rare, which explains their overly sparse occurrence [56]. Thus, to detect interesting motion, concentrating on some locally unique characteristics of the temporal flow itself is likely to yield better results. This observation points towards the use of optical flow, which was also used in MoSIFT [14].

3.1.2 Quantitative Analysis

Based on the observation that an object forms tubes stretching in the temporal dimension and lacks sharp temporal discontinuities, we anticipated that temporal slices will lack higher Fourier frequencies compared to the spatial slices. We computed the average magnitude of Fourier transform (FT) of pixel intensities for 1-D slices across height, width, and time dimensions by keeping the other two dimensions fixed. As shown in

Fig. 3.2, in semi-log scale representing exponential of the frequency, the slope of FT of the time dimension is much more negative than that of the height and width dimensions. That is, if we model the non-zero frequencies of FT as $kf^{-\alpha}$, where f is the spatial or temporal frequency, and α and k are positive constants, then α, which is the negative of slope in semi-log scale, is nearly twice as large for the temporal dimensions as compared to the spatial ones. Therefore, time dimension relatively lacks higher frequencies and its spectral power decays more rapidly with increase in frequency compared to the two spatial dimensions.

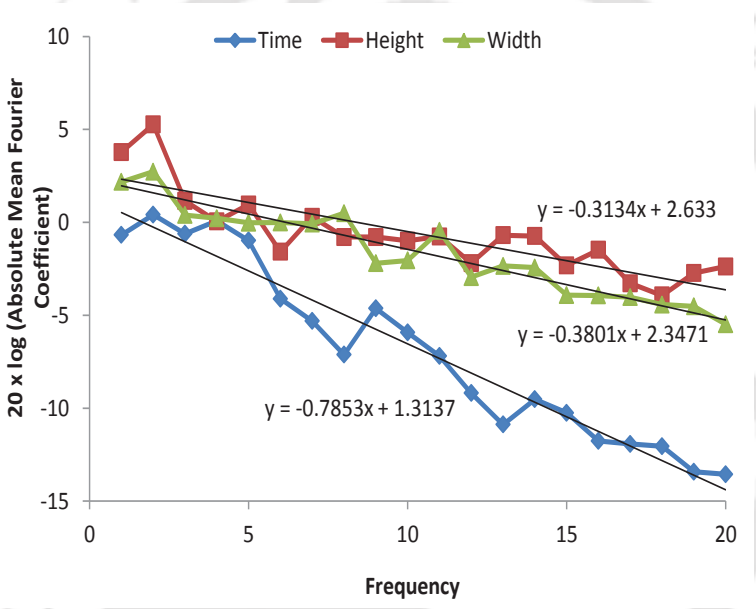


Figure 3.2: Log of mean absolute magnitudes of Fourier coefficients of spatial and temporal slices reveal relative predominance of lower frequencies in the temporal dimension.

3.2 Our early attempt at treating temporal dimension differently

Fig. 3.3 depicts the major components of the proposed system for interest point detection. The initial frame is sampled with a grid to generate candidate key points. The candidate key points are tracked using variational optical flow to generate long point trajectories.

The trajectories thus obtained are filtered to remove trajectories that may be deemed as not interesting. The remaining trajectories are subjected to a curve simplification algorithm to generate key points on the trajectories that are localized in the temporal dimension.

In the following subsections, we take a closer look at each of the components.

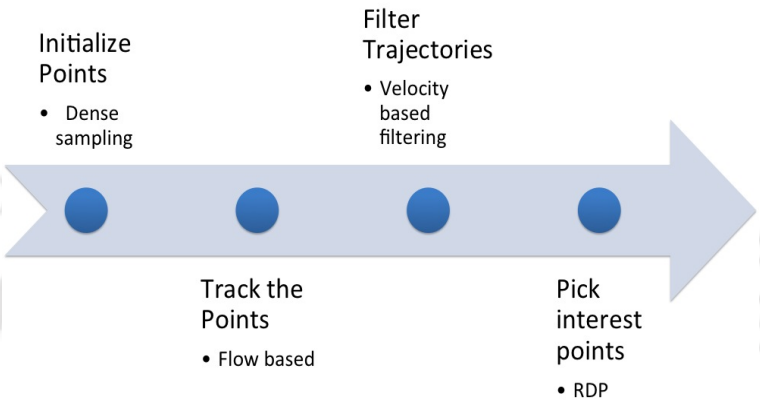


Figure 3.3: System diagram for interest point detection

3.2.1 Initial set of points

A grid of candidate key points is formed from the input frame by densely sampling the input frame at regular spatial intervals. In our experiments, we used sampling grids of size 2, 4 and 8 pixels to generate candidate interest points. We found through visual inspection that the trajectories obtained by tracking key points generated by sampling the input frame were of similar structure over a range of comparable grid sizes. Therefore, we used a fixed grid size of 8 pixels to sample the initial frame. Alternatively, the system can also be initialized with any particular set of points: For example, the candidate key points could be initialized according to the key points generated by SIFT.

3.2.2 Trajectory extraction

The candidate key points are tracked with the help of variational optical flow vectors. A filtering mechanism is used to remove trajectories formed by tracking of candidate key

points sampled from the background.

3.2.2.1 Tracking with Variational flow

We track the candidate key points from frame t to frame $t+1$ using its location in the t^{th} frame (x_t, y_t) and the flow[39] vectors computed at that position $f_t(x_t, y_t)$ to estimate the position of the key point (x_{t+1}, y_{t+1}) in the $(t+1)^{th}$ frame as shown in the Equation 3.1. To compute the flow, we use the state-of-the-art algorithm by Brox et. al[7] which uses descriptor matching of candidate key points between adjacent frames to improve the flow computations.

$$(x_{t+1}, y_{t+1}) = (x_t, y_t) + f_t(x_t, y_t) \tag{3.1}$$

A similar procedure is repeated for tracking key points in the $(t+2)^{th}$ frame. If the predicted position of a key point obtained by the computation earlier lies outside the boundary of the image, it indicates that the key point is likely to have gone out of the scene and it is no longer required to track the key point. The trajectory of a candidate key point consists of the sequential ordering of the predicted positions of the key point at every time instant within the video.

The images in Table 3.1 show, a grid of points being tracked through the video using optical flow. The trail left by each point indicates its trajectory.

3.2.2.2 Filtering the trajectories

A lot of the trajectories obtained through tracking of flow vectors are "not interesting" due to the presence of the background and noise in the video. Trajectories obtained from the background of the scene are typically not stationary due to slight camera motion and errors in computation of the optical flow vectors. Hence, we employ a velocity—based filtering scheme to exclude trajectories that are not interesting.

The instantaneous velocity of the key point at time t, $v_t(x_t, y_t)$ is defined as:

$$v_t(x_t, y_t) = |x_t - x_{t-1}| + |y_t - y_{t-1}| \tag{3.2}$$

We use this definition instead of the regular Euclidean velocity as it lowers the computational complexity without significantly changing the decision boundary. A lower threshold on the mean of the instantaneous velocity of the candidate key point along the length of the entire trajectory is used to remove trajectories with very little motion and an upper threshold on the maximum instantaneous velocity within a trajectory is used to remove trajectories corrupted by noise. The images in Table 3.2 show the trajectories of filtered points. It can be observed that our filtering approach removes almost all the background points. Fig. 3.4 shows the 3D plot of the trajectories after filtering. The Z axis is the time axis, X and Y axis represent the spatial coordinates.

3.2.3 Interest Point Detection

After filtering the trajectories by employing an instantaneous velocity—based thresholding, we use Ramer—Douglas—Peucker (RDP) algorithm to find key points that can be localized in time within the remaining trajectories. This is based on the intuition that any action can be broken up into certain key "moments" that can describe the entire trajectory of motion.

3.2.3.1 Ramer Douglas Peucker Algorithm

The Ramer—Douglas—Peucker algorithm (RDP) is an iterative algorithm that seeks to approximate a planar curve by a series of line segments as shown in Algorithm 1. This algorithm is also known under various names such as Douglas—Peucker algorithm [22], iterative end—point algorithm [80] and split—and—merge algorithm [83]. The algorithm has found applications in various areas such as vector graphics processing, cartographic generalization and robotics. ϵ is a parameter controls the curvature of the

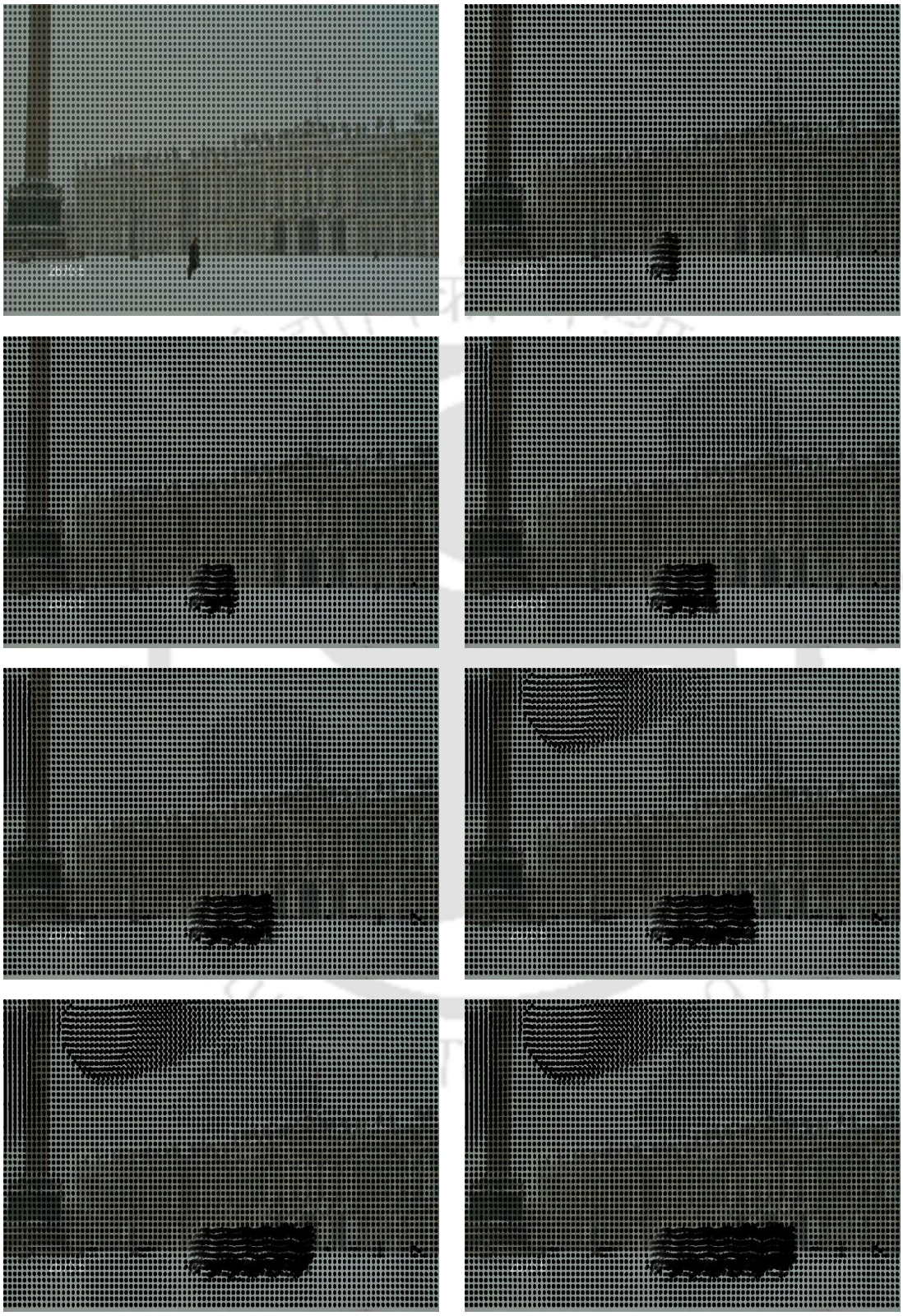


Table 3.1: Initial grid of points tracked through using Variational flow.

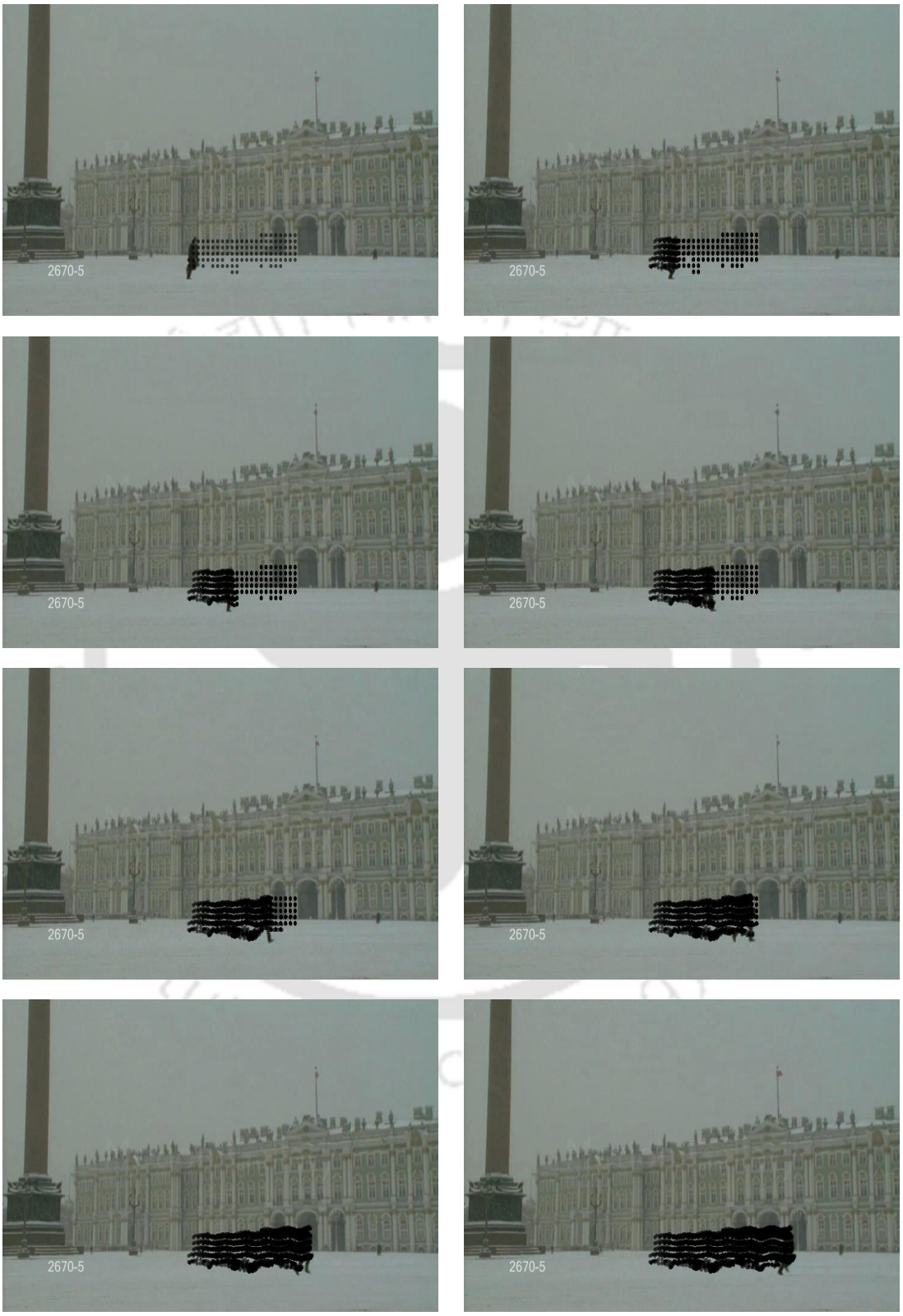


Table 3.2: Trails of the filtered points being tracked through the video

curve while approximating the curve by series of points. If the point is closer than a given distance ϵ, it removes all there in-between points.

Algorithm 1 Curve approximation by a series of line segments using Ramer—Douglas—Peucker algorithm

procedure RDP(X, ϵ)
  X := Ordered list of points that represents the trajectory
  n := Number of points in the trajectory
  a := Start point of the trajectory
  b := End point of the trajectory
  d := Array of distances of points in the trajectory from the line joining a and b
  for all $i := 1$ *to* n **do**
    $d[i] \leftarrow$ distance of i^{th} point in the trajectory from the line joining a and b
  end for

  $d_{max} \leftarrow$ maximum of the computed distances

  if $d_{max} \leq \epsilon$ **then**
    return (a, b)
  else
    c is the point on the trajectory that is the farthest from the line joining a and b, i.e. the point corresponding to d_{max}
    $left \leftarrow$ Left half of the trajectory from a to c
    $right \leftarrow$ Right half of the trajectory from c to b
    return RDP($left$, ϵ) $\bigcup$ RDP($right$, ϵ)
  end if
end procedure

A sample reconstruction of a three-dimensional curve using RDP is shown in Fig. 3.5.

As seen from the Fig. 3.5, the algorithm performs exceptionally well in preserving the shape of the trajectory with much fidelity.

3.2.4 Experiments and Results

A dataset was created for the purpose of evaluating the proposed system model. In order to test the invariance of interest points, scaled and compressed versions of the videos were added to the dataset. Finally, we perform a comparative study of our proposed system with Dollar's space-time features [21] and n-SIFT [15] against these transformations.

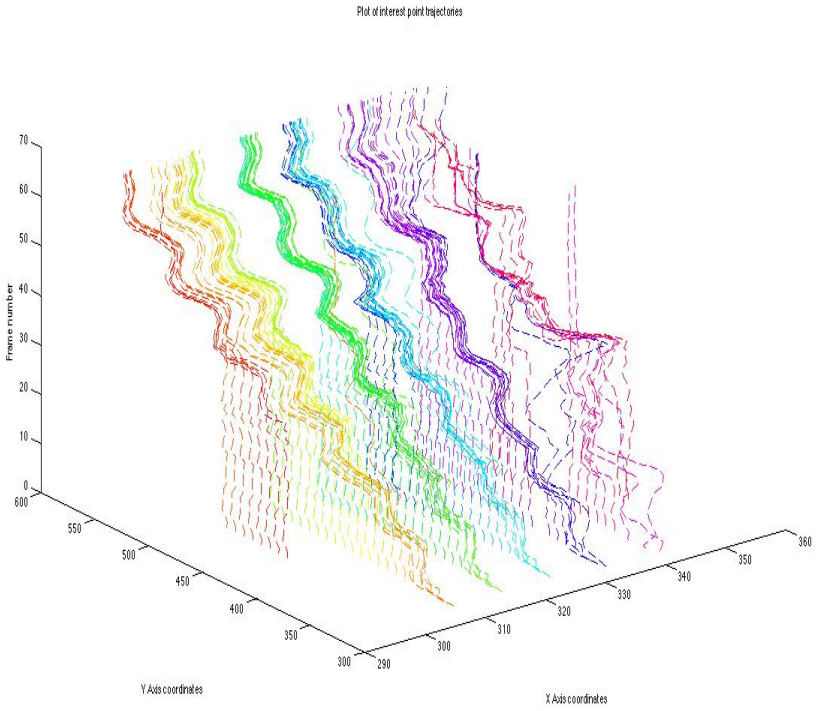


Figure 3.4: 3D plot of filtered trajectories

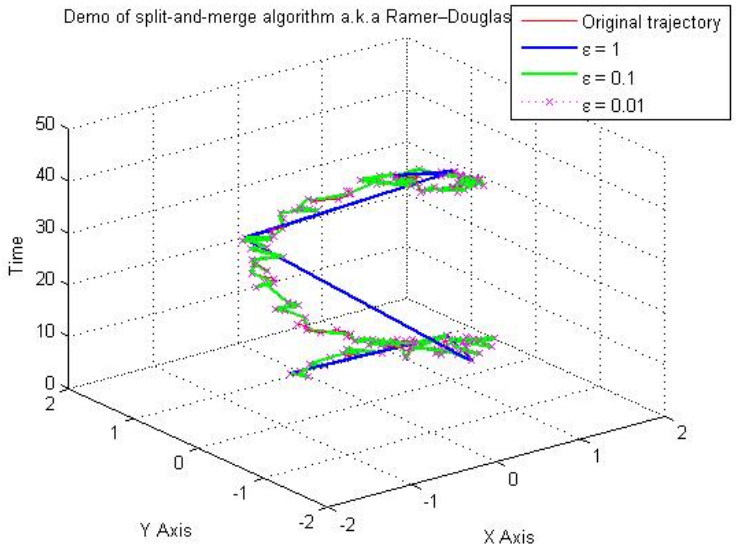


Figure 3.5: Trajectory reconstruction with RDP for varying values of ϵ

3.2.4.1 Evaluation dataset

We created an evaluation dataset of 100 videos by choosing videos primarily from the KTH action recognition dataset [89]. For every video taken from the KTH database, the compressed and scaled versions of the video were also included in the evaluation set in order to test the performance of our method.

3.2.4.2 Quantitative Analysis

RDP algorithm gives an ordered list of points (few points from each trajectory) which we consider as interesting. We use *repeatability* as the metric to measure the stability of these points. Repeatability between two sets of points A, B is computed as follows:

$$R(A,B) = \frac{numel(A \cap B))}{min(numel(A), numel(B))} \tag{3.3}$$

To find the intersection between the two sets, we count the number of unique matches. Unique match constraint enforces that a point in one set can be matched to at-most to one point in the other set.

3.2.4.3 Effect of scale

Videos in the dataset were scaled to 0.5, 0.75 and 1.5 times their original scale. The repeatability scores for each scale versus the value of epsilon have been listed in the Table 3.3.

There is no clear-cut trend that can be observed in the values of ϵ versus repeatability. In general, we expect repeatability to reduce with increase in ϵ, but it is not found to be true. Each scale has a particular ϵ value for which a high repeatability is achieved. Further analysis is required to examine the effect of ϵ on the repeatability of interest points in great detail.

Scale \ Epsilon	0.1	1	3	5
0.5	0.8765	**0.9393**	0.8543	0.7550
0.75	0.9024	0.9425	0.8977	**0.9649**
1.5	0.8978	**0.9724**	0.8495	0.9542

Table 3.3: Repeatability for different scales and epsilons

3.2.4.4 Effect of compression

Videos in the dataset were subjected to MPEG compressions with 150, 175, 200, 250, 275 and 300 compression ratios. Repeatability scores for different values of epsilon have been listed in the Table 3.4.

In general, as the value of epsilon increases, the repeatability decreases because larger epsilon results in fewer points being picked, thus reducing the probability of finding an exact match.

Compression ratio \ Epsilon	0.1	1	3	5
150	**0.8317**	0.8100	0.7549	0.7766
175	0.8183	**0.8249**	0.8183	0.7728
200	**0.8304**	0.8135	0.7999	0.7752
250	0.8428	**0.8458**	0.8193	0.7895
325	**0.8619**	0.8618	0.8366	0.8131

Table 3.4: Repeatability for different compression levels and epsilons

3.2.4.5 Comparative study

We compare the repeatability score of our algorithm with the implementations of Dollar's space time features [21] and n-SIFT [15]. The comparisons of the different methods with different values of ϵ s are listed in the Table 3.5 to Table 3.12.

Detector \ Scale	0.5	0.75	1.5
n-SIFT	0.6794	0.5720	0.4447
Dollar's STF	0.1739	0.3656	0.2274
Our Algorithm	**0.8765**	**0.9024**	**0.8978**

Table 3.5: Comparison of repeatability under scale changes with epsilon = 0.1

Detector \ Scale	0.5	0.75	1.5
n-SIFT	0.6794	0.5720	0.4447
Dollar's STF	0.1739	0.3656	0.2274
Our Algorithm	**0.9393**	**0.9425**	**0.9724**

Table 3.6: Comparison of repeatability under scale changes with epsilon = 1

Detector \ Scale	0.5	0.75	1.5
n-SIFT	0.6794	0.5720	0.4447
Dollar's STF	0.1739	0.3656	0.2274
Our Algorithm	**0.8543**	**0.8977**	**0.8495**

Table 3.7: Comparison of repeatability under scale changes with epsilon = 3

Detector \ Scale	0.5	0.75	1.5
n-SIFT	0.6794	0.5720	0.4447
Dollar's STF	0.1739	0.3656	0.2274
Our Algorithm	**0.7550**	**0.9649**	**0.9542**

Table 3.8: Comparison of repeatability under scale changes with epsilon = 5

Detector \ Compression ratio	150	175	200	250	325
n-SIFT	0.5225	0.5090	0.4979	0.4807	0.4525
Dollar's STF	0.2447	0.2319	0.2797	0.2870	0.1733
Our Algorithm	**0.8317**	**0.8183**	**0.8304**	**0.8428**	**0.8619**

Table 3.9: Comparison of repeatability under compression with epsilon = 0.1

Detector \ Compression ratio	150	175	200	250	325
n-SIFT	0.5225	0.5090	0.4979	0.4807	0.4525
Dollar's STF	0.2447	0.2319	0.2797	0.2870	0.1733
Our Algorithm	**0.8100**	**0.8249**	**0.8135**	**0.8458**	**0.8618**

Table 3.10: Comparison of repeatability under compression with epsilon = 1

Detector \ Compression ratio	150	175	200	250	325
n-SIFT	0.5225	0.5090	0.4979	0.4807	0.4525
Dollar's STF	0.2447	0.2319	0.2797	0.2870	0.1733
Our Algorithm	**0.7549**	**0.8183**	**0.7999**	**0.8193**	**0.8366**

Table 3.11: Comparison of repeatability under compression with epsilon = 3

Detector \ Compression ratio	150	175	200	250	325
n-SIFT	0.5225	0.5090	0.4979	0.4807	0.4525
Dollar's STF	0.2447	0.2319	0.2797	0.2870	0.1733
Our Algorithm	**0.7766**	**0.7728**	**0.7752**	**0.7895**	**0.8131**

Table 3.12: Comparison of repeatability under compression with epsilon = 5

3.2.5 Discussion

We have shown that giving temporal a dimension different treatment than spatial dimension gives better results. The repeatability of these interest points is quite high under various transformations such as scaling and compression, when compared to other popular methods such as Dollar's space-time features [21] and n-SIFT [15]. Therefore, a more suitable algorithm is required which can take account of continuity of objects in temporal dimension.

CHAPTER 4

ACTION RECOGNITION USING LOCAL FEATURES

Working with a sparse set of informative points – interest points – as opposed to all the pixel locations can reduce the processing requirements of automated image and video analysis software. We propose a new video interest point detector based on the following unmet needs. There is a simplifying assumption inherent in existing video interest point detectors (except MoSIFT [14]) that time and space dimensions are similar. This assumption does not take into account the fact that objects persist in time while they have sharp boundaries in space. Additionally, for videos shot using a moving camera existing video interest point detectors (including MoSIFT) are unable to restrict interest points to objects that are in relative motion with respect to the background. Our interest point detector solves these problems by using vector calculus on optical flow. It yields interest points that are robust to video common transformations and relevant to action recognition. Additionally, we suggest a way to improve the descriptor of video interest points by including information about their relative locations. In a bag-of-words video representation, there is no obvious way to incorporate the information about locations of neighboring interest points, which limits its action recognition accuracy. Using the combination of the proposed detector and descriptor in a bag-of-words video representation yielded high action recognition

accuracy on four publicly available datasets.

4.1 A flow-based interest point detector for action recognition in videos

In order to incorporate continuity of objects in the temporal dimension, the proposed method is based on optical flow, which is similar to the idea behind MoSIFT. The proposed method departs from MoSIFT in its use of vector calculus on optical flow to detect relative motion. Since, optical flow forms tube-like patterns, vector calculus is a natural choice to measure differential motion between objects (usually between target and background) due to its wide applications in fluid kinematics. We treat optical flow as a vector field for each frame. Our interest points are those points that have a curl [103] magnitude greater than a threshold. We also experimented with divergence, but the empirical results were not satisfactory. Fig. 4.2 shows that high divergence magnitudes were seen all along a moving edge. However, high curl magnitudes were seen for only a spatially isolated set of points where the direction of relative motion of an object is parallel to its boundary. For example, on a person moving sideways, these interest points will be near the top of head and bottom of feet as illustrated in Fig. 4.1. We should clarify that this curl is non-zero not because the object is rotating, as is the usual purpose of calculating curl in vector calculus. The curl is non-zero because a closed-loop line integral of optical flow around a boundary point with tangential motion (e.g., top of the head) gets unidirectional contribution from the moving side of the boundary (e.g., the head with tangential flow), and no contribution from the stationary side (e.g., background above the head with no flow). The locations of interest points for a moving object detected using the proposed method are thus predictable.

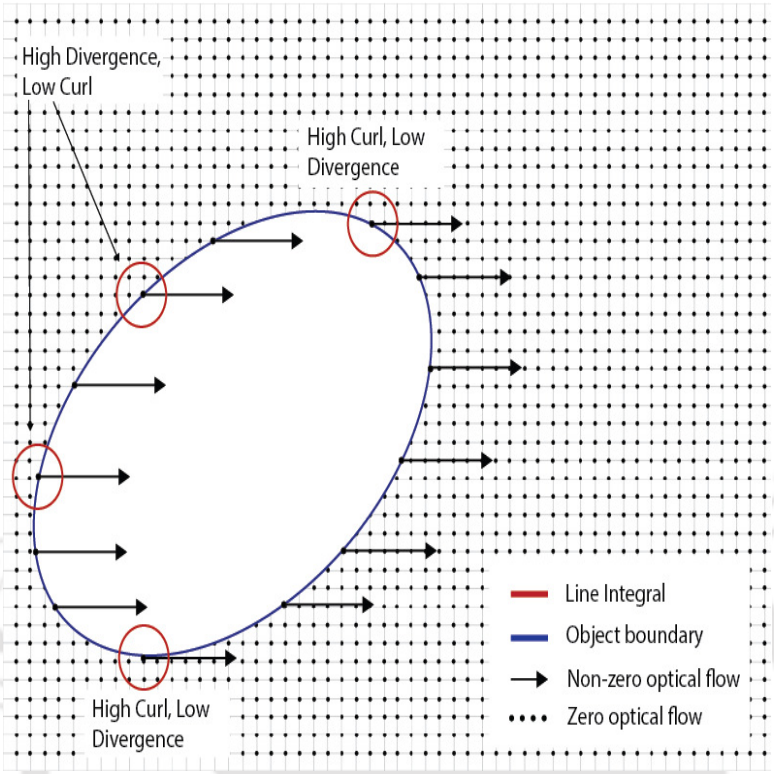


Figure 4.1: Illustration of the optical flow of an object boundary with line integrals for divergence and curl.

4.1.1 Proposed method

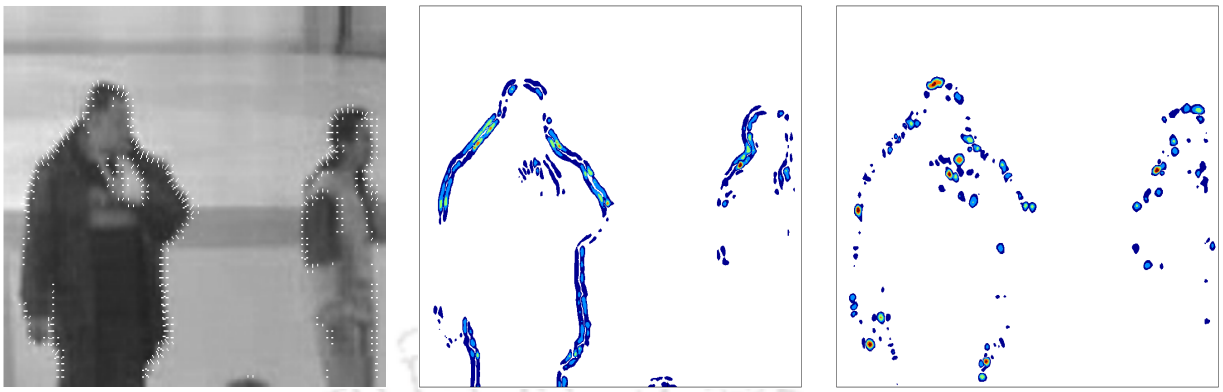
Let the velocity vector $\vec{V}$ of a point in a frame, as represented by its optical flow, be expressed as a linear combination of unit vectors in x direction ($\vec{i}$) and y direction ($\vec{j}$) as:

(4.1) $$\vec{V} = u\vec{i} + v\vec{j}$$

For completeness, we reproduce the equation to calculate the curl of $\vec{V}$ as:

(4.2) $$\nabla \times \vec{V} = \frac{\partial V_u}{\partial v} - \frac{\partial V_v}{\partial u}$$

Note that, $\nabla \times \vec{V}$ is a vector. So, we take its magnitude, and find local maxima in a



(a) (b) (c)

Figure 4.2: Example of vector calculus applied to optical flow of the video: (a) Sample frame with optical flow overlaid, contours of (b) divergence and (c) curl magnitudes of the optical flow of frame in (a).

frame in order to find locally isolated points. If $\left|\nabla \times \vec{V}\right| > T$ at a point, and it is also a local maxima then we consider it as an interest point, where T is a threshold. The threshold T can be fixed empirically to give the desired detection sensitivity while trading-off with specificity. Alternatively, one can pick a desired number of interest points by sorting the local maximas according to their curl magnitude and picking the top, say, 50 points for each frame. In this paper, we have considered a fixed threshold for all videos, which was $T = 0.05$.

4.1.2 Experimental Results

We evaluated proposed stability and robustness of the interest point detected using the proposed method. We expected more stable and robust interest points to also improve action recognition performance when used with appropriate descriptors in a bag of features model.

4.1.2.1 Stability and Robustness

The proposed and other methods for detecting video interest points were tested for robustness and stability with respect to two common transformations in video copy search — compression and scaling. For this, we used repeatability rate and displacement

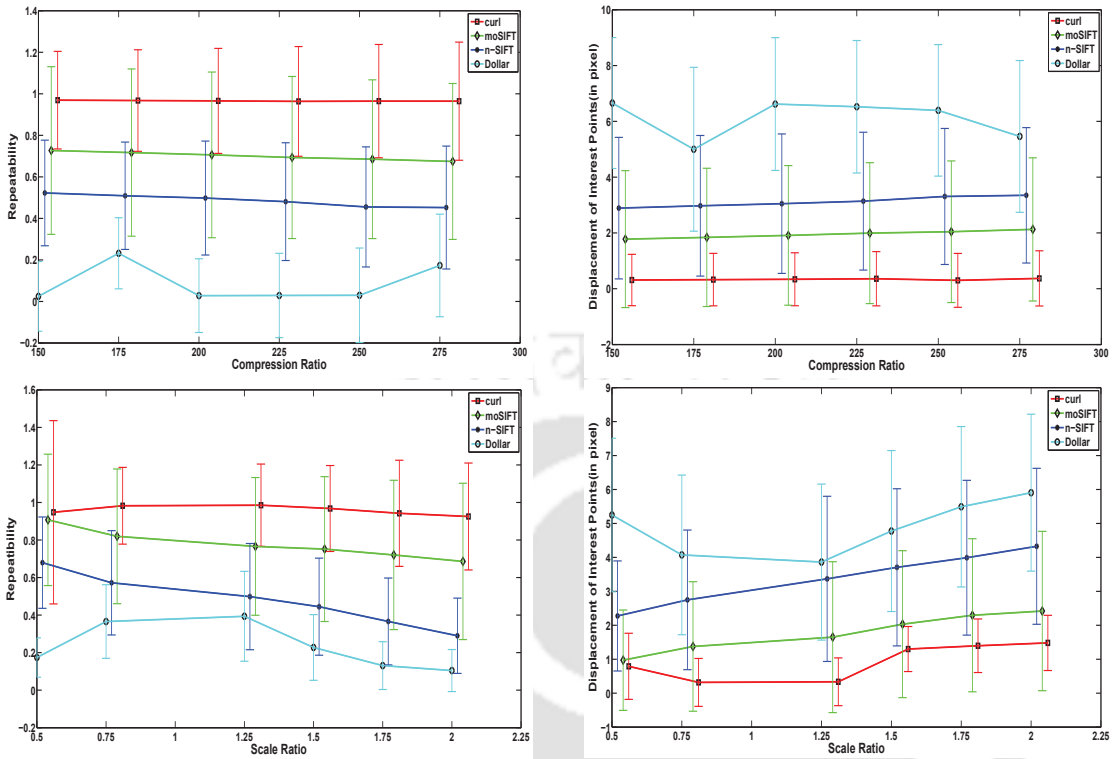


Figure 4.3: Repeatability and displacement for video compression and spatial scaling for the compared methods: Proposed method is labeled Curl. Results were averaged over 100 videos. Standard deviation is shown as error bars. A slight x-axis offset was added to the line plots to avoid visual overlap of error bars for clarity.

of interest points between pairs of corresponding frames in the original and transformed videos as performance metrics. The repeatability rate [87] is defined as the number of points repeated between two images (corresponding frames) as a fraction of the minimum number of detected points between the two images. The repeatability rate $r(\epsilon)$ for an image I', which is a transformed image of I, is defined by Equation 4.3

$$r(\epsilon) = \frac{|R(\epsilon)|}{\min(n, n')} \tag{4.3}$$

where n and n' are the number of interest points detected in the overlapping part of images I and I', respectively. $R(\epsilon)$ is the set of matched point pairs between the two images in a search window of size ϵ around the expected match location in image I' based

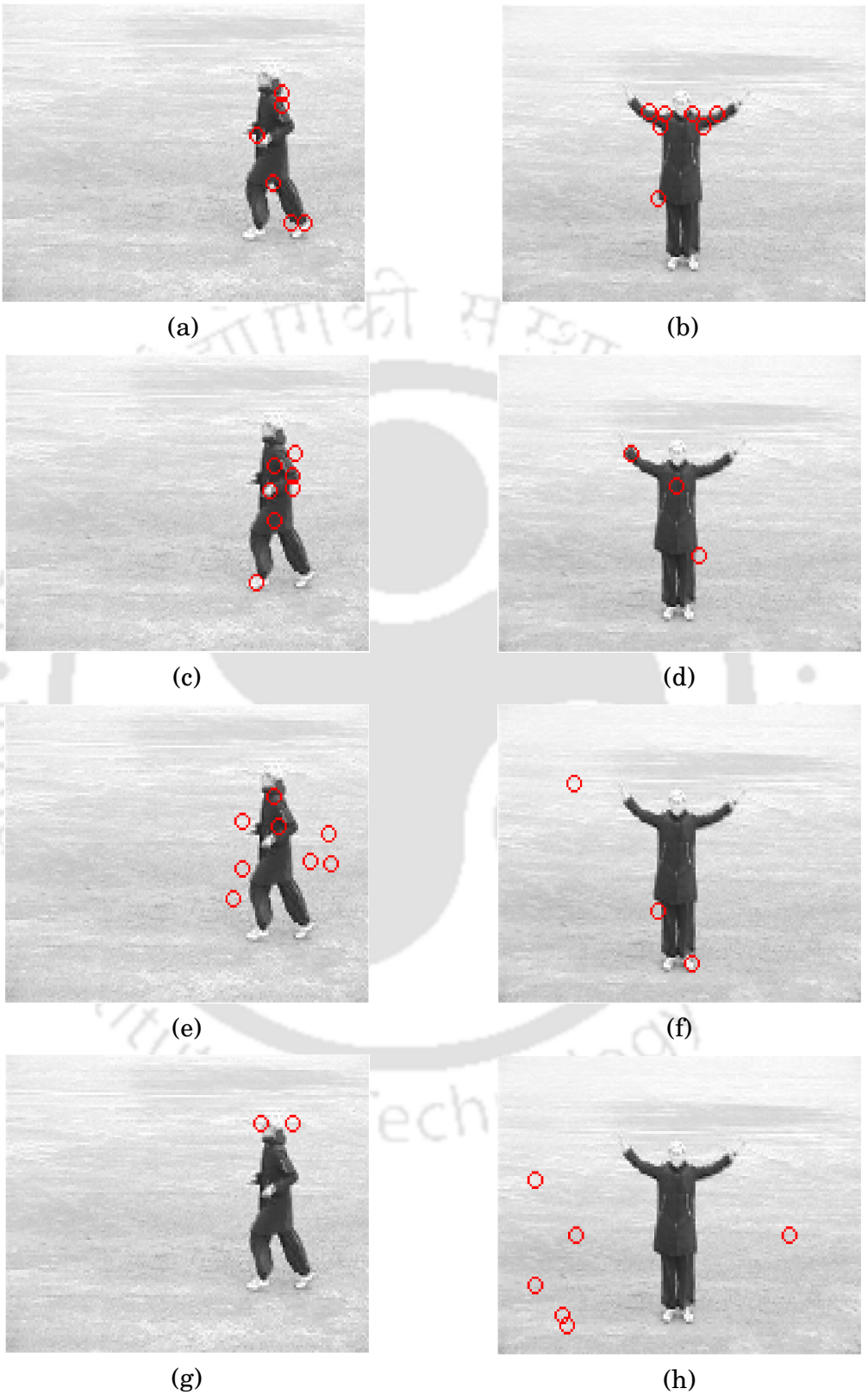


Figure 4.4: Examples of interest points detected by various methods, (a)-(b) Curl, (c)-(d) Mo-SIFT, (e)-(f) n-SIFT, (g)-(h) Dollar.

on the location of an interest point in the image I. Ideally, an interest point should be found in the exact corresponding locations in the original and transformed images. That is, for a compressed video, the interest point should be in the exact same location, but for the scaled video, the pixel coordinates of the predicted interest point will also be scaled by the same factor as the entire frame. However, due to appearance changes arising from the transformations, the interest point may be displaced slightly, or may not be found at all.

We have considered a window of $3 \times 3 \times 3$ for matching the interest point. Interest points can be matched only within this window with respect to the original video. Although we could have chosen a larger window size such as $5 \times 5 \times 5$ or $7 \times 7 \times 7$, but larger windows would have increased chances of spurious matches.

Displacement of interest point is measured in terms of distance, i.e., how far from its corresponding location in the original frame can an interest point be found in the transformed frame. To calculate displacement, we search for the interest point in a much larger neighborhood than the one used for calculating repeatability.

The video dataset used for the present experiments were obtained from KTH [89] and Weizmann [4] containing various videos showing human actions e.g. walking, running, boxing. A total 100 videos from both the datasets were considered for the experiment. We took the original video set and computed the transformed video set with different compression ratios and scaling factors. The original video set was in compressed format with a compression ratio of 1/3. These were further compressed with compression ratios of 1/150, 1/175, 1/200, 1/225, 1/250 and 1/275. We also scaled the original videos in x and y dimensions by factors of 0.5, 0.75, 1.25, 1.5, 1.75 and 2. FFmpeg software was used to compress and scale the videos (using the MPEG4 codec for compression and bicubic interpolation for scaling). In total, for each original video, we had 6 compressed and 6 scaled videos. The repeatability and displacement measures were averaged over all frames taken from these 100 videos.

The proposed method was implemented in Matlab® and compared with other methods studied — *Dollar* [21], n-SIFT [15] and Mo-SIFT[14]. We used the original authors'

Table 4.1: Average accuracy for the combination of proposed detector with various local descriptors on KTH and Weizmann dataset

	3D-SIFT	HOG3d	HOG	HOF	HOG/HOF
KTH	84.99%	85.39%	85.91%	**93.4%**	87.39%
Weizmann	**93.64%**	88.54%	87%	88.54%	87.51%

implementation for *Dollar* with their recommended parameters. We implemented n-SIFT and MoSIFT based on the algorithms described in the respective publications, and empirically tuned their parameters to give reasonable sensitivity-specificity trade-off. We have used Horn-Schunck method for optical flow [39], which is a library function in Matlab®.

The repeatability curves for compressed and scaled videos are shown in Fig. 4.3. In both cases, the proposed method performs better than other techniques that were studied. MoSIFT results are closer to the proposed method because it is also based on optical flow, and it makes use of the unique properties of the temporal dimension. Similarly, results for displacement of interest points versus compression ratios and scale ratios are shown in Fig. 4.3. It can be observed that the average displacement of interest points is smaller for the proposed method compared to other methods. Standard deviation is also shown in each graph which shows that the proposed method has less variation and it is more stable. The proposed method showed more variance in repeatability as compared to other methods when videos were scaled down. This is because at a lower scale the density of points was less. This led to some interest points not being detected in some videos. As the scale increases the density of points also increases leading to a decrease in variance of repeatability.

Additionally, qualitative analysis, as represented by Fig. 4.4, showed that the interest points detected by the proposed method appear only on relative motion boundaries. These points are neither too sparse nor do they appear on the stationary background, thus making them potentially useful for action recognition. This is not the case with the other methods studied.

4.1.2.2 Action Recognition

The proposed detector was combined with local descriptors such as HOG/HOF [57], HOG3d [50] and 3D-SIFT [90] for action recognition on two datasets. The KTH actions dataset consists of six human action classes: walking, jogging, running, boxing, waving, and clapping. The sequences were recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. The Weizmann dataset consists of ten human action classes: walk, run, jump, bend, one-hand wave, two-hands wave, jump in place, jumping jack and skip with the homogeneous outdoor background. A video sequence was represented as a bag of features extracted from the video [73]. Features were first quantized into visual words and a video was represented as a frequency histogram over the visual words. Vocabularies were constructed with k-means clustering. We set the number of visual words to 4000 as suggested by [108]. Then the histogram of visual words occurrence was formed for each video which was used for classification using a non-linear support vector machine [11]. We used leave-one-out cross-validation method for classification [5].

The results of a combination of the proposed detector with local descriptors are shown in Table 4.1. From the table, we can conclude that no single category of features performs best on all kinds of the dataset. 3D-SIFT descriptors (not to be confused with n-SIFT detector) shows better results for Weizmann dataset because it captures shape in 3D and dataset has static background with no camera movement, illumination variation, and zooming. In case of KTH dataset due to the presence of camera movement, illumination variation and zooming the 3D shape characteristics were affected. Hence, HOF shows better results because it captures motion effectively. The best combination of proposed detector and descriptor is compared with state-of-art action recognition methods in Table 4.2. In order to compare various interest point detectors, we emphasize comparing our results with those that use a different detector-descriptor combination in a bag-of-features paradigm. The proposed method performs better in all the cases when compared to other bag-of-features models with an average accuracy of 93.64% in case of Weizmann dataset and 93.4% in case of KTH dataset. The methods that outperform the proposed

Table 4.2: Compairing the average accuracy of the best combination of detector and descriptor with state-of-art methods on KTH and Weizmann dataset

	KTH	Weizmann
Our method	**93.4%**	**93.64%**
Schuldt, Laptev , and Caputo [89]	71.83%	–
Wong and Cipolla [114]	87.7%	–
Jhuang et al. [43]	91.7%	98.8%
Scovanner, Ali, and Shah [90]	–	82.6%
Laptev et al.[57]	91.8%	–
Niebles, Wang, and Fei-Fei [73]	81.5%	90%
Klaser , Marszalek , and Schmid [50]	91.4%	84.3%
Willems, Tuytelaars , and Van Gool [112]	84.3%	–
Liu and Shah [65]	93.8%	71.69%
Yang, Wang, and Mori [119]	75.71%	–
Sun, Chen, and Hauptmann [99]	94%	97.8%
Jia Liu et al. [66]	73.5%	87.5%
Kovashka and Grauman [53]	94.53%	–
Fei Hu et al. [40]	74.4%	75.8%
Samanta and Chanda [85]	93.51%	–
Gilbert, Illingworth,and Bowden [32]	94.50%	–
Wang et. al. [107]	94.20 %	–
Peng et al. [77]	93%	–
Ballan et al. [2]	92.66%	–
Shi et al. [91] (CNN)	96.8%	–
Shuiwang et al. [44] (CNN)	90%	–

method in action recognition take temporal order into account, which is not used in approaches based on bag-of-features.

4.1.2.3 Discussion

We proposed a new method for detecting interest points in videos based on vector calculus of optical flow to make use of the unique properties of temporal dimension as described in Chapter 3. To evaluate the robustness and stability of various interest point detectors, we have used repeatability rate and displacement of a pixel between an original video and its transformed video by changing its compression factor and spatial scale. The results show that the proposed method gives interest points that are robust and stable and potentially more useful (not too sparse, nor on the stationary background) as compared

to the other techniques studied. The proposed detector also shows better results for the higher level task such as action recognition.

However, the limitation of this method is fixed threshold, which gives too sparse or too dense points for different videos. Also, the interest point detection is done for every frame which gives redundant dense points. To avoid this some kind of temporal localization is required.

4.2 Action recognition using temporally localize interest points

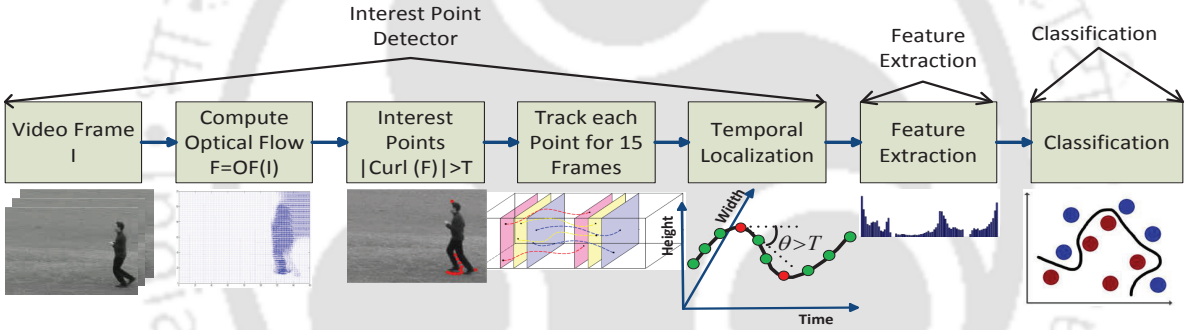


Figure 4.5: Block diagram of the proposed action recognition system.

4.2.1 Proposed Algorithm

Video classification using interest points has three major parts – IP detection, video representation, and classification. Differential motion, whether it is between objects or between close-by time instants for the same object, contain important information about action classes. Such information is somewhat lost in local IP descriptor-based methods especially when descriptor fusion methods (such as bag-of-words) are also used. Trajectory-based and sequential methods retain temporal information but they lead to cumbersome representation and classification algorithms. The proposed algorithm combines the desirable features of these three classes of video classification algorithms.

First, we detect IPs based on a modification of COF[117], which captures only points associated with an action. These points are tracked to obtain dense trajectories, capturing temporal information. Then, we perform temporal localization and prune uninformative points by retaining only an informative subset of points on the trajectories. Next, we compute a video representation for classification that is based on spatial and, more importantly, temporal distances between consecutive retained points on the trajectories, thus capturing information important for action recognition. These innovations have yielded substantial improvements compared to state-of-the-art on benchmark datasets. We now describe each module as shown in Fig. 4.5 in more detail as follows:

Adaptive interest point detection: Optical flow captures object persistence and smoothness in time dimension. The IP detector proposed in [117] was based on curl of optical flow to capture the relative tangential motion between objects such as foreground and background. For the optical flow (velocity) vector $\vec{V}$ of a point in a frame expressed as a linear combination of unit vectors in $\vec{i}$ and $\vec{j}$ in x and y directions respectively as $V_x\vec{i} + V_y\vec{j}$, the curl is defined as:

$$\nabla \times \vec{V} = \frac{\partial V_x}{\partial y} - \frac{\partial V_y}{\partial x} \tag{4.4}$$

If $\left|\nabla \times \vec{V}\right|$ is greater than a threshold T at a point, and it is also a spatial local maxima then we consider it as an interest point. In [117], the threshold T was fixed across videos, which caused the following problems due to scene change between videos. While a high threshold yields too sparse IPs for certain videos, a low threshold introduces spurious IPs based on non-zero curl due to background movement, noise, and video compression artifacts that vary from scene to scene. We adapted the threshold for each video according to the base level of curl magnitude that occurs even when there is no activity in that scene. We computed the threshold separately for each video as follows:

$$T_i = \min_j(\max_k(\left|\nabla \times \vec{V}_{ijk}\right|)) \tag{4.5}$$

where $\left|\nabla \times \vec{V}_{ijk}\right|$ denotes the curl magnitude of point k in a video frame j for video i.

Tracking: After detecting points in each frame, tracking was done using KLT tracker [101]. Trajectories tend to drift during tracking because of noise. Hence, we fixed the trajectory length to 15 frames as suggested in [107].

Pruning-based temporal localization: Temporal localization was done based on the change in the direction of trajectory as an indicator of its unpredictability, as shown in Fig. 4.6. If the angle between the tangents at a point and its predecessor on a trajectory was more than 20 deg, then the point was retained otherwise it was pruned.

Video representation: Since we do not use any IP descriptor, we retain spatial and, importantly, temporal information associated with the motion of an action by capturing the relation between consecutive temporally localized (unpruned) points on the trajectories. This information is characterized by both distance and angle between these points. Weighted orientation histograms of projections of line segments connecting consecutive retained IPs on to xy, yt and tx planes, were calculated, where x and y denotes the spatial dimensions of the frame and t denotes the time dimension. The weight (contribution) was proportional to the length of the line segment, while the bin was decided by the orientation of its projection on that plane. The histogram had 36 bins dividing the angle range of 0 deg to 180 deg into intervals of 5 deg. Histograms for the three planes were concatenated together and normalized to form the video representation.

Classification: After forming a compact video representation, we used it as input for an SVM to classify the actions. This is similar to a lot of state-of-the-art methods.

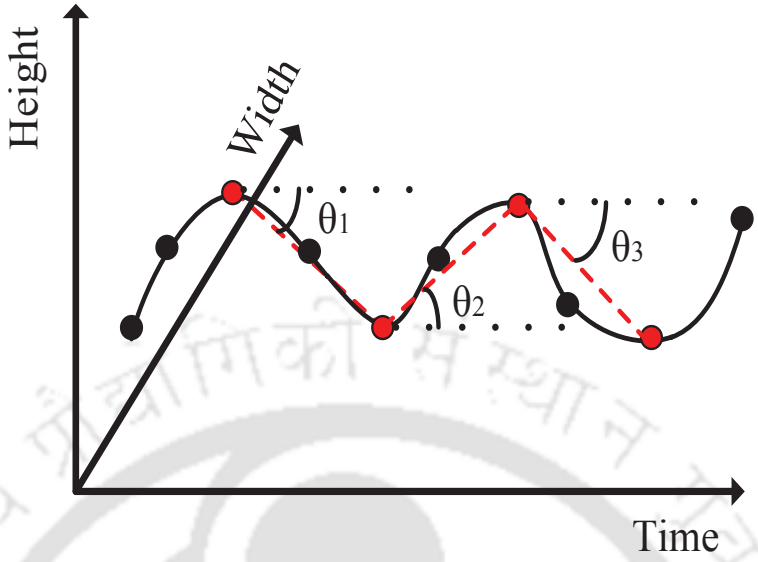


Figure 4.6: Illustration shows the trajectory of an interest point. Red dots are the temporally localized points. Change in slope is indicated by the angle.

4.2.2 Experiment and results

We compared the performance of our proposed technique with several state-of-the-art and a few pioneering legacy techniques on a simple and a complex video action datasets.

Dataset description: KTH database consists of simple scenarios of six types of human actions - walking, jogging, running, boxing, hand waving and hand clapping [89]. The actions were performed by 25 subjects in different scenarios such as outdoors and indoors, with different scale and clothing variations. UCF 11 is a complex data set with videos taken from "the wild", and consists of 11 action categories - basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog [64]. This dataset has variations in camera motion, object appearance and pose, object scale, viewpoint, background clutter, illumination conditions, etc. For each category, the videos are grouped into 25 groups with more than 4 clips in each group. The video clips in the same group share some common features, such as the same actor, similar background or viewpoint. It has been suggested to use entire groups instead of splitting their constituent clips into training or testing sets (specifically, leave-one-group-out) to test robustness of

action recognition techniques to unseen scene variations.

Table 4.3: Comparison of the average accuracy of the proposed method with state-of-art methods on KTH dataset. Methods not based on IPs are denoted by *.

Methods	Accuracy(%)
Proposed method	**98.20%**
Yadav et. al [117]	93.40%
*Jhuang et. al. [43]	96.00%
Kovashka and Grauman [53]	94.53%
Gilbert, Illingworth,and Bowden [32]	94.50%
Wang et. al. [107]	94.20 %
Laptev et al. [57]	91.80%
Wong and Cipolla [114]	87.7%
Niebles, Wang, and Fei-Fei [73]	81.5%
Klaser , Marszalek , and Schmid [50]	91.4%
Willems, Tuytelaars , and Van Gool [112]	84.3%
Liu and Shah [65]	93.8%
Yang, Wang, and Mori [119]	75.71%
Sun, Chen, and Hauptmann [99]	94%
Jia Liu et al. [66]	73.5%
Fei Hu et al. [40]	74.4%
Samanta and Chanda [85]	93.51
Schuldt, Laptev , and Caputo [89]	71.83%
Peng et al. [77]	93%
Ballan et al. [2]	92.66%
Shi et al. [91] (CNN)	96.8%
Shuiwang et al. [44] (CNN)	90%

Experimental setup: Classification of actions was done using Libsvm package [27]. Best parameter selection was done using 3-fold cross-validation on the testing dataset. KTH dataset was divided into 80% training and 20% testing set randomly. The accuracy was averaged over several random selections of training and testing data. For UCF11 dataset we have followed the leave-one-group-out cross-validation(LOOCV) approach as suggested in [64]. LOOCV scheme leads to 25 cross-validation results for each action class, which were averaged.

Action recognition results: Table 4.3 and Table 4.4 show comparison of action recognition performance of the proposed method and state-of-the-art methods. KTH dataset contains simple actions in a constrained environment for which state-of-the-art meth-

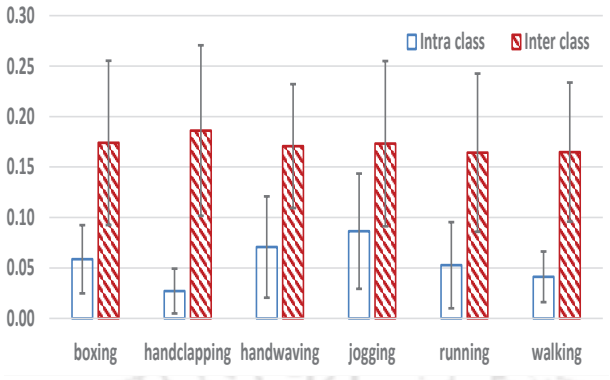
Table 4.4: Comparison of the average accuracy of the proposed method with state-of-art methods on the UCF11 dataset. Methods not based on IPs are denoted by *, and methods that do not seem to have left entire groups of clips out for testing or cross-validation are denoted by ?.

Methods	Accuracy(%)
Proposed method	**91.30%**
Wang et. al. [107]	84.20 %
*?Mota et. al. [71]	75.40 %
*Ikizler-Cinbis and Sclaroff [41]	75.21 %
Liu, Luo, and Shah [64]	71.20%
Wang et al. [106]	85%
Peng et al. [77]	67%
Zhu et al. [126]	89%
Sapienza, Cuzzolin, and Torr [86]	87%
Mahdyar et al. [81] (CNN)	90%

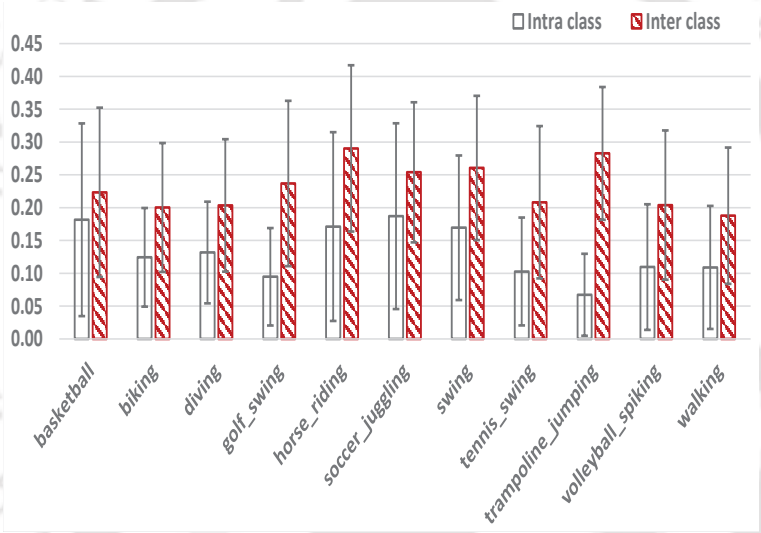
ods have already achieved good results. The proposed method showed better results than other methods that use IPs for KTH dataset and almost as good as more complex methods, as shown in Table 4.3. Importantly, for UCF11 dataset, the proposed method outperformed all the state-of-the-art methods, including those not based on IPs, as shown in Table 4.4.

Class separation: We have also shown the discriminative ability of the proposed video representation by comparing inter- and intra-class mean-squared distance in Fig. 4.7. The graph shows low inter-class and high intra-class variance that the distinctness factor of descriptors are high for most classes for both the datasets.

Impact of adaptive threshold: As shown in Fig. 4.8, the proposed thresholding method improved the quality of IPs detected when compared with the COF [117] and selective-STIP [10], which have shown to yield more meaningful IPs than other techniques. Our interest points are sparse as well as on the boundary of the moving object where relative motion is large whereas in case of selective-STIP and COF the number of IPs is either too large (some of which are on the background) or IPs are not exactly on the boundary of moving object. Motion boundary is important for characterizing an action when the



(a)



(b)

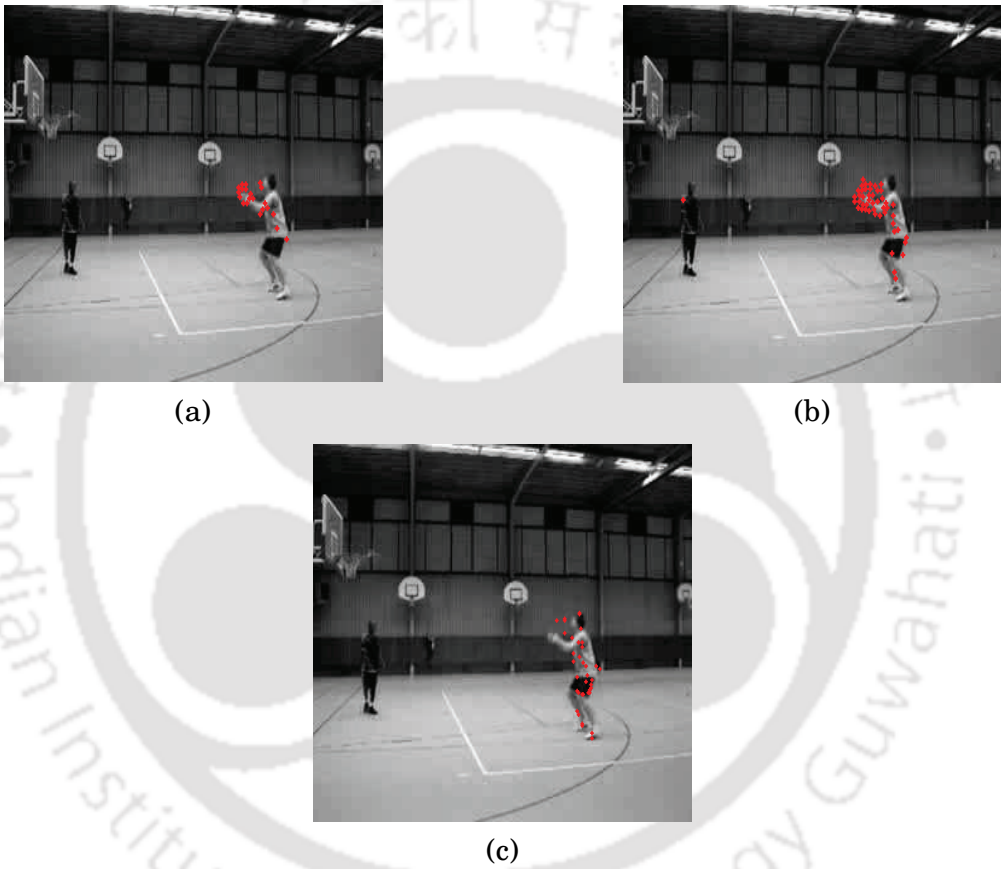
Figure 4.7: Inter- and intra-class squared distances for the proposed video representation for (a) KTH and (b) UCF11 datasets.

background is not static.

Impact of temporal localization:The proposed method has shown a drastic improvement with temporal localization as shown in Table 4.5. The reason is that our video representation was based on the orientation of connecting temporal localized points on the trajectories, and does not use local descriptors commonly used in other IP-based video representations. These points are action-specific which results in a different histogram

Table 4.5: Comparison of the average accuracy of the proposed method with and without temporal localization.

Ineterest point extraction	KTH	UCF11
With temporal localization	**98.2%**	**91.3%**
Without temporal localization	57.2%	39.5%



(a) (b)

(c)

Figure 4.8: Example of interest points of the proposed method (a) with adaptive threshold, (b) with fixed threshold [117], and (c) Selective-STIP [10]

for different actions. If we consider all the points on the trajectories rather few points then most of the consecutive pairs of points will have small changes in orientation, and will contribute to only a few bins of the histogram. This will make it difficult to distinguish between action classes.

4.2.3 Discussion

We have proposed a video representation for action recognition that performs better than state-of-art methods on two widely used benchmark datasets. IPs are detected on motion boundaries that are important to discard moving background, and retain only those IPs that represent new information based on the large change in trajectory direction. Thus it is much more compact than using entire trajectories. Additionally, we have shown that temporal localization plays an important role in improving the information contained in certain video representations where local IP descriptors are not used. We conjecture that using temporal localization will improve the performance of video representations based on local IP descriptors as well.

4.3 Interest Point Detector for Videos based on Differential Motion

In this section, we extend the proposed interest point detector in section 4.1. We have introduced an adaptive thresholding to control the number of interest points and also introduced the scale-space invariance. We also proposed a modification in existing descriptor by incorporating the location of the interest points.

4.3.1 Adaptive threshold for curl of optical flow

In [117], the threshold T was fixed across videos, which caused the following problems due to scene change between videos. While a high threshold yields too sparse IPs for certain videos, a low threshold introduces spurious IPs based on non-zero curl due to background movement, noise, and video compression artifacts that vary from scene to scene. We propose the following adaptive threshold for each video according to the base level of curl magnitude that occurs even when there is no activity in that scene:

$$T_i = \underset{j}{\text{median}}(\max_{k}(\left|\nabla \times \vec{V}_{ijk}\right|)) \tag{4.6}$$

where $\left|\nabla \times \vec{V}_{ijk}\right|$ denotes the curl magnitude of point k in a video frame j for video i.

4.3.2 Scale-adaptive interest points

As shown in Fig. 4.2 curl image, $I_c(x,y) = \left|\nabla \times \vec{V}\right|$ consists of blobs like structure at different scales. Scale selection plays an important role in detecting interest points. Koenderink (1984) and Lindeberg (1994) [52, 63] has shown that the only possible scale-space kernel is the Gaussian function. To achieve scale invariance Lowe has used Gaussian scale pyramid to detect points at each scale and keep only those points which have the maxima [67]. Similarly, we have also detected the interest point as described above for various Gaussian scales and selected those points which have maxima at a certain scale.

The scale space image is obtained by convolution of Gaussian kernel $G(x,y,\sigma)$ with the curl image $I_c(x,y)$.

$$L(x,y,\sigma) = G(x,y,\sigma) * I_c(x,y) \tag{4.7}$$

To detect a blob second order derivative of Gaussian is computed which is also known as Laplacian operation.

$$\nabla^2 L(x,y,\sigma^2) = \nabla^2 G(x,y,\sigma^2) * I_c(x,y) \tag{4.8}$$

The amplitudes of these derivatives decrease with scale. To obtain invariance, scale normalized derivatives should be used, given by

$$L^{norm}(x,y,\sigma^2) = \sigma^2 \nabla^2 G(x,y,\sigma^2) * I_c(x,y) \tag{4.9}$$

The characteristic scale is defined as the one that produces the peak of Laplacian response. To remove points with low amplitudes we use the threshold T as given in Equation 4.6. Laplacian can be approximated by the difference of Gaussian as shown in [67]. We followed the same approach as described in [67] to compute the difference of Gaussian using scale space pyramid as shown in Fig. 4.9.

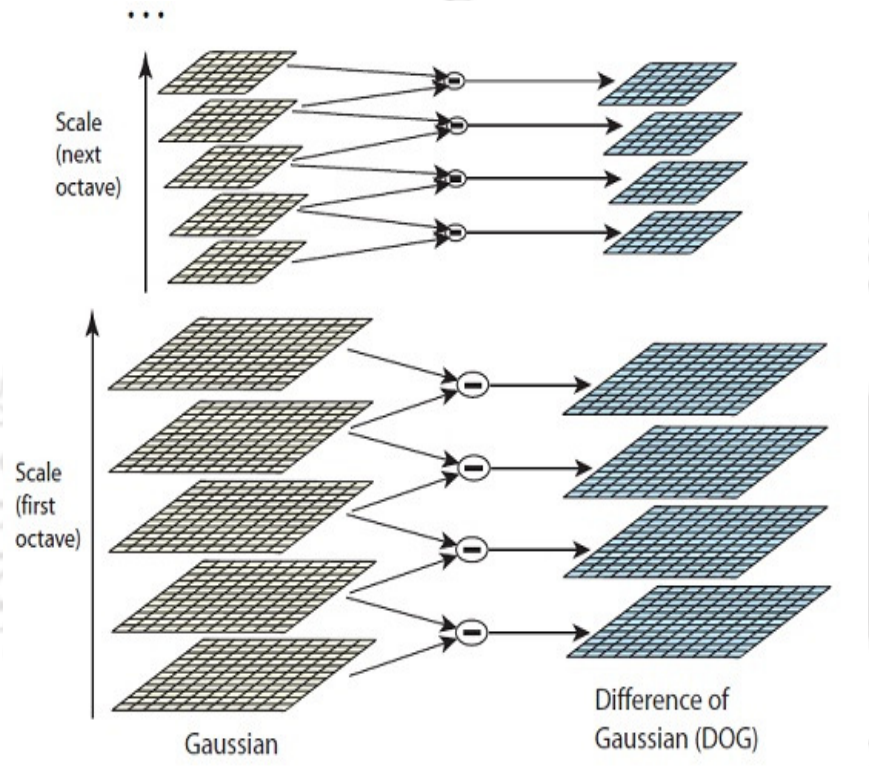


Figure 4.9: DOG scale-space pyramid [67].

4.3.3 Appearance feature extraction

Some neighborhood texture or edge information is required as a descriptor for matching interest points or for computing features for video classification. We extracted STIP [56] features for each interest points. STIP features are based on histogram-of-gradient (HOG) and histogram-of-flow (HOF). The implementation uses four orientations for HOG and five orientation bins for HOF, resulting in respectively 72- and 90-dimensional descriptors.

4.3.3.1 Location feature extraction

State-of-the-art interest point feature methods do not consider the location of interest point while computing the feature. Locations of interest points with respect to other neighborhood interest points are important to capture the 3-D shape of the graph formed by the interest points. A bag-of-words model does not capture the relative graph structure between interest points but it is computationally efficient. In order to use the bag-of-word model, we included the local neighborhood graph information in the descriptor itself. To do so we formed the histogram of distance of interest point with respect to the neighboring interest point as shown in the Fig. 4.10. The red dot is the interest point for which feature is being computed. We divide the spatial region around interest point into 8 equal angular partitions, each 45 degrees wide. Distances of the neighboring interest points from the center point in question are counted into three circular regions with radii 2, 4, and 8 pixels respectively. Similarly, temporal distance is also divided into three circular radii of 2, 4, and 8 frames. A 3-D histogram of dimensions $8 \times 3 \times 3$ is formed, which gives a 72-dimensional feature vector after flattening.

This location histogram along with STIP local intensity descriptor gave a feature with a total of 234 dimensions.

4.3.4 Experiments and Results

In this section, we describe the experimental settings, datasets used and performance evaluation of the proposed interest point detector and descriptor. Robustness and stability of the detector were checked using repeatability and localization error between otherwise identical videos with different compression and scale. The efficacy of the detector and descriptor was also examined for human action recognition using standard datasets and bag-of-words video representation.

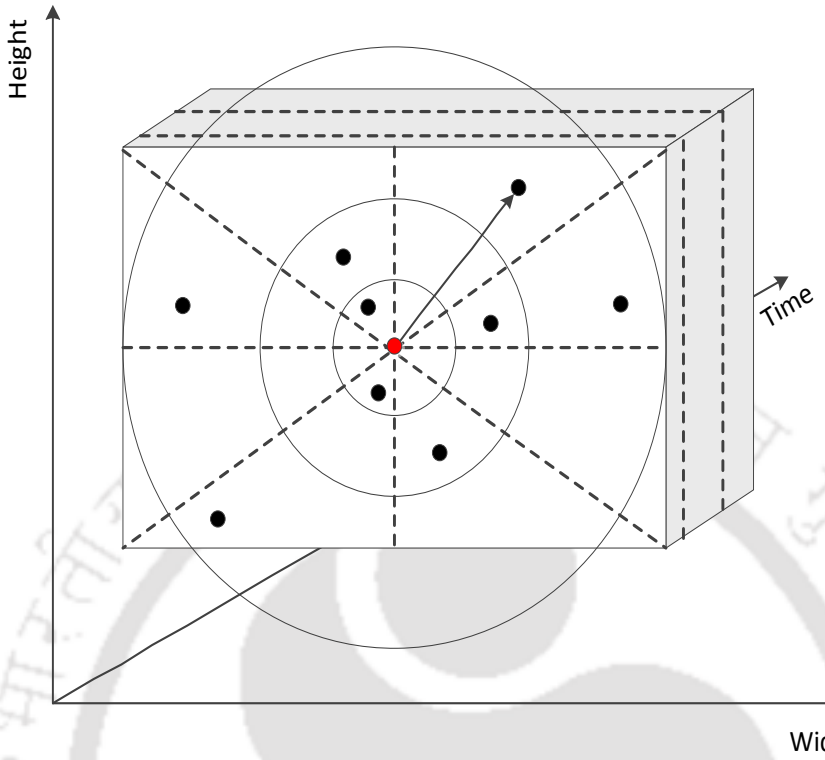


Figure 4.10: Illustration of computation of distance and direction histogram using the neighboring interest points for each interest point.

4.3.4.1 Datasets used

Four popular datasets were used for the experiments. The first was KTH, which is a simple action database that consists of six actions– boxing, hand-clapping, hand-waving, jogging, running and walking [89]. Each action was performed four times by 25 subjects in different scenarios such as outdoors and indoors, with different scale (distance from camera) and clothing resulting in 100 videos per action. The second dataset was Weizmann, which is a simple and small dataset consisting of 10 actions – walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, and skip [4]. Each action has 10 videos with different actors shot with a static camera. The third dataset was UCF11, which is a complex action dataset which consists of 11 actions – basketball shooting, biking, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking

with a dog [64]. This dataset has variations in camera motion, object appearance and pose, object scale, viewpoint, background clutter, illumination conditions, etc. For each category, the videos are grouped into 25 groups with at least 4 clips in each group. The video clips in the same group share some common features such as the same actor, similar background or viewpoint. Finally, we also tested on UCF-101, which is very large and complex dataset consisting of 101 action classes taken from realistic videos, collected from YouTube [96].

4.3.4.2 Robustness and stability of interest point detector

The proposed and other methods for detecting video interest points were tested for robustness and stability with respect to two common transformations in video copy search-compression and scaling. For this, we used repeatability rate and displacement of interest points between pairs of corresponding frames in the original and transformed videos as performance metrics.

4.3.4.3 Repeatability

The repeatability rate [87] is defined as the number of points repeated between two images (corresponding frames) as a fraction of the minimum number of detected points between the two images. The repeatability rate $r(\varepsilon)$ for an image I', which is a transformed image of I, is defined by equation 4.10 :

$$r(\varepsilon) = \frac{|R(\varepsilon)|}{\min(n, n')} \tag{4.10}$$

where n and n' are the number of interest points detected in the common part of images I and I' respectively. $R(\varepsilon)$ is the set of matched point pairs between the two images in a search window of size ε around the expected match location in image I' based on the location of an interest point in the image I. Ideally, an interest point should be found in the exact corresponding locations in the original and transformed images. That is, for a compressed video, the interest point should be in the exact same location, but for the

scaled video, the pixel coor6dinates of the predicted interest point will also be scaled by the same factor as that of the entire frame. However, due to appearance changes arising from the transformations, the interest point may be displaced slightly, or may not be found at all.

We have considered a window of $3 \times 3 \times 3$ for matching the interest points for contribution to repeatability. We didn't choose a larger window size such as $5 \times 5 \times 5$ or $7 \times 7 \times 7$, to avoid spurious matches.

4.3.4.4 Displacement

Displacement of interest point is measured in terms of distance, i.e., how far from its corresponding location in the original frame can an interest point be found in the transformed frame. To calculate displacement, we search for the interest point in a larger neighborhood than the one used for calculating repeatability.

For repeatability and localization error, KTH and Weizmann datasets were used. We took the original video set and computed the transformed video set with different compression ratios and scaling factors. The original video set was in compressed format with compression ratio of 3:1. These were further compressed with the compression ratios of 150:1, 175:1, 200:1, 225:1, 250:1 and 275:1. We also scaled the original videos in x and y dimensions by factors of 0.5, 0.75, 1.25, 1.5, 1.75 and 2. FFmpeg® software was used to compress and scale the videos using the MPEG4 codec for compression and bi-cubic interpolation was used for scaling. In total, for each original video, we had 6 compressed and 6 scaled videos. The repeatability and displacement measures were averaged over all frames taken from these videos.

The proposed method was implemented in Matlab® and compared with other methods – Dollar, n-SIFT, MoSIFT, and STIP. We have used original authors' implementation for Dollar and STIP. We have implemented n-SIFT and MoSIFT based on the algorithms described in the respective publications and empirically tuned their parameters to give reasonable sensitivity-specificity trade-off.

The repeatability curves for compressed and scaled videos are shown in Fig. 4.11(a)

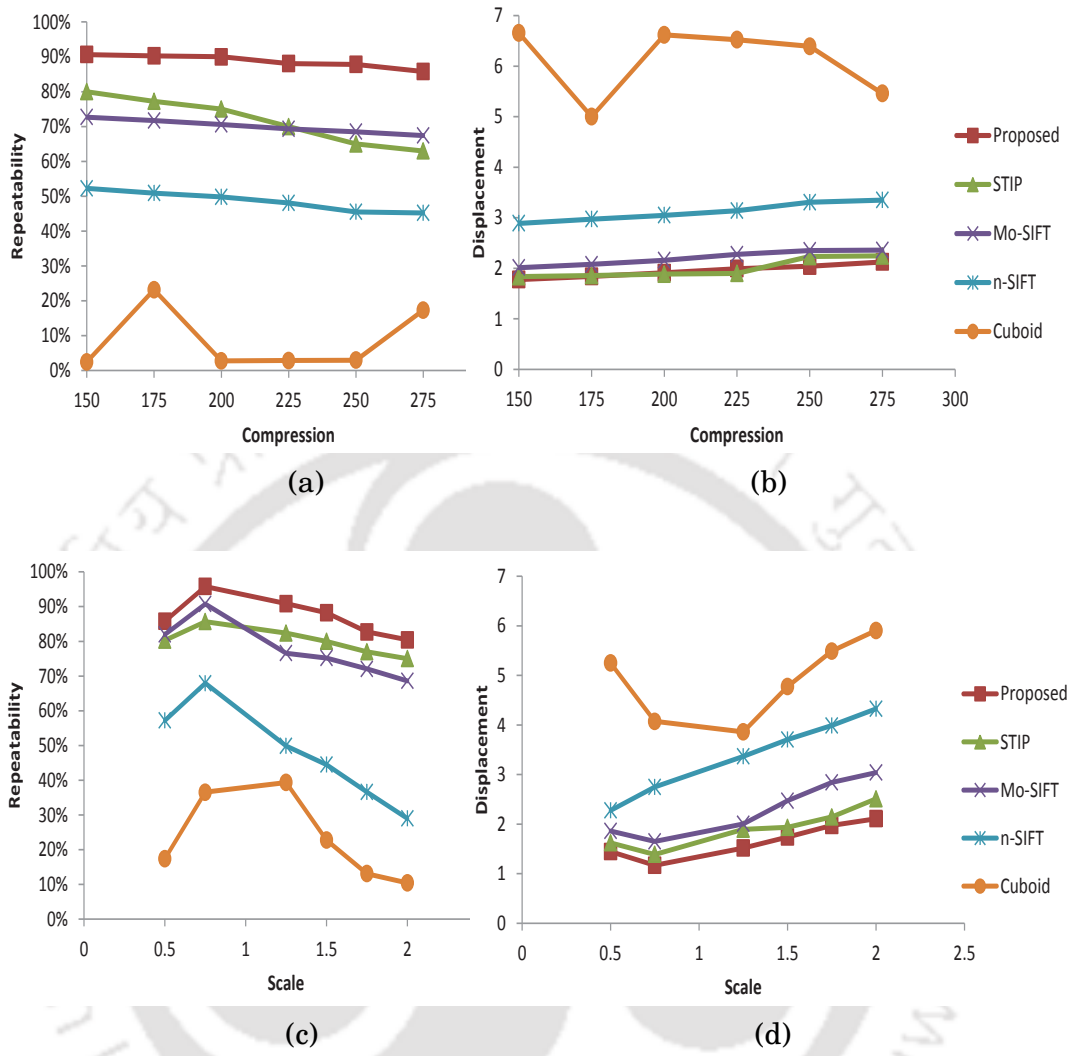


Figure 4.11: Repeatability and displacement for video compression and spatial scaling for the compared methods. Results were averaged over 100 videos.

and Fig. 4.11(c) respectively. In both cases, the proposed method performs better than other techniques that were studied. MoSIFT results are closer to the proposed method because it is also based on optical flow, and it also makes use of the unique properties of the temporal dimension. Similarly, results for displacement of interest points versus compression ratios and scale ratios are shown in Fig. 4.11(b) and Fig. 4.11(d) respectively. It can be observed that the average displacement of interest points is smaller or comparable for the proposed method compared to the other methods.

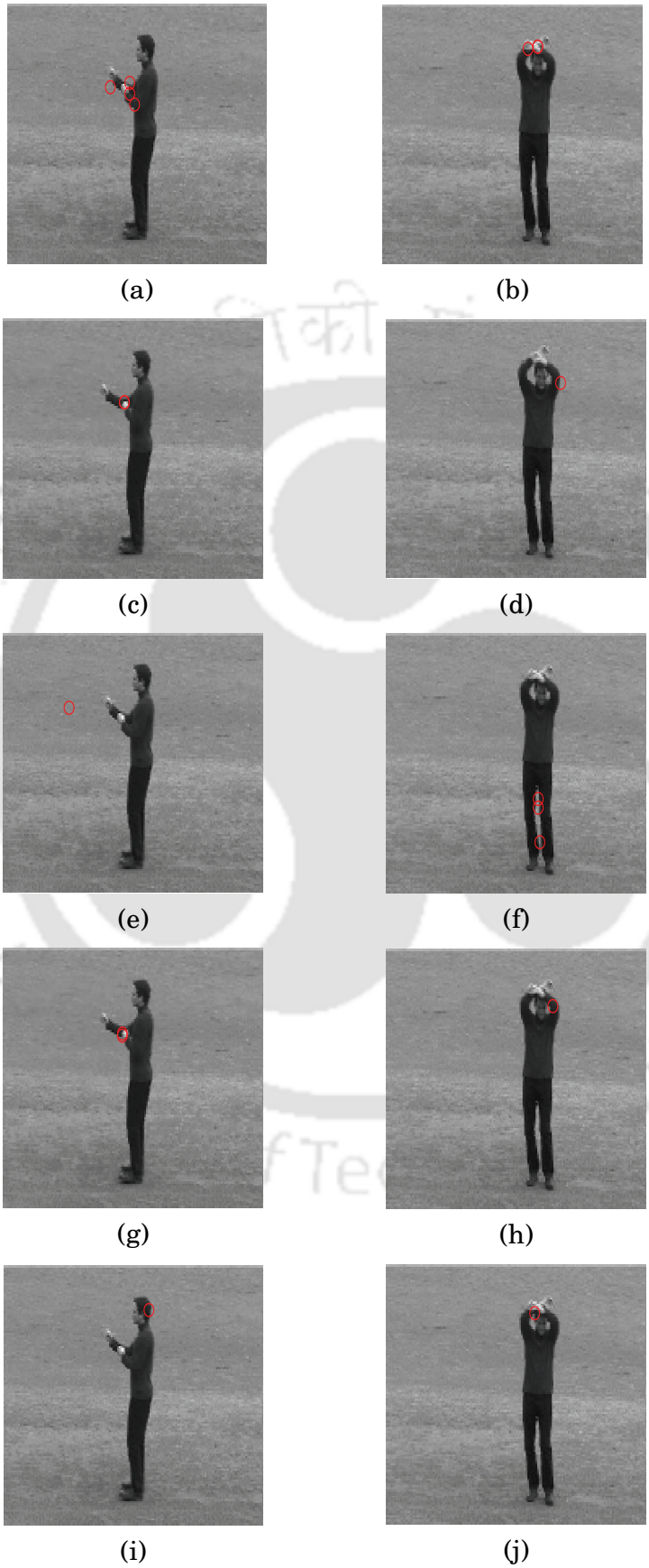


Figure 4.12: Example frames for boxing and hand-waving video sequences from KTH dataset with interest points using various detectors: (a)-(b) proposed method, (c)-(d) Mo-SIFT (e)-(f) n-SIFT, (g)-(h) STIP, and (i)-(j) Cuboid.

Table 4.6: Recognition rates of various detector-descriptor combinations on KTH dataset.

	Curl	MoSIFT	n-SIFT	STIP	Cuboid
HOG	86%	71%	76%	82%	77%
HOF	88%	80%	84%	92%	76%
HOG-HOF	93%	84%	87%	93%	84%
HOG-HOF-Loc	**99%**	87%	89%	96%	86%

Table 4.7: Recognition rates of various detector-descriptor combinations on Weizmann dataset.

	Curl	Mo-SIFT	n-SIFT	STIP	Cuboid
HOG	87%	63%	77%	74%	74%
HOF	88%	69%	75%	87%	79%
HOG-HOF	90%	71%	80%	87%	80%
HOG-HOF-Loc	94%	77%	83%	90%	83%

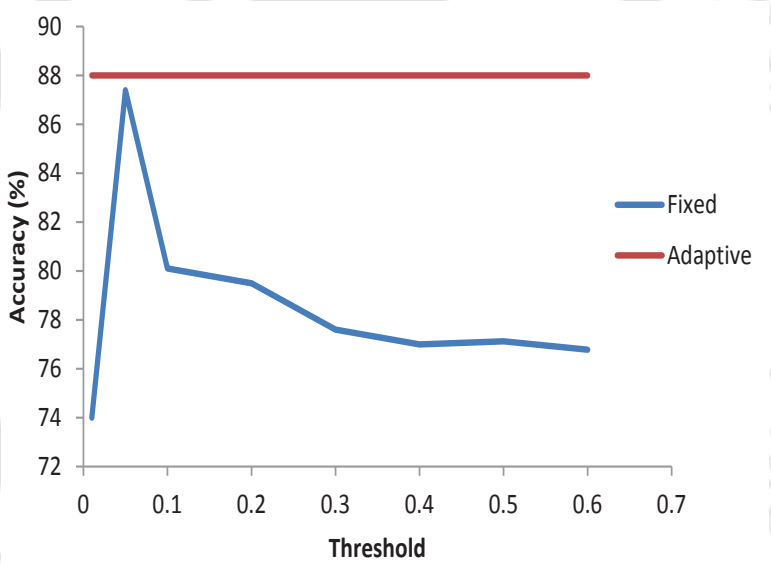


Figure 4.13: Comparison of fixed vs adaptive threshold.

4.3.4.5 Comparison of fixed threshold and adaptive threshold

We compared using a fixed threshold for all videos [117] (for various values of the fixed threshold) with the proposed scheme of using an adaptive threshold on KTH dataset. In case of adaptive threshold, each video has a different threshold for detecting interest point which depends on the motion content of the video as determined by Equation 4.6. Therefore, adaptive threshold gives an optimum number of interest points for each video. In case of fixed threshold, some videos yield a lot of interest points and some very

few, which affects the performance. As shown in Fig. 4.13, using an adaptive threshold improves the performance.

4.3.5 Qualitative comparison of interest point detectors

We have compared the visual quality of the proposed detector with state-of-the-art methods. As shown in the Fig. 4.12 the proposed interest points appear only on relative motion boundaries. Other methods such as Mo-SIFT, n-SIFT, and Cuboid give very sparse points, some of which also lie on the background. The proposed method gives neither too sparse nor too dense interest points.

4.3.5.1 Comparison of interest point detectors and descriptors for action recognition

We have compared the proposed interest point detector with state-of-the-art detectors along with various descriptors as shown in Table 4.6 and Table 4.7. For comparison, we have done human action classification on two simple datasets KTH and Weizmann. We have used HOG, HOF, their combination, and the location feature proposed in Section 4.3.3.1 to see the relative contribution of each feature. The features were quantized using k-means with a code-book size of 2,000 and used in a bag-of-words model. A support vector machine was trained using LibSVM [11] for classification.

It is evident from Table 4.6 and Table 4.7 that HOF performs slightly better than HOG, but their combination performs much better than both on KTH dataset, and slightly better than both on Weizmann dataset. When the location feature is added to the mix, the results improve substantially, especially on Weizmann dataset. Going forward we use HOG, HOF, and location features. In later experiments, we also optimize the code-book size and use a GMM based feature computation instead of k-means, which further improved the recognition rates.

4.3.5.2 Application of the proposed video representation to action recognition

The combination of interest point detector and descriptor has many applications such as human action recognition, event detection, video retrieval etc. We tested the utility of our proposed method for human action application and compared it with the state-of-the-art methods. Human action recognition can also be useful for event detection and video retrieval tasks. We have considered KTH, UCF11, and UCF101 datasets for this application, Weizmann is very small dataset and already very high accuracies have been achieved on it by others.

The framework for human action recognition includes feature extraction, code-book generation and classification. In [76] best practices were described for action recognition using a bag-of-words model, which we have used here. Feature extraction is done using HOG-HOF because they are widely used due to their easy usages and good performance. Descriptors (HOG-HOF-loc) were concatenated and PCA-whitened to ensure that features have same variance through different dimensions. For code-book generation generally k-means or GMM is used. K-means performs hard assignment of feature to codeword while GMM makes soft assignment of feature to each mixture component based on posterior probabilities. But unlike k-means, GMM delivers not only the mean information of code words, but also the shape of their distribution. Code-book generation was done using a Gaussian mixture model (GMM). GMM is a generative model to describe a distribution as an additive mixture of Gaussian distributions with different means, variances, and mixing coefficients. It can be described as follows:

$$\boldsymbol{p(x;\theta) = \sum_{k=1}^{K} \pi_k N(x;\mu_k,\sum_k)} \tag{4.11}$$

where K is the mixture and $\theta = \pi_1, \mu_1, \sum_1, ..., \pi_k, \mu_k, \sum_k$ is the set of model parameters. $N = (\pi_k; \mu_k, \sum_k)$ is an M-dimensional Gaussian distribution. The optimal parameters of GMM are learned through maximum likelihood estimation as $\text{argmax}_\theta \ln p(X;\theta)$ for a

given feature set $X = x_1, x_2, ...x_M$.

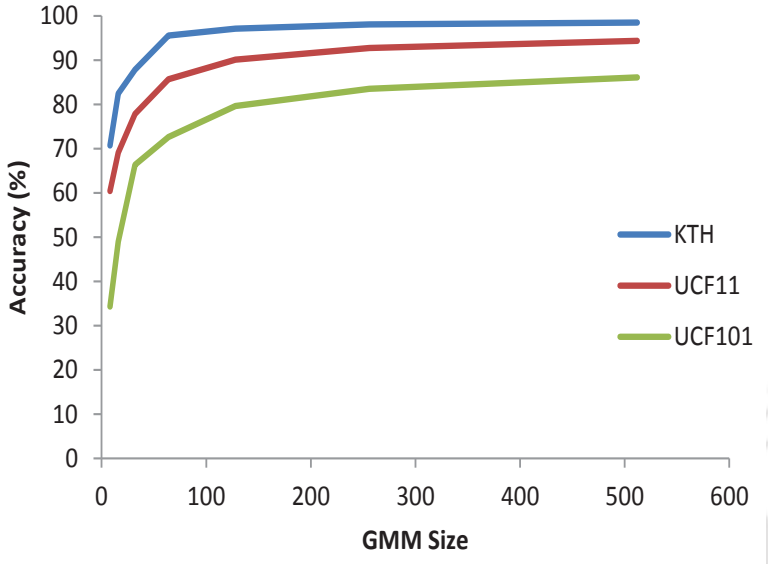


Figure 4.14: Comparison of accuracy with different GMM size.

After GMM, feature encoding was done using super vector method as described in [76] to get the final representation of the video. There are different ways to implement a super vector. One of them is Fisher vector, which uses Fisher kernel that combines the benefits of generative and discriminative approaches. As suggested in [76], we also used l_2-normalization. After encoding a linear SVM was used for action classification.

To generate the codebook, we randomly sampled 256,000 features from the training set to learn GMMs. The number of Gaussian components ranged from 16 to 512. The comparison with different GMM sizes is shown in Fig. 4.14. While for KTH and UCF11, the recognition accuracy saturated at 256, and for UCF101 it kept increasing till 512, presumably due to the diversity of action classes in the dataset.

After selecting the optimal GMM size, we compared the performance of our method with state-of-the-art action classification methods given in the literature. Table 4.8 shows the comparison of the proposed method with state-of-the-art methods. The proposed method outperformed all the state-of-the-art methods on KTH and UCF11 datasets. For UCF-101 the results were comparable to CNN-based (deep learning) methods with much faster training and testing. For UCF-101, our method takes less than 3 hours to train

Table 4.8: Action recognition accuracy of our and state-of-the-art methods.

	KTH	UCF11	UCF101
Proposed method	**99%**	**94%**	**84%**
Yadav, Shukla, and Sethi [118]	98.2%	91.3%	–%
Jhuang et. al. [43]	96.00%	–	–
Gilbert, Illingworth,and Bowden [32]	94.50%	–	–
Laptev et al.[57]	92%	–	–
Wang et al. [106]	95%	85%	–
Peng et al. [77]	93%	67%	–
Ballan et al. [2]	92.66%	–%	–
Zhu et al. [126]	–	89%	–
Wang, Qiao, and Tang [110]	–	–	**92%**
Karpathy et al. [48] (CNN)	–	–	63%
Shi et al. [91] (CNN)	96.8%	–	92.2%
Sapienza, Cuzzolin, and Torr [86]	–	87%	–
Kovashka and Grauman [53]	95%	–	–
Simonyan and Zisserman [93] (CNN)	–	–	88%
Shuiwang et al. [44] (CNN)	90%	–	–
Mahdyar et al. [81] (CNN)	–	90%	–
Wong and Cipolla [114]	87.7%	–	–
Niebles, Wang, and Fei-Fei [73]	81.5%	–	–
Klaser , Marszalek , and Schmid [50]	91.4%	–	–
Willems, Tuytelaars , and Van Gool [112]	84.3%	–	–
Liu and Shah [65]	93.8%	–	–
Yang, Wang, and Mori [119]	75.71%	–	–
Sun, Chen, and Hauptmann [99]	94%	–	–
Jia Liu et al. [66]	73.5%	–	–
Fei Hu et al. [40]	74.4%	–	–
Samanta and Chanda [85]	93.51	–	–
Schuldt, Laptev , and Caputo [89]	71.83%	–	–
Wang et. al. [107]	–	84.20 %	–
Mota et. al. [71]	–	75.40 %	–
Ikizler-Cinbis and Sclaroff [41]	–	75.21 %	–
Liu, Luo, and Shah [64]	–	71.20%	–

without using GPU, while CNN take days to train on the same machine while using a GPU.

4.4 Discussion

We proposed an interest point detector based on the differential motion which treats spatial and temporal dimension differently. The differential motion was computed by using curl of optical flow. Use of differential motion put the focus of feature extraction on relative motion. Our detector is robust and stable compared to common video transformation such as scaling and compression due to its emphasis on relative motion. We have also proposed a descriptor which uses the location of other interest points in the neighborhood of an interest point along with the state-of-the-art HOG/HOF descriptors. Experiments have shown that the combination of the proposed detector and descriptor performs better than other interest point detectors and descriptors or human action recognition application. In the future, interest point detector and deep learning features may be combined for faster and more accurate recognition.

CHAPTER 5

HUMAN ACTION RECOGNITION USING GLOBAL FEATURES

We present a technique for human action recognition in videos. Our key idea is to use relative motion between a subject performing the action of interest and its background instead of using its absolute motion. We do so by computing divergence of optical flow, which isolates the relative motion, unlike previous methods. Further, we also isolate temporally differential motion by computing differential motion maps. We propose a representation of videos of different sizes by projecting differential motion maps on three Cartesian planes. Even after this projection, the dimension of the feature vector is very high. Motivated by the observation that the underlying actions and motions are additively captured in our feature, we propose using specific dimension reduction and classification techniques. We suggest and show that semi-supervised non-negative matrix factorization performs better than other techniques to reduce the feature dimension, and l_2-regularized collaborative classifier performs better than other popular classifiers such as support vector machine and random forest. Our technique tested competitively on two benchmark human action recognition datasets.

5.1 Introduction

Inspired by [13], we proposed differential motion maps for computing the representation of videos. In [13], this concept is applied to RGB-D data. We have modified it and applied to the differential motion. The projection of depth motion map is easily understandable but the same projections in case of differential motion don't make sense. However, it gives better results in case of differential motion maps. Therefore, we have proposed another method where the projection on Cartesian planes are very simple and understandable without affecting the accuracy much. Now we describe both methods:

5.2 Proposed Method 1

The proposed method is an implicit combination of motion- and shape-based approaches. The main theme of our feature design is to capture differential motion in both space and time. This theme manifests in different ways in our feature extraction steps as we capture the motion between an actor and the background, as well as changes in that motion from frame to frame. Then, the dimension of the feature is reduced using a technique that is suited to the nature of the feature. Finally, classification is also done using a method suited for the extracted and compressed feature. The complete process is shown in Figure 5.1.

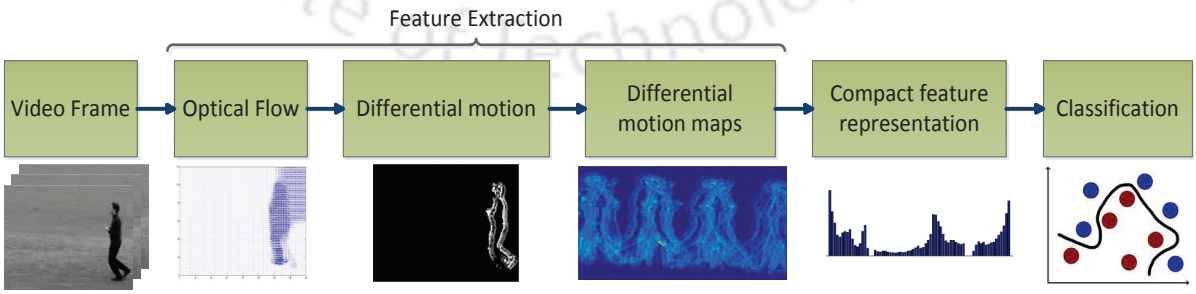


Figure 5.1: Block diagram of the proposed action recognition system.

5.2.1 Feature extraction

While we started with optical flow, we made specific additions to the feature extraction to robustly separate out different motions in the scene.

5.2.1.1 Absolute motion

Optical flow denotes point-wise absolute motion vectors between two consecutive frames and is used by several action recognition techniques [107, 117, 118] to compute motion-based features. It captures object persistence and smoothness in the time dimension. Assuming constant brightness of a physical point, its motion can be tracked using directed image gradients in spatial and temporal directions [39, 68] There are different methods to solve for the unknowns V_1 and V_2 that represent horizontal and vertical components of optical flow (or velocity on image plane) using various constraints for a robust solution. We have used Lucas-Kanade [68] method to determine the optical flow. Flow computed for an example frame in Fig. 5.2(a) is shown in Fig. 5.2(b).

5.2.1.2 Spatio-temporal differential motion

Optical flow is not enough to distinguish between different actions classes, especially when the camera is moving. This is because the motion of the background with respect to the camera also yields non-zero optical flow. Therefore, we propose detecting large spatial changes in optical flow on the boundaries of figure and ground for action recognition.

Since optical flow forms a vector field, large spatial changes in optical flow can be captured by computing its divergence. On the boundary of an object that is in motion with respect to its background, the line integral of divergence accumulates a larger magnitude on one side of motion boundary than the other side which is unable to completely counter that contribution. In contrast, at a point around which the motion is constant contributions to the divergence line integral cancel out. Thus, we get high divergence magnitude at motion boundaries, as shown in Fig. 5.2(c).

Formally, this divergence can be computed as follows:

$$R(x,y,t) = \nabla.\vec{V}(x,y,t) = \frac{\partial V_1(x,y,t)}{\partial x} + \frac{\partial V_2(x,y,t)}{\partial y} \tag{5.1}$$

5.2.1.3 Temporal differencing to capture acceleration

Constant motion is uninteresting. Sudden changes in velocity, such as those at the onset and end of atomic actions, are more interesting and are vital for defining an action. For example, while clapping, the moment when the hands start moving towards each other, or when they suddenly stop after coming together are iconic to identify that action. Therefore, we take a point-wise absolute difference of divergence of pairs of consecutive frames to capture acceleration as shown in Equation 5.15.

$$D_f(x,y,t) = |R(x,y,t+1) - R(x,y,t)| \tag{5.2}$$

5.2.1.4 Projection to Cartesian planes

For a video of size $X \times Y \times T$, we compute acceleration for $X \times Y \times (T-1)$ points, which can be a very large number for long videos. To compress this information, we project the differential acceleration maps onto the three orthogonal Cartesian planes defined by (x,y,t) coordinates to get three differential motion maps (DMMs), which we refer to as front, side, and top view projections. Without loss of generality, we assume that the frame timestamps are represented by integers, and show the front view projection as follows.

For the front view, in which the dimension t is eliminated, this operation is defined as:

$$DMM_{front}(x,y) = \sum_{t=1}^{T-1} D_f(x,y,t) \tag{5.3}$$

The equations above show how to calculate the frontal view DMM for a video with T

frames, although the summation only runs till $T-1$ because of that number of frame-wise differences computed in the previous step.

Computing the side and top view projections of a frame is different from that of the front view because a frame is two dimensional (2-D), and its projection will lead to a meaningless 1-D map. We follow the proposal by [13] to project depth motion maps on to the side and top views. To obtain 2-D side and top view projections, a frame needs to map to 3-D first. For this, we can assume that magnitude of divergence is equivalent to depth. The high value of divergence is treated as being equivalent to points that are closer to the camera in RGB-D space, and hence more important for action recognition. Hence, we can map a 2-D frame to 3-D to visualize the side view, where the third dimension is equal to the magnitude of divergence. Using Equation 5.4 the side view projection is computed for each frame. Then, a point-wise absolute difference of these projections is taken for two consecutive frames capture changes in motion as shown in Equation 5.5. Dimension t is eliminated using Equation 5.6 to obtain the differential motion maps of the side view projection

$$S(x,r,t) = y \tag{5.4}$$

where r is equal to $R(x,y)$ for the frame t.

$$D_s(x,r,t) = |S(x,r,t+1) - S(x,r,t)| \tag{5.5}$$

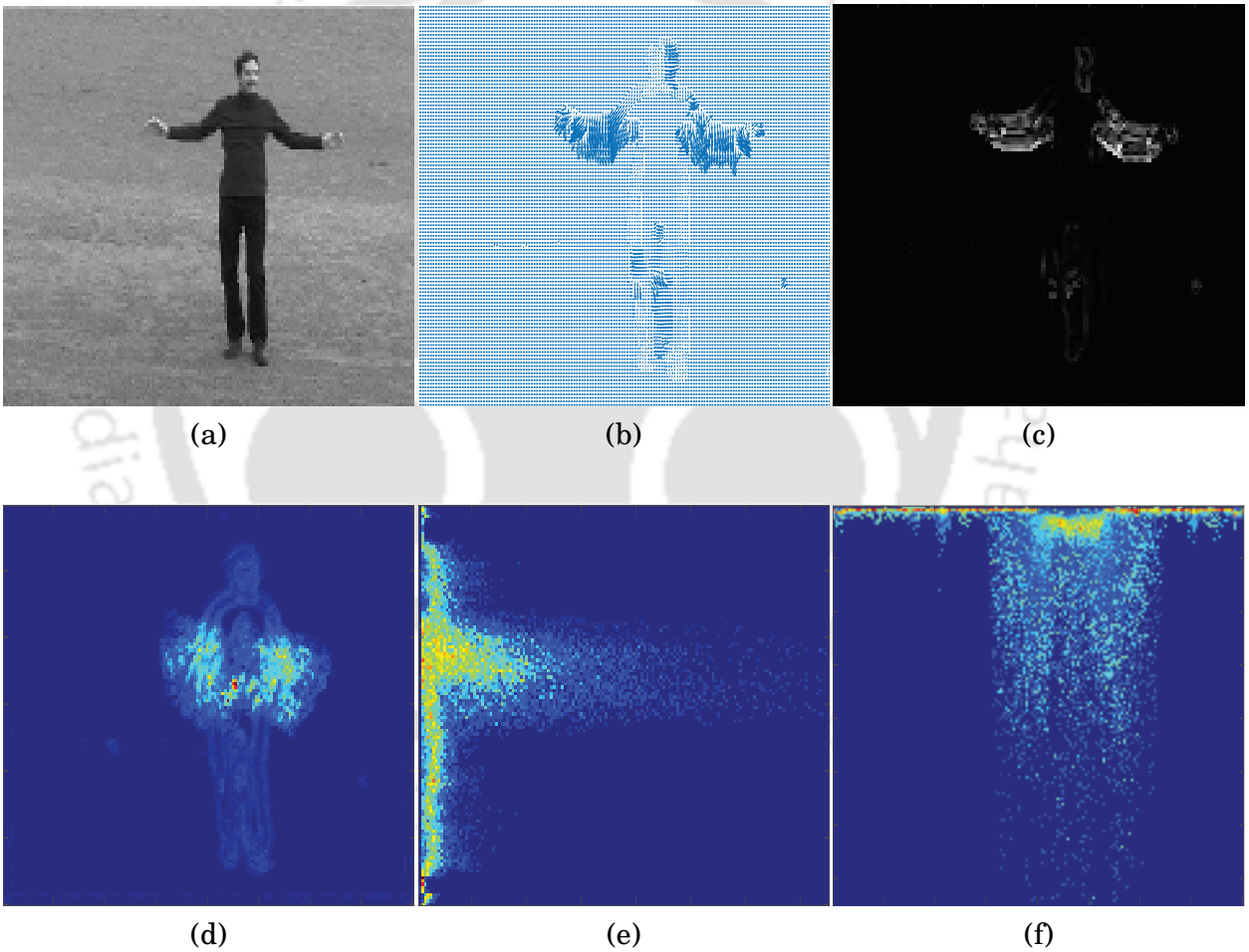
$$DMM_{side}(x,r) = \sum_{t=1}^{T-1} D_s(x,r,t) \tag{5.6}$$

Similarly for top view Equations 5.7-5.9 are used for the projection.

$$T(r,y,t) = x \tag{5.7}$$

$$D_t(r,y,t) = |T(r,y,t+1) - T(r,y,t)| \tag{5.8}$$

$$DMM_{top}(r,y) = \sum_{t=1}^{T-1} D_t(x,y,t) \tag{5.9}$$



(a) (b) (c)

(d) (e) (f)

Figure 5.2: Example frames for hand-clapping video sequence: (a) Original frame, (b) Optical flow (c) Divergence magnitudes of the optical flow, (d) Front view ($xy-$plane) differential motion map, (e) Side view ($yt-$plane) differential motion map, (f) Top view ($xt-$plane) differential motion map.

Thus, the entire video is used to generate three 2-D projected maps corresponding

to the front, side and top views as shown in Fig. 5.2(d)-(f). For a hand-clapping video sequence, we can easily see that the movements of hands have been captured in the front, side, and top views. Additionally, one can note that if there are two actions taking place in two different space-time locations in a video, the resultant DMMs will be additive sum of their individual DMMs owing to Equation 5.3. For this reason, we later suggest dimension reduction and classification techniques that are better suited to model this property of DMMs.

5.2.1.5 Resizing DMMs to a constant dimension

To deal with different sizes of motion maps for different actions, bi-cubic interpolation was used to resize all projection views to a fixed size. All values of the maps were normalized between 0 and 1 to avoid large values dominating the feature set. In our implementation, we projected the front view to a size of 100×50, side view to 50×82, and top view to 100×82, giving a total of a feature vector of little over $17,000$ dimensions.

5.2.2 Compact feature representation

Our extracted feature was very high dimensional. Therefore, dimension reduction techniques were applied to reduce the dimensions without affecting the performance. We tried to match the dimension reduction technique to the nature of the extracted feature (vectorized DMMs). In particular, we conjectured that linear techniques such as principal component analysis (PCA) and semi-supervised non-negative matrix factorization (SSNMF) [61] would be better suited than nonlinear methods such as kernelized PCA (KPCA) to reduce a feature that is additively formed from the underlying actions. Further, because the action can be present or absent, we expected non-negativity constraint used by SSNMF to model the feature better than PCA. SSNMF has the added advantage that it can use class labels to find only those additive components that help in classification. To test this hypothesis, we also used kernel PCA, which is a nonlinear dimension reduction technique, which we expected to perform worse than PCA and SSNMF. For

each of these, we experimented with a range of target (reduced) dimensions and other hyper-parameters.

5.2.2.1 Principal component analysis (PCA)

PCA is a widely used linear dimension reduction technique that projects a given set of vectors on to a small set of orthogonal eigenvectors of their covariance matrix. We conjectured that PCA will be helpful for reducing the dimensions of vectorized DMMs because, as we mentioned above, the DMMs are additive in the underlying features of multiple actions within the same video. As one of those actions could be the one that defines the class, it is likely to be captured by a principal component.

5.2.2.2 Semi-supervised non-negative matrix factorization (SSNMF)

Non-negative matrix factorization (NMF) is a popular method for low-rank approximation of non-negative matrix such that its mixing components and coefficients are also non-negative. Unlike PCA, these components need not be orthogonal. Non-negative additivity is desirable as atomic actions can either be present or absent in a video, but they cannot be subtractive due to the use of absolute value operators used in DMMs.

In semi-supervised NMF, the data and label matrices are jointly incorporated into NMF [61]. Suppose the data matrix $X = [x_1, x_2, ...x_n] \in \mathbb{R}^{m \times n}$ consists of m-dimensional non-negative data vectors each of which belongs to k classes. Label matrix is given by $Y = [y_1, ...y_n] \in \mathbb{R}^{k \times n}$, where each y_i is a vector of binary numbers such that j^{th} entry is 1 and remaining elements are 0, if x_i belongs to class j. Then, X and Y were jointly factorized, sharing common factor matrix S.

$$J = \|W \odot (X - AS)\|^2 + \lambda \|L \odot (Y - BS)\|^2 \tag{5.10}$$

where λ is a trade-off parameter used for determining importance of supervised term, $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{k \times r}$ are basis matrices for X and Y, respectively, and $L \in \mathbb{R}^{k \times n}$ is a weight matrix to handle missing labels. SSNMF fully makes use of labeled data in

minimization of Equation 5.10.

5.2.2.3 Kernel principal component analysis (KPCA)

KPCA is a nonlinear dimension reduction technique in which one performs PCA in a kernel space [88]. A gram matrix K is formed whose each entry in position (i, j) is defined by a Mercer kernel $k(x_i, x_j)$ for a pairs of data vectors (x_i, x_j), such that K is symmetric and positive semi-definite.

$$k(x_i, x_j) = k(x_j, x_i) = \phi(x_i).\phi(x_j) \tag{5.11}$$

where, ϕ defines a feature of possibly infinite dimension, if k is a Mercer kernel function.

A popular choice for k is an isotropic Gaussian function with a fixed standard deviation, which helps KPCA explore smooth but possibly non-planar manifolds over the underlying data. We used KPCA to compare the effect on nonlinear dimension reduction with PCA and SSNMF.

5.2.3 Classification

After feature extraction and its dimension reduction, a classifier was used to classify the actions. Just like with dimension reduction, we searched for a classifier that modeled the data to be a linear mixture of components, and classified based on estimating that model. One such classifier is l_2-regularized collaborative classifier (LRCC) [13], which we have used in this work. We have also compared it with various state-of-the-art classifiers, which are support vector machine (SVM) and random forest (RF). Now, we describe each of these classifiers in more detail:

5.2.3.1 L2-regularized collaborative classifier (LRCC)

LRCC is based on a linear and sparse representation of the data [13]. Consider a dataset with C classes, in which the training samples are arranged column-wise $\mathbf{A} =$

$[\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_c] \in \mathbb{R}^{d \times n}$, where $\mathbf{A}_j (j = 1, ..., C)$ is the subset of the training samples associated with class j, d is the dimension of training samples and n is the total number of training samples from all the classes. A test sample $\mathbf{g} \in \mathbb{R}^d$ can be represented as a sparse linear combination of the training samples, which can be formulated as $\mathbf{g} = \mathbf{A}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_C]$ is an $n \times 1$ vector of coefficients corresponding to all the training samples and $\boldsymbol{\alpha}_j (j = 1, ..., C)$ denotes the subset of the coefficients associated with the training samples from the class j, i.e., $\mathbf{A}_j$. The solution for $\boldsymbol{\alpha}$ can be found by numerically solving for $\mathbf{g}$ using l_2 regularization on mixing coefficients as follows:

$$\hat{\alpha} = \arg\min_{\alpha} \{ \|g - A\alpha\|_2^2 + \lambda \|L\alpha\|_2^2 \} \tag{5.12}$$

where λ is regularization parameter and $\mathbf{L}$ is the Tikhonov regularization matrix. The class label of $\mathbf{g}$ can be obtained from equation 5.12 as proposed in [115] as follows:

$$class(g) = \arg\min_{j} (e_j) \tag{5.13}$$

where $e_j = \left\| g - A_j \hat{\alpha}_j \right\|_2$.

5.2.3.2 Support vector machine (SVM)

SVM is a state-of-the-art linear classifier which maximizes the margin between two classes. Its kernelized version is widely used for action recognition, in which a Gaussian radial basis function is used as a kernel to map the feature to a high dimensional space. Because SVM is a binary classifier, for multi-class classification one-vs-all approach was followed using the LibSVM package [27].

5.2.3.3 Random forest (RF)

RF is another state-of-the-art classifier based on the principle that a set of classifier performs better than an individual one. It is an ensemble classification algorithm which uses trees as the base classifiers. RF uses two hyper-parameters: the number of classifi-

cation trees desired, k, and the number of prediction variables, m, to be tested at each node of a tree. The final value of the class assigned to each example will be equal to the most frequent value for the total number of k trees generated. We have used the RF as explained in [19].

5.2.4 Experiments and Results

We now describe the datasets used, experiments conducted, and their results to validate our feature extraction method and choices of dimension reduction and classification methods.

5.2.4.1 Datasets used

Two popular datasets were used in our experiments – KTH and UCF11. KTH is a simple action database which consists of 6 actions – boxing, hand-clapping, hand-waving, jogging, running and walking [89]. Each action was performed four times by 25 subjects in different scenarios such as outdoors and indoors, with different scale and clothing, resulting in 100 videos per action. UCF11 is a complex action database which consists of 11 actions – basketball shooting, biking, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog [64]. This dataset has variations in camera motion, object appearance and pose, object scale, viewpoint, background clutter, illumination conditions, etc. For each class, the videos were grouped into 25 groups with at least 4 clips in each group. The video clips in the same group share some common features such as the same actor, background, or viewpoint.

5.2.4.2 Experimental setup

For KTH dataset we followed the same procedure as mentioned in [89]. We divided the dataset into training and testing sets based on the subjects. For UCF11, it has been suggested to use all videos in a group for training or testing instead of splitting them into training or testing sets (specifically, leave-one-group-out cross-validation (LOOCV)) to

test the robustness of action recognition techniques to unseen scene variations [64]. This emulates a real scenario where the actor, background, or viewpoint of a test video may not have appeared in the training set at all. Hyper-parameter selection for the classifiers was done based on cross-validation on the training set. The accuracy was averaged over several random selections of training and testing data for the KTH dataset. For UCF11 dataset we followed the LOOCV approach, which leads to 25 cross-validation results for each action class, which were averaged.

We have used a machine with Intel Xeon E5-2643 / 3.3 GHz processor with 16GB RAM. Implementation was done in Matlab. Computational complexity is $O(m^3 + m^2 r) + O(n_c r) + O(p^2 N r)$, where m is the dimension of the feature vector, nc is the number of classes, r is the number of training samples, p is the number of warp parameters and N is the number of pixels in a sample.

5.2.4.3 Design choices

We next describe the results of the comparison of different dimension reduction techniques, hyper-parameter variation, and classifiers.

5.2.4.4 Comparison of dimension reduction techniques

PCA and SSNMF showed better performance than KPCA for both the datasets because both the methods decompose a feature vector into different components that may capture different parts of the videos. SSNMF gave better performance than PCA overall because it considers class labels while reducing the dimension. Both SSNMF and PCA model videos as linear combinations of underlying features, which we assume are constituent atomic actions. SSNMF combines them only additively, which is intuitive because an action can be present or absent, which is positive or zero contribution, but it should not be subtracted from a video representation as a negative contribution. Additionally, because we used semi-supervised NMF, its learning algorithm [61] can pick out discriminative components. For KTH and UCF11, SSNMF gives high accuracy at feature dimension

of 80 as shown in Fig. 5.3 and 5.4, while PCA tries to achieve the same at a higher dimension.

We compared different classifiers with different dimension reduction techniques and hyper-parameter settings.

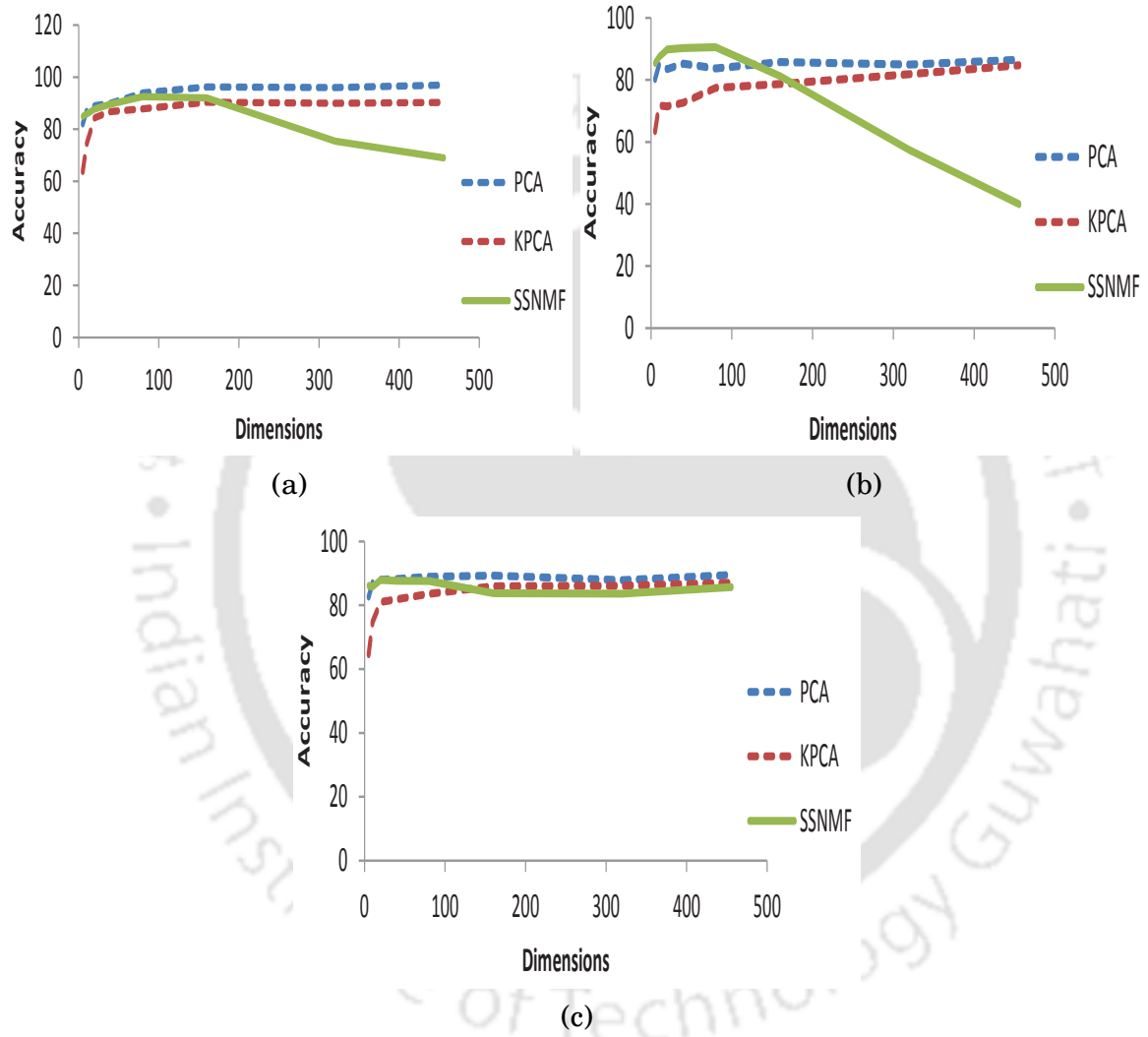


Figure 5.3: Comparison of (a) LRCC (b) SVM and (c) RF with different dimension reduction techniques for KTH dataset.

5.2.4.5 Hyper-parameter selection

Hyper-parameter selection was done for LRCC by varying regularization parameter λ as shown in Fig. 5.5. The parameter λ with best recognition rate was chosen for further

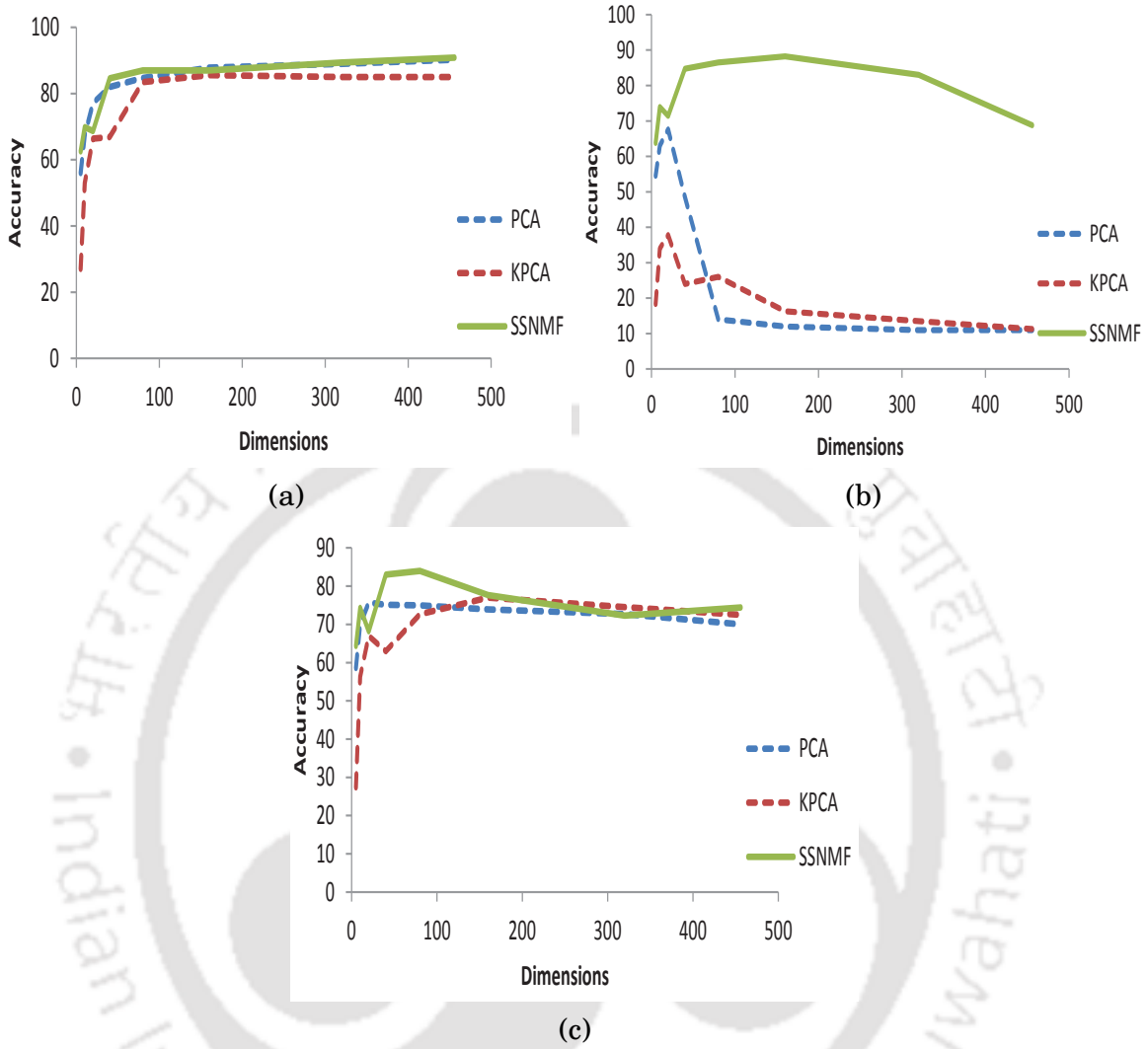


Figure 5.4: Comparison of (a) LRCC (b) SVM and (c) RF with different dimension reduction techniques for UCF11 dataset.

testing. For KTH dataset $\lambda = 0.01$ gave best result and for UCF11 it was 0.1. For SVM, hyper-parameters were the width of the Gaussian kernel and the weight of the slack penalty. For SSNMF, the hyper-parameter was the weight of label error (λ), for random forest hyper-parameters were number-of-trees and number of variables tested at each tree node. All hyper-parameters were found by grid search using cross-validation on training datasets.

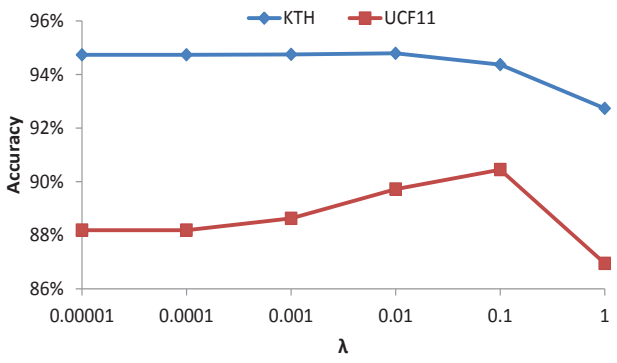


Figure 5.5: LRCC recognition rates for (a) KTH and (b) UCF11 datasets for various values of λ (in log scale).

5.2.4.6 Classifier comparison

We have compared recognition accuracies of the proposed method with different classifiers for different dimension reduction methods as shown in Fig. 5.3 and 5.4. LRCC showed better results for both KTH and UCF11. For KTH dataset, LRCC performed only slightly better than other classifiers because it is a simple dataset and there is not much scope for improvement. UCF11 consists of complex activities on which recognition rates were lower compared to KTH, and LRCC outperformed other classifiers by a larger margin. LRCC performed better because it models a video as a linear combination of several prototype feature vectors from the training data. This allows it to decompose a video's feature vector into its components arising from both in-class and out-of-class actions in the same video. The coefficient of in-class component helps it classify better for complex videos with out-of-class actions such as those in UCF11. SVM with a Gaussian kernel cannot do such decomposition. Similarly, RF also fails to obtain such a decomposition of the test data into a linear combination of the training data.

5.2.4.7 Absolute vs. differential motion

We have also compared feature representation using differential motion and optical flow. In case of optical flow, feature representation is done using the magnitude of optical flow instead of using differential motion. Recognition accuracy using differential motion was

Table 5.1: Comparison of optical flow and differential motion for KTH and UCF11 datasets.

Dataset	Optical flow	Differential motion
KTH	65.0%	94.8%
UCF11	44.1%	90.5%

better than that using optical flow for both the datasets as shown in Table 5.1. This shows that optical flow is not enough to separate the movement of objects of interest when there is another movement in the background due to camera motion or other actions in the scene.

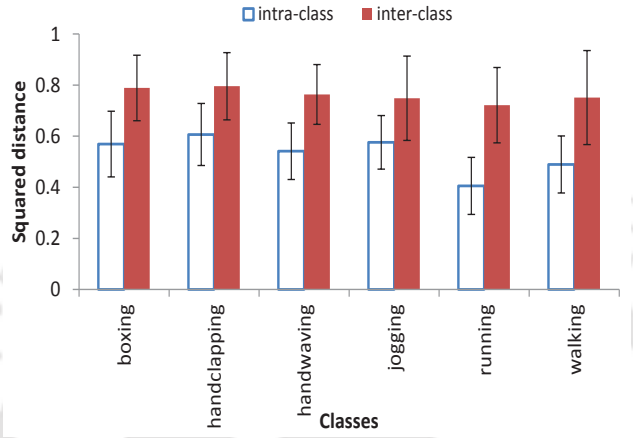
5.2.4.8 Comparison with state-of-the-art

As shown in Table 5.3 and Table 5.4, we can clearly see that the proposed method performed better than the state-of-the-art methods. It also performs better than CNN-based methods because CNN requires a large number of samples to train. Additionally, our method takes only a few hours to train without GPU, whereas CNN-based techniques take days to train even with a GPU. For our method, in line with several contemporary techniques, scale and view invariance were ensured by the diversity of training patterns but not explicitly per se. Empirically, we have demonstrated using UCF11 that our method can deal with action in large frames that may even include additional out-of-class actions in the background.

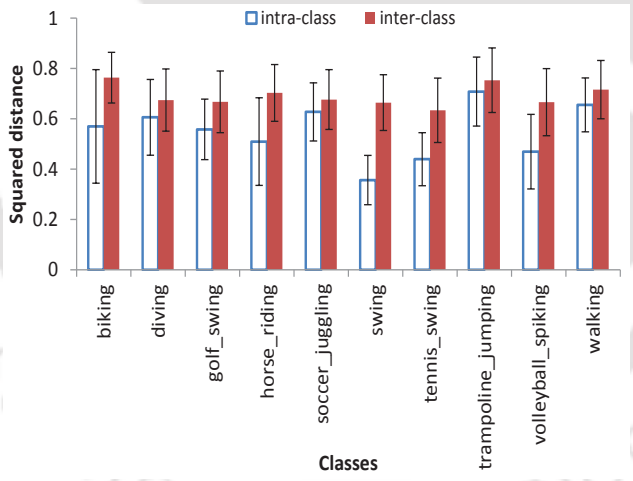
5.2.4.9 Class separation

To show that the proposed features are discriminative, mean-square distances between inter- and intra-class were computed as shown in Fig. 5.6. For KTH, it is clearly visible from the figure that the overlap between inter- and intra-class distances was small which led to a better performance. As UCF11 is a complex dataset, features were less discriminative for some actions such as diving, soccer juggling, trampoline jumping and walking, which affected the performance of the classifiers. Intra-class feature distance was still less compared to inter-class distance for UCF11, which shows that the feature

can separate the classes. This we can also verify from the confusion matrix as shown in Table 5.5 and Table 5.6, classes with larger inter- and intra-class distances have high accuracies compared to the classes with less inter- and intra-class distances.



(a)



(b)

Figure 5.6: Inter- and intra-class mean squared distances (bars) and their variances (error bars) for the proposed video representation for (a) KTH and (b) UCF11 datasets.

5.2.4.10 Difference between recognizing simple vs. complex activities

To further examine the modeling abilities of LRCC, we show class recognition accuracy on UCF11 on three simple and three complex classes in Table 5.2. In simple (first three)

Table 5.2: Class recognition accuracy for different classifiers.

class	LRCC	SVM	RF
golf-swing	100%	98.3%	100%
soccer-juggling	100%	100%	100%
swing	93.3%	88.2%	89.9%
diving	99.2%	88.9%	65.1%
tennis-swing	94.0%	84.3%	89.6%
horse-riding	83.0%	77.3%	75.9%

Table 5.3: Recognition accuracy on KTH dataset.

Method	Average accuracy
Proposed method	94.8%
Wang et al. [106]	94.2%
Kovashka and Grauman [53]	94.5%
Gilbert, Illingworth,and Bowden [32]	94.5%
Laptev et al. [57]	91.8%
Shuiwang et al. [44] (CNN)	90.2%

classes, there is no background motion or action, while in complex (last three) classes, there is other motion present in the scene besides that of the action of interest. We expected LRCC to perform better than the other classifiers on the complex classes due to its ability to model videos as additive combinations of primitives, some of which might represent the actions of interest, while others represent background motion. The results in Table 5.2 are in line with this expectation. Additionally, Table 5.3 and Table 5.4 show that the best results obtained using SSNMF and LRCC are competitive with the state-of-the-art on the two datasets.

We have also shown the confusion matrices for both the datasets in Table 5.5 and Table 5.6 with respect to the best results obtained using SSNMF and LRCC. In case of KTH dataset, first three activities are confusing with each other because of the common hand motion, jogging and running are also confusing with each other because they differ only in terms of speed. Similarly, for UCF11 all complex activities are confusing with each other compared to simple activities.

Table 5.4: Recognition accuracy on UCF11 dataset.

Method	Average accuracy
Proposed method	90.5%
Wang et al. [106]	88.2%
Mahdyar et al. [81] (CNN)	89.5%
Wang et al. [107]	84.2%
Mota et al. [71]	75.4%
Ikizler-Cinbis and Sclaroff [41]	75.2%
Liu, Luo, and Shah [64]	71.2%

Table 5.5: Confusion matrix for KTH dataset.

		Predicted					
		boxing	handclapping	handwaving	jogging	running	walking
Actual	boxing	0.97	0.02	0.01	0	0	0
	handclapping	0.06	0.91	0.03	0	0	0
	handwaving	0.03	0.01	0.96	0	0	0
	jogging	0	0	0	0.93	0	0.7
	running	0	0	0	0.06	0.94	0
	walking	0	0	0	0	0	1

5.2.5 Discussion

We proposed a feature representation based on differential motion map for action recognition. Differential motion captures the motion information very effectively and shows better performance compared to state-of-the-art methods. Front-view ($xy-$plane), side-view ($yt-$plane) and top-view ($xt-$plane) differential motion map captures the action structure (like hands, legs etc.) as well as motion variations, for example visually we can depict the simple actions such as clapping as shown in Fig 5.2. We have also compared the use of different classifiers and dimension reduction techniques. l_2-regularized collaborative representation with a distance-weighted Tikhonov matrix proved to be a better classifier than SVM and RF for our task. Also, SSNMF was found to be better than PCA, which was found to be better than KPCA. This shows the importance of using an

Table 5.6: Confusion matrix for UCF11 dataset.

Actual \ Predicted	basketball	biking	diving	golf-swing	horse-riding	soccer-juggling	swing	tennis-swing	trampoline-jumping	volleyball-spiking	walking-with-dog
basketball	0.94	0.01	0.01	0	0.01	0	0	0.01	0	0	0.02
biking	0	0.80	0	0	0.07	0.01	0	0.01	0.01	0	0.1
diving	0	0	1	0	0	0	0	0	0	0	0
golf-swing	0	0	0	1	0	0	0	0	0	0	0
horse-riding	0	0.01	0	0	0.83	0	0	0.01	0	0	0.15
soccer-juggling	0	0	0	0	0	1	0	0	0	0	0
swing	0	0.03	0	0	0.03	0	0.93	0	0	0	0.01
tennis-swing	0	0	0	0	0.04	0	0	0.94	0	0	0.02
trampoline-jumping	0	0	0.01	0	0.05	0	0.01	0	0.89	0.01	0.03
volleyball-spiking	0.01	0	0	0	0.12	0	0	0	0	0.86	0.01
walking-with-dog	0	0.13	0.02	0	0.21	0	0.02	0	0.02	0.01	0.59

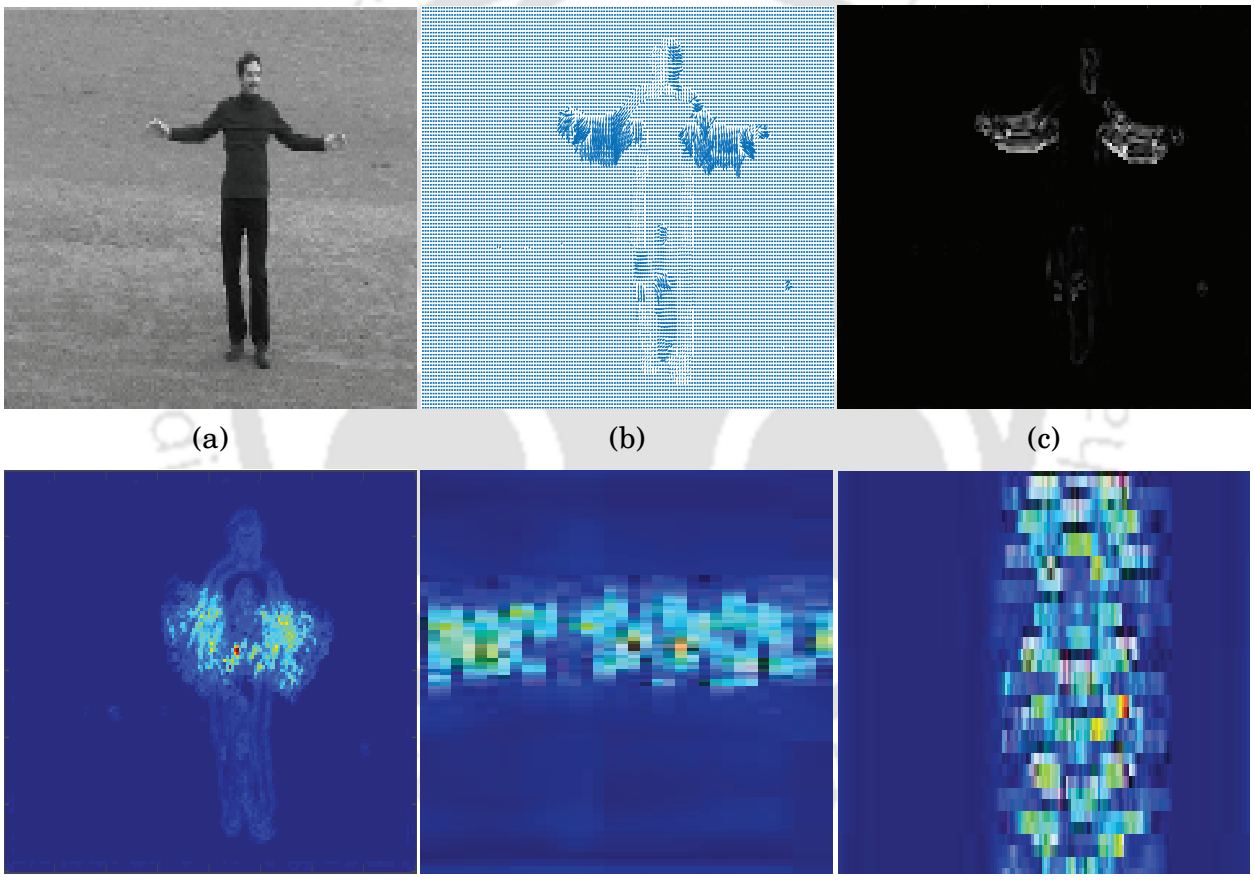
appropriate classifier based on the features at hand. A comparison of differential motion and optical flow based feature was also done to show that differential motion gives better feature representation. The proposed method was tested on two popular datasets – KTH and UCF11. This technique or its variants can be tested on larger datasets.

5.3 Proposed Method 2

Inspired by method 1, we proposed this method with the slight change in mathematics. Our main idea for capturing unique features of each action class is to capture motion between an actor and the background, as well as changes in that motion from frame to frame. Optical flow is not enough to distinguish between different action classes, especially when the camera is moving because it gets overwhelmed by the motion of the background with respect to the camera. On the other hand, spatio-temporal points of high acceleration represent the change in the direction of motion or onset or end of

an atomic action. Such points are vital in defining an action. For example, in clapping, the moment when the hands start moving towards each other, or when they suddenly stop after coming together are iconic. Therefore, the proposed algorithm is based on the computation of differential motion. Differential motion maps capture structure and shape information. We obtain the same results with this simple math compared to method 1.

The proposed method has two major parts: feature extraction and classification.



(a) (b) (c)

(d) (e) (f)

Figure 5.7: (a) Example frames for hand-clapping video sequence, (b) optical-flow, (c) divergence magnitude of optical-flow (d) Front view ($xy-$plane) differential motion map, (e) Side view ($yt-$plane) differential motion map, (f) Top view ($xt-$plane) differential motion map.

5.3.1 Feature extraction

To compute motion information we use optical flow based on Lucas-Kanade method [68]. For each frame in the video, the flow is computed. Optical flow (velocity) vector $\vec{V}$ of a point in a frame is expressed as a linear combination of unit vectors in $\vec{i}$ and $\vec{j}$ in x and y directions respectively as $V_1\vec{i} + V_2\vec{j}$. Divergence is a differential operation on the vector field defined by flow map as follows:

$$R(x,y,t) = \nabla.\vec{V}(x,y,t) = \frac{\partial V_1(x,y,t)}{\partial x} + \frac{\partial V_2(x,y,t)}{\partial y} \tag{5.14}$$

The magnitude of divergence of optical flow is usually high along the edges of a moving object when those edges are moving perpendicular to the tangent of the edge relative to their background, as shown in Fig. 5.7. This is because there is a unidirectional motion on one side of the point on the edge, which gives a net non-zero contribution to the line integral used for computing divergence around the point. Then, a point-wise absolute difference of divergence of pairs of consecutive frames as shown in Equation 5.15 to capture the change in motion, where, without loss of generality, we assume that the frame timestamps are represented by integers.

$$D_f(x,y,t) = |R(x,y,t+1) - R(x,y,t)| \tag{5.15}$$

To compress the information, we project the divergence map for each frame onto the three orthogonal Cartesian planes defined by (x,y,t) coordinates as proposed in [13]. This absolute difference of divergence projected to the three planes is accumulated for each pair of consecutive frames for the entire video. Thus, the entire video is used to generate three 2D projected maps corresponding to front, side and top views as shown in Fig. 5.7. These differential motion maps are analogous to, yet different from, depth motion maps [13]. For the front view in which the dimension t is eliminated, this operation is defined

as:

$$DMM_{front}(x,y) = \sum_{t=1}^{T-1} D_f(x,y,t) \tag{5.16}$$

Similarly for Side and top view:

$$DMM_{side}(y,t) = \sum_{x=1}^{m} D_f(x,y,t) \tag{5.17}$$

$$DMM_{top}(t,x) = \sum_{y=1}^{n} D_f(x,y,t) \tag{5.18}$$

To deal with different sizes of motion maps for different actions, bi-cubic interpolation was used to resize all projection views to a fixed size. All values of the maps were normalize between 0 and 1 to avoid large values dominating the feature set, and to normalize for video length N. The size of DMMs for front side and top views were $m_f \times n_f$, $m_s \times n_s$, $m_t \times n_t$ respectively. For an action video sequence, a feature vector of size $(m_f \times n_f + m_s \times n_s + m_t \times n_t) \times 1$ was formed by concatenating the vectorized differential motion maps of three views.

From differential motion maps in Fig. 5.7 we can easily see that the movements of hands have been captured for hand-clapping video sequence in case of the front, side and top views. In Fig. 5.3, the horizontal axis denotes the width and vertical axis denotes the height of the video sequence and high values at the center shows the hand-clapping motion from front-view. Similarly in Fig. 5.3, the horizontal axis denotes the time dimension and vertical axis denotes the height of the video sequence and high values along the horizontal direction shows the position of hand motion from side-view. In Fig. 5.3, the horizontal axis denotes the width of the video sequence and vertical axis denotes the time dimension and high values along the vertical direction shows the position of hand motion from the top-view.

5.3.2 Classification

As evident from our feature design, the feature is a linear additive mixture of individual features associated with various atomic actions in the videos due Equations 5.15–5.18. For classification l_2-regularized collaborative classifier [13] (LRCC) is used, which is explained in section 5.2.3.1.

5.3.3 Experiments and Results

5.3.3.1 Experimental setup

Two popular datasets were used for the experiments. KTH is simple action database which consist of 6 actions including boxing, hand-clapping, hand-waving, jogging, running and walking [89]. UCF11 is a complex action database which consists of 11 actions including basketball shooting, biking, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog [64]. For KTH dataset we followed the same procedure as mentioned in [89]. We divided the dataset into training and testing sets based on the subjects. For UCF11, It has been suggested to use entire groups instead of splitting their constituent clips into training or testing sets (specifically, leave-one-group-out cross-validation (LOOCV)) to test the robustness of action recognition techniques to unseen scene variations [64]. Classification of actions was done using LRCC. The accuracy was averaged over several random selections of training and testing data. For UCF11 dataset we have followed the LOOCV approach, which leads to 25 cross-validation results for each action class, which were averaged.

Principal component analysis (PCA) was applied to reduce the dimensionality of the features. The PCA transform matrix was obtained using the training feature and then applied to the test features. The largest eigenvalues that explained 85% of the variance were kept. This reduced the dimension from approximately 17,000 to 400. Dimensionality reduction improved the computational efficiency without affecting the performance. We have considered the dimension after which accuracy becomes constant. Regularization

Table 5.7: Recognition accuracy for KTH and UCF11 datasets.

Method	KTH	UCF11
Proposed method	96.98%	90.24%
Yadav, Shukla, and Sethi [118]	98.2%	91.3%
Kovashka and Grauman [53]	94.53%	90.45%
Gilbert, Illingworth,and Bowden [32]	94.50%	–
Wang et al. [107]	94.20%	84.20%
Laptev et al. [57]	91.80%	–
Shuiwang et al. (CNN) [44]	90.2%	–
Mahdyar et al. (CNN) [81]	–	89.5%
Ikizler-Cinbis and Sclaroff [41]	–	75.21%
Liu, Luo, and Shah [64]	–	71.20%

parameter λ was tuned by grid search using cross-validation on training datasets.

5.3.3.2 Recognition results

Comparison with the state-of-the-art: We compare our results with the state-of-the-art methods, as shown in Table 5.7. The proposed method performed better than the state-of-the-art methods for KTH as well as UCF11 datasets. LRCC performs better because LRCC models a video as a linear combination of several prototype feature vectors from the training data. This allows it to decompose a video,Äôs feature vector into its components arising from both in-class and out-of-class actions in the same video. The coefficient of in-class component helps it classify better for complex videos with out-of-class actions such as those in UCF11. In line with several contemporary techniques, scale and view invariances are ensured by the diversity of training patterns but not explicitly per se. Empirically, we have demonstrated using UCF11 that our method can deal with action in large frames that may even include additional out-of-class actions in the background. Our method performs outperformed CNN-based methods [44, 81] because CNN requires a large number of samples for training also training time is more in case of CNN. Here, we focus on the small dataset where CNN does not perform well.

5.3.4 Discussion

We proposed a feature representation based on differential motion map for action recognition. Differential motion captures the motion information very effectively and shows better performance compared to state-of-the-art methods. Differential motion maps capture the action structure as well as motion. A comparison of differential motion and optical flow based feature was also done to show that differential motion gives better feature representation.

Both methods give approximately same results because only front motion maps contribute most to the feature, side and top view doesn't give much information in both the cases. The proposed method was tested on two popular datasets KTH and UCF11. Our future goal is to test it on larger datasets.

CHAPTER 6

HUMAN ACTION RECOGNITION USING DEEP LEARNING

In this chapter, we have proposed a method to detect abnormal events for human group activities. Our main contribution is to develop a strategy that learns with very few videos by isolating the action and by using supervised learning. First, we subtract the background of each frame by modeling each pixel as a mixture of Gaussians(MoG) to concatenate the higher order learning only on the foreground. Next, features are extracted from each frame using a convolutional neural network (CNN) that is trained to classify between normal and abnormal frames. These feature vectors are fed into long short term memory (LSTM) network to learn the long-term dependencies between frames. The LSTM is also trained to classify abnormal frames, while extracting the temporal features of the frames. Finally, we classify the frames as abnormal or normal depending on the output of a linear SVM, whose input is the features computed by the LSTM.

6.1 Introduction

Nowadays significant amount of research is being done in the field of automated video surveillance for personnel and asset safety. The need for automation of analysis of

surveillance video is increasing day-by-day to reduce the manual workload. Person detection, tracking, activity recognition, and event recognition are the areas where researchers are focusing. Working with multiple cameras has its own set of challenges such as tracking a person from one camera to other, group activity recognition etc. Public areas can be monitored with automated systems with less manpower. However, detecting an abnormal event from a given video is a challenging problem that is highly contextual.

We propose a method to design a trainable surveillance system to enable situational awareness and determination of potential threats on mobile assets (e.g. vehicles) in transit. BMTT-PETS 2017 Surveillance Challenge is aimed at identifying various abnormal activities which occur in real-world scenarios. PETS 2016 dataset [74], also known as ARENA dataset, is used in this paper. The dataset scenarios were recorded from multiple cameras mounted on a vehicle (truck) and involve multiple people demonstrating the event.

There are various challenges in activity recognition in naturally captured videos. Just a one-minute-long video of frame size 640×480 and captured at 30 frames per second can have about half a billion pixels. For the same action class, color and intensities of these pixels can change due to variations in environments, viewpoints, noise, and actor movements. Variations in environments are caused by moving background, occlusion, lighting, and co-occurring of actions of interest with confounding secondary activities. Videos of the same action class taken from different viewpoints have high intra-class variation. For example, a video of walking a dog on the grassy background has more pixels in common with that of the sport being played on a grassy field than with that of walking a different dog on an urban street. Further, many action classes appear similar, such as walking, jogging, and running, which differ mostly in speed and stride length. Lighting and sensor differences also contribute to variations in pixels of videos of the same action. Common video transformations such as compression and scaling while storing or uploading the video also add to variations in videos of the same action. Additionally, actions of the same class are seldom performed identically in space and time even by the same actor.

In this paper, we are focusing on one of the atomic activities given in the ARENA dataset. The dataset covers a variety of tasks involving low-level video analysis (detection, tracking), mid-level analysis (simple event detection) and high-level analysis (complex "threat" event detection).

The remainder of this paper is organized as follows: Section 6.2 describes the proposed method followed by experiment and result in Section 6.2.5 and conclusion in Section 6.3.

6.2 Abnormal Event Detection on BMTT-PETS 2017 Surveillance Challenge

We have focused on a single atomic level activity detection given in BMTT-PETS surveillance dataset which is "person falling or pushed to the ground". In this section, we discuss the proposed method as shown in Fig. 6.1. Due to the small number of videos available for this activity we had to make several changes to the usual ways of applying deep learning. We have filtered uninformative parts of the videos, followed by supervised higher-order feature extraction. First, we subtract the background to remove unwanted static visual features. Then we use a Convolutional Neural Network for feature extraction followed by an LSTM network for sequence learning.The output of the last fully connected layer in CNN is passed as an input to the LSTM network. Finally, we use a linear SVM to get the classification scores. To further improve the discrimination between normal and abnormal frames that the entire network has already predicted, we have performed temporal averaging on the final predicted scores. Our method demonstrates how non-deep learning-based methods such as background subtraction and linear SVMs can be combined with deep learning frameworks such as CNN and LSTM to build machine learning systems with fewer data.

Figure 6.1: Block diagram of the proposed method.

6.2.1 Background subtraction

Mixtures of Gaussians are used for background subtraction as proposed by [47]. Each pixel in the scene is modeled by a mixture of K Gaussian distributions. The probability that a certain pixel has a value of X_N at time N can be written as:

$$p(x_N) = \sum_{j=1}^{K} w_j \eta(x_N; \theta_j) \tag{6.1}$$

where W_k is the weight parameter of the k^{th} Gaussian component. $\eta(x;\theta_k) = \eta(x;\mu_k,\Sigma_k)$ is the normal distribution of the k^{th} component represented by:

$$\eta(x;\mu_k,\Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)} \tag{6.2}$$

where μ_k is the mean and $\Sigma_k = \sigma_k^2 I$ is the covariance of the k^{th} component.

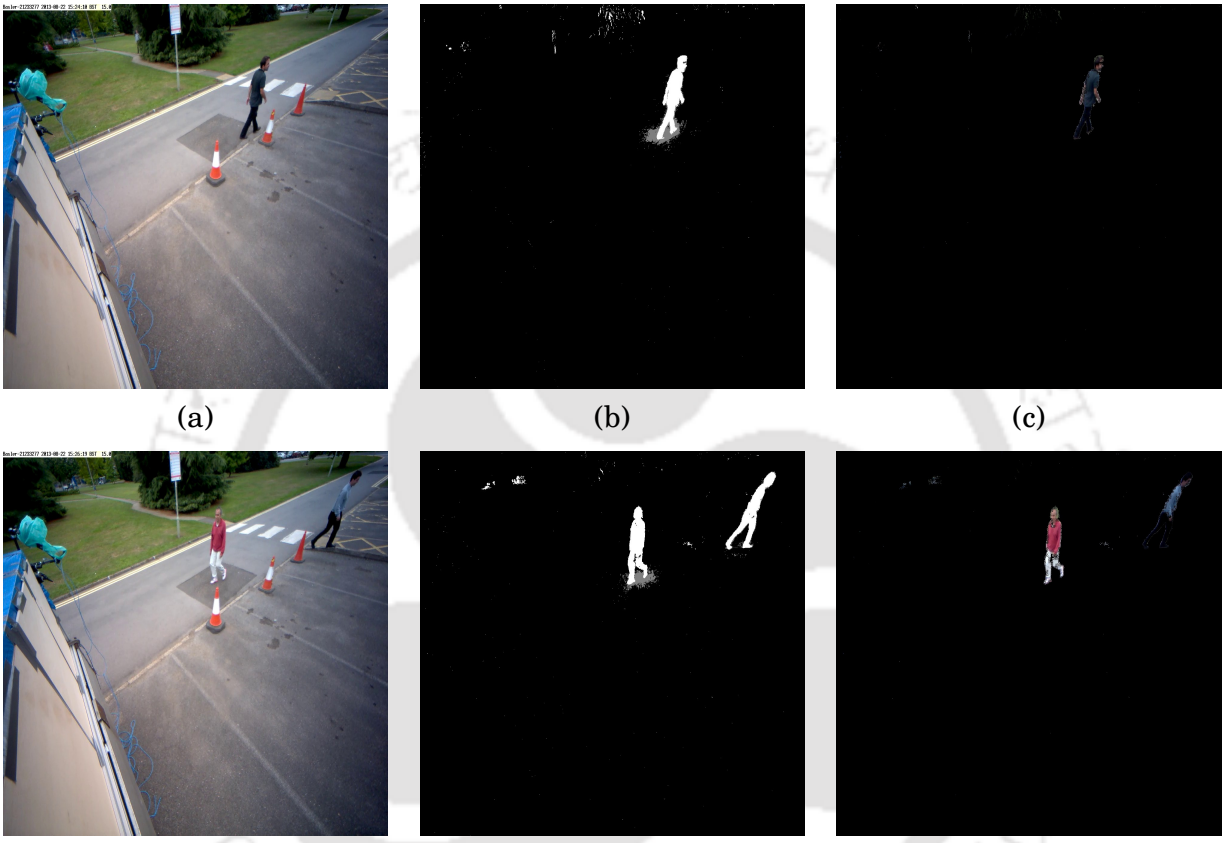
The first B distributions are used as mixtures of the background model of the scene, where B is estimated as:

$$B = \arg\min_b \left(\sum_{j=1}^{b} w_j > T \right) \tag{6.3}$$

The threshold T is the minimum fraction of the background model. In other words, it is the minimum prior probability that the background is in the scene. Background subtraction is performed by marking a foreground pixel as any pixel that is more than 2.5 standard deviations away from any of the B distributions. The first Gaussian component

that matches the test value will be updated by the following update equations:

In Fig. 6.2 images are shown after background subtraction. We can clearly see that the moving object has been separated from the background, which removes unwanted information.



(a) (b) (c)

(d) (e) (f)

Figure 6.2: Example frames for Background subtraction, (a)-(d) Original frame (b)-(e) Binary image after background subtraction, (c)-(f) Background subtraction overlapped with original.

6.2.2 Spatial feature extraction

Features from the raw input frames are extracted using a deep CNN with the same architecture as VGG-16 [92] as shown in Fig. 6.3. This network is employed to extract spatial features as well as for high accuracy image recognition, which is crucial when the frames have distinguishable abnormalities or objects. The network contains 16

trainable (convolutional and fully connected) layers along with several static pooling and dropout Layers. In [92], the authors have shown that the depth of the network plays an important role in its performance. Unnecessary addition of extra layers may not significantly improve the performance while increasing the computational complexity.

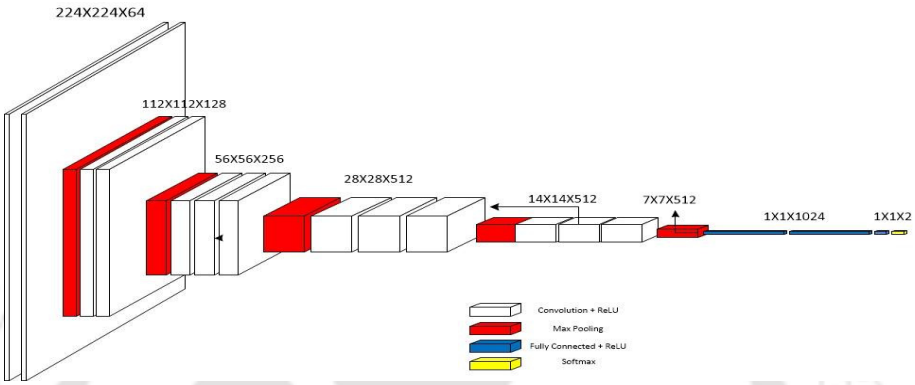


Figure 6.3: CNN Architecture based on VGG16.

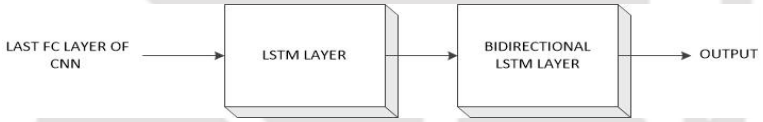


Figure 6.4: LSTM architecture.

6.2.3 Temporal feature extraction

To further improve the model and recognition accuracy, we make use of the temporal relationship between the frames by passing the output of last fully connected layer as input to an LSTM network as shown in Fig. 6.4. An LSTM specializes in learning long-range dependencies while being unaffected by diminishing or exploding gradient issues that affect earlier recurrent neural networks when trained using backpropagation through time. The non-linear activation gates manipulate the amount of information being stored in the memory cells. Our LSTM architecture consists of two layers. The first layer is made up of 1024 hidden units followed by a bi-directional second layer with 512 hidden units. The standard recurrent neural networks have restrictions on the amount

of data that is provided as input. i.e. they have limitations on the input data flexibility. In such networks, the future input information cannot be reached from the current state. To overcome this problem, we use a bidirectional LSTM where we have a forward LSTM and a backward LSTM running in reverse time with their features concatenated at the output layer, thus allowing us to combine the useful features of the past and the future.

6.2.4 Classification

For classification of the event frames, we tried two methods: the average fusion of the networks and a linear SVM classifier. While classifying using average fusion, we have averaged the outputs of CNN's and LSTM's softmax layers and predicted the abnormalities based on the scores. We also tried a linear SVM classifier. SVM has been proven to be a state-of-the-art linear classifier which maximizes the margin between two classes. More importantly, it works well with high-dimensional and low sample size datasets. We have trained the SVM using one-vs-all approach using scikit learn libraries [27].

6.2.5 Experiment and Results

In this section, we describe the datasets, experimental setting for training and testing followed by recognition results.

6.2.6 Datasets

The provided dataset is a multi-sensor dataset, as used for PETS 2014 to PETS 2016, which addresses protection of trucks (the ARENA Dataset). The dataset includes a set of abnormal events such as the person falling on ground, the person speeding up, the person loitering, the person suddenly changing directions etc. We focus only on the "person falling or pushed to the ground" event detection. There are three folders – 11-04, 11-03 and 08-02 which depict the different ways in which a person can fall on the ground. Each folder has frame sequences of videos from four different cameras mounted on the

Table 6.1: Labeling of the Start and End frames for each folder

Folder	Start frame	End frame
11-04 TRK-RGB1	1377185170222	1377185175622
11-03 TRK-RGB1	1377185040756	1377185045090
08-02 TRK-RGB2	1377181598164	1377181608514

truck. We have classified the frames into two classes – "normal" and "abnormal"(where the person of interest is falling) and labeled them as shown in Table 6.1. The task was to predict the starting and ending frames of the abnormal activity sequence that is taking place. For cross-validation or testing, the data was split at the folder level because each folder contained video frames from a different scene (e.g. parking location, the background of the truck etc).

6.2.6.1 Results and Discussion

We have experimented with different combinations of CNN, LSTM, and SVM. Each combination is explained as follows:

6.2.7 CNN

We have used the CNN architecture as shown in Fig. 6.3 and tested on folders 11-04 (TRK-RGB1), 11-03 (TRK-RGB1), 08-02 (TRK-RGB2) consisting of 729, 329, 1056 frames respectively, taking into consideration only the cameras which capture the abnormal activity that is occurring. The VGG-16 CNN architecture is first trained from scratch on 11-04 (TRK-RGB1 and TRK-RGB2) and 11-03 (TRK-RGB1 and TRK-RGB2) data and then tested on 08-02 (TRK-RGB2). We repeat the same process by initializing the model with new weights but by training on 08-02(TRK-RGB1 and TRK-RGB2), 11-04 (TRK-RGB1 and TRK-RGB2) from scratch and testing the model on 11-03 (TRK-RGB1). Similarly, we train on 11-03(TRK-RGB1 and TRK-RGB2), 08-02(TRK-RGB1 and TRK-RGB2) and test on 11-04 (TRK-RGB1). The results for all the cases are mentioned in the Table 6.2.

Discussion: We have not used pre-trained weights for VGG-16 convnet because the

pre-trained model has already adapted itself to the huge number of classes (1000) and it would be difficult for the network to learn the weights when we are performing a binary classification with a considerably small dataset. Thus, training the network from scratch gave better results compared to those using the pre-trained model.

6.2.8 CNN+SVM

In Section 6.2.7, we have used the softmax layer as the output layer of CNN for all training and testing cases. In this section, we explore the benefits of adding a support vector machine for classification of the features determined using CNN. The methodology of training and testing remains the same, as mentioned in Section 6.2.7.

Discussion: Since support vector machines are robust to a decrease in the number of training samples for classification, incorporating it as the output layer improved results over softmax as shown in Table 6.2.

6.2.9 CNN+LSTM

We have incorporated LSTM network on top of our CNN architecture to model the temporal information in the visual channel. LSTMs are one of the most widely used RNN models that learn long-range dependencies of the sequential data by incorporating memory cells. These cells are protected by non-linear gates (with $tanh$ or sigmoidal activation functions) which makes LSTMs immune to vanishing and exploding gradients, unlike the traditional RNNs. These gates control the amount of information that needs to be stored in memory, forgotten and passed on to the next hidden unit or to the output layer. Therefore, after adding LSTM and using softmax as a classifier, we can see the improvement over CNN and CNN-SVM.

Discussion: In the second layer of LSTM network we have used the bi-directional LSTM hidden units instead of the usual hidden units. The reason being: bi-directional LSTM layer acts as a combination of both a forward LSTM and a backward LSTM which runs reverse in time and the features of both are merged to get the output. Through this method, we predict the output based on the information from the past and the future.

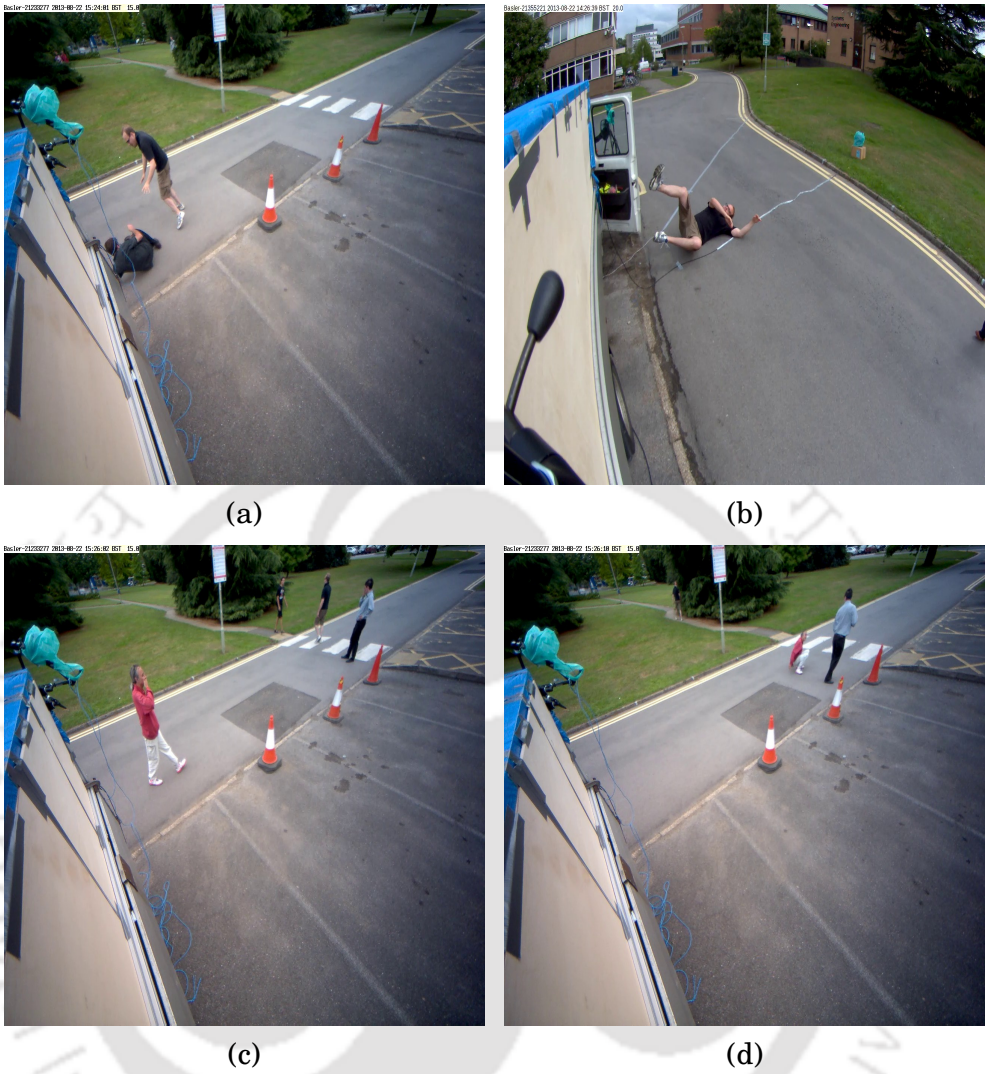


Figure 6.5: Example frames from datasets (a) 11-03, (b) 08-02, (c) 11-04 normal frame and (d) 11-04 abnormal frame.

Therefore, recognition accuracy further increases compared to previous combinations as shown in Table 6.2.

6.2.10 CNN+LSTM+SVM

Since we are working on a very small dataset compared to other video classification problems consisting of hours of videos as the training data, the usage of SVM's proves to be beneficial. We incorporate a final SVM layer on top of the CNN+LSTM architecture so as

Table 6.2: Performance comparison on the given data-sets.

Method	11-04	11-03	08-02
CNN	83.3%	80%	80%
CNN+SVM	83.8%	80.7%	80.1%
CNN+LSTM	84.2%	81.3%	80.2%
CNN+LSTM+SVM	85.3%	83.4%	82.9%
CNN+LSTM+SVM+TA	85.3%	96.4%	94.9%

Table 6.3: Start and end frame for CNN-LSTM-SVM-TA.

Folder	Start frame	End frame
11-04 TRK-RGB1	1377185162089	1377185163755
11-03 TRK-RGB1	1377185040889	1377185045356
08-02 TRK-RGB2	1377181598664	1377181606314

to learn the final decision boundary between the normal and abnormal sequences/frames. This gives the best results compare to CNN, CNN-SVM, and CNN-LSTM as shown in Table 6.2.

Discussion: We now incorporate the best performing networks on top of each other for final classification of the frames. We observe that the proposed classification method performs better than the average fusion method, where the predicted probability outputs of CNN and LSTM softmax layers are merged and averaged to predict the class.

6.2.11 CNN+LSTM+SVM+TA

To remove the discontinuity in the prediction of frames we have used temporal averaging after the CNN+LSTM+SVM model. For temporal averaging we choose a particular length of frames and based on the number of majority predictions, we change the predicted label of the entire length of frames to the majority label.

6.2.12 Results

For training VGG-16 we have used the same setting as the author has suggested in [92], but with few changes. The input size is $224 \times 224 \times 3$ and dropout regularization for the two fully-connected layers was 0.5. A change in the number of neurons in the

last two Fully Connected Layers was made and was set to 1024 instead of 4096. The learning rate was initially set to 10^{-4}, then decreased by a factor of 10. We fine-tuned on any two folders using the normal vs. abnormal frame label and tested on the third one for cross-validation at the folder level. Hence, we get three accuracy number for three folders as shown in the Table 6.2. From Table 6.2 we can see that adding LSTM to CNN improves the results and further adding SVM again improves the results. The starting and ending frames are also mentioned in the Table 6.3. 11-03 and 08-02 have shown correct predictions of frames. But in case of 11-04, it fails to identify the correct frames because it is different from the other two datasets. The model has not seen such type of activity that far away from the camera and for a very short duration when compared to other two folders. As shown in Fig. 6.5, the abnormal activity is happening in the vicinity of the vehicle in case of 11-03 and 08-02 unlike 11-04. Therefore, the network is unable to classify correctly in case of 11-04 frames and it classifies those frames as abnormal where the person is walking near the vehicle.

To further improve the classification accuracy and to predict the start and end frames as close as possible to the ground truth, we have performed temporal averaging on the predictions of the above-mentioned network (CNN+LSTM+SVM+TA). We observe that the accuracies on 11-03 and 08-02 jumps to 96 and 95 percent respectively, and the predicted start and end frames are very close to the ground truth. This method is not applicable to 11-04 because the initial predictions don't cover the abnormal event.

6.3 Discussion

Among all the combinations, CNN-LSTM-SVM-TA model has shown the best performance. CNN perform best when a large number of training samples are provided for training, whereas for less number of samples it can be used for feature extraction instead of classification. These features mostly contain spatial dependencies among frames but not the temporal ones. Therefore, to overcome these issues, we combined CNN features with an LSTM model and an SVM classifier. LSTM learns temporal relationships be-

tween frames and the SVM classifies the frames independent of the number of redundant samples. SVM can thus be used instead of softmax for classification. We have evaluated these models for one type of activity, but it can also be further extended to other activities.

CHAPTER 7

SUMMARY AND DISCUSSION

This chapter concludes this thesis. In Section 7.1 we summarize our contributions in the field of vision-based human activity analysis and discuss the limitations of the proposed methods in section 7.2. This evaluation serves as a starting point for further discussion concerning future research in the field, which is presented in section 7.3.

7.1 Summary

In this thesis, we have done a qualitative and quantitative analysis of spatial and temporal dimension, where we show that the spatial and temporal dimensions are different. The temporal dimension is different from depth. Most state-of-the-art methods treat temporal dimension as depth which is inappropriate. To prove this hypothesis we have proposed interest point detectors which are different from traditional interest point detectors.

In the first method, we densely sampled the pixels to define interest point in a frame. We use optical flow to track those points and form trajectories of moving pixels which gives spatially localized points. Then we using RDP algorithm we approximate

trajectories with line segments to get temporally localized points.

In the second method, we detect interest points using magnitude of curl of optical flow. An adaptive threshold is used to detect interest points over different spatial scales. The robustness of detector is shown through various experimentation which shows high repeatability under common video transformations such as scaling and compression. For human action recognition, we have proposed two representations. First is based on the histogram of temporally localized points on trajectories. We track each point to form trajectories then temporal localization is done based on the change in the direction of trajectories. If the change is more than a threshold we keep that point otherwise we discard it. Then we form a histogram of orientations of line joining those temporally localized points. This histogram is then used for action recognition. In the second representation, we modified the old descriptors such as HOG-HOF. We added location information of interest points which was not used before. We form a histogram based on the orientation of neighbouring interest point with respect to a point, which is then concatenated with HOG-HOF. Action recognition results shows improvements over HOG-HOF methods.

We have also proposed a feature representation based on global approach. Differential motion maps were calculated and projected onto the Cartesian planes. These maps were than vectorize and concatenated to form the feature vector. The dimension of feature vector is then reduced by using PCA, KPCA and SSNMF. We have compared these dimension reduction techniques and found that SSNMF gives better results because it considers the labels. For classification we have used SVM, RF, and LRCC. We have compared these classifiers and shown that LRCC is better than SVM and RF because it models the complex actions as the mixture of simple actions.

As deep learning is getting popular nowadays, we have also tried different combinations of CNN and LSTM for abnormal event detection. Our CNN model was trained first using BMTT-PETS dataset then LSTM network was trained using the output of CNN as input to LSTM. After training features are extracted using CNN-LSTM. The last layer of LSTM is treated as the feature vector. SVM classifier is used for classification as the

data size is very less. Based on the experiments we can suggest that for low sample size SVM can be used for classification and CNN for feature extraction.

7.1.1 Key Contributions

Following are the key contributions:

1. Proposed a spatio-temporal interest point detector for videos based on differential motion.
2. Proposed a modification in the descriptor for interest point based on the location of interest point.
3. Proposed a feature representation using differential motion for human action recognition.
4. Proposed a combination of CNN-LSTM-SVM for abnormal event detection.

7.2 Limitations

The main limitation of interest point based method is that it uses hand-crafted features that are suitable for small datasets, but for large and complex datasets CNN based approaches perform better. Therefore, proposed method performs better for small datasets such as KTH, Weizmann and UCF11. The number of interest point varies from video to video because of the motion content due to which number of descriptor also varies. Hence, while constructing codebook if the per-class descriptor is not balanced it might affect the final representation.

In case of differential motion maps, the extracted feature is based on global information. Therefore, for complex activities, the information content is not very rich compared to single person activity. LRCC classifier models the complex activities to some extent.

In case of CNN-LSTM model, the dataset size is a limitation. To train these models a large amount of datasets is required. In case BMTT-PETS the size of the dataset is very

small. Use of SVM gives good results, but the feature was not properly trained because of low sample size.

7.3 Future Work

Based on the limitations we are proposing the following future scope:

1. In future, hand-crafted features and CNN based features can be combined together to boost the performance.
2. In [106] dense trajectories have shown better results for a given interest point detector. Same can be done here for the interest point detector proposed in Chapter 4.
3. In case of abnormal event detection, training with the large dataset and using SVM as classifier might give some more improvement in results due to the wide margin classification properties of an SVM.
4. Try ReProCS (a robust subspace tracking algorithm) for removing the background and apply the interest point detector for action recognition task. Check if performance improves for all of them or some of them.

BIBLIOGRAPHY

[1] M. AHMAD AND S.-W. LEE, *Hmm-based human action recognition using multiview image sequences*, in Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 1, IEEE, 2006, pp. 263–266.

[2] L. BALLAN, M. BERTINI, A. DEL BIMBO, L. SEIDENARI, AND G. SERRA, *Effective codebooks for human action representation and classification in unconstrained videos*, IEEE Transactions on Multimedia, 14 (2012), pp. 1234–1245.

[3] P. R. BEAUDET, *Rotationally invariant image operators*, in Proc. 4th Int. Joint Conf. Pattern Recog, Tokyo, Japan, 1978, 1978.

[4] M. BLANK, L. GORELICK, E. SHECHTMAN, M. IRANI, AND R. BASRI, *Actions as space-time shapes*, in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 2, IEEE, 2005, pp. 1395–1402.

[5] U. M. BRAGA-NETO AND E. R. DOUGHERTY, *Is cross-validation valid for small-sample microarray classification?*, Bioinformatics, 20 (2004), pp. 374–380.

[6] D. BREZEALE AND D. J. COOK, *Automatic video classification: A survey of the literature*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38 (2008), pp. 416–430.

[7] T. BROX AND J. MALIK, *Large displacement optical flow: descriptor matching in variational motion estimation*, IEEE transactions on pattern analysis and machine intelligence, 33 (2011), pp. 500–513.

[8] A. A. CHAARAOUI, P. CLIMENT-PÉREZ, AND F. FLÓREZ-REVUELTA, *Silhouette-based human action recognition using sequences of key poses*, Pattern Recognition Letters, 34 (2013), pp. 1799–1807.

[9] A. A. CHAARAOUI AND F. FLÓREZ-REVUELTA, *A low-dimensional radial silhouette-based feature for fast human action recognition fusing multiple views*, International scholarly research notices, 2014 (2014).

[10] B. CHAKRABORTY, M. B. HOLTE, T. B. MOESLUND, J. GONZALEZ, AND F. X. ROCA, *A selective spatio-temporal interest point detector for human action recognition in complex scenes*, in Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1776–1783.

[11] C.-C. CHANG AND C.-J. LIN, *Libsvm: a library for support vector machines*, ACM transactions on intelligent systems and technology (TIST), 2 (2011), p. 27.

[12] S. CHEEMA, A. EWEIWI, C. THURAU, AND C. BAUCKHAGE, *Action recognition by learning discriminative key poses*, in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 1302–1309.

[13] C. CHEN, K. LIU, AND N. KEHTARNAVAZ, *Real-time human action recognition based on depth motion maps*, Journal of real-time image processing, 12 (2016), pp. 155–163.

[14] M.-Y. CHEN AND A. HAUPTMANN, *Mosift: Recognizing human actions in surveillance videos*, (2009).

[15] W. CHEUNG AND G. HAMARNEH, *n-sift: n-dimensional scale invariant feature transform*, IEEE Transactions on Image Processing, 18 (2009), pp. 2012–2021.

[16] S. CHUN AND C.-S. LEE, *Human action recognition using histogram of motion intensity and direction from multiple views*, IET Computer Vision, 10 (2016), pp. 250–257.

[17] J. COOPER, S. VENKATESH, AND L. KITCHEN, *Early jump-out corner detectors*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15 (1993), pp. 823–828.

[18] C. CORTES AND V. VAPNIK, *Support-vector networks*, Machine learning, 20 (1995), pp. 273–297.

[19] A. CRIMINISI, J. SHOTTON, AND E. KONUKOGLU, *Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning [internet]*, Microsoft Research, (2011).

[20] L. DENG, D. YU, ET AL., *Deep learning: methods and applications*, Foundations and Trends® in Signal Processing, 7 (2014), pp. 197–387.

[21] 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, IEEE, 2005, pp. 65–72.

[22] D. H. DOUGLAS AND T. K. PEUCKER, *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature*, Cartographica: The International Journal for Geographic Information and Geovisualization, 10 (1973), pp. 112–122.

[23] L. DRESCHLER AND H.-H. NAGEL, *On the selection of critical points and local curvature extrema of region boundaries for interframe matching*, in Image sequence Processing and Dynamic Scene Analysis, Springer, 1983, pp. 457–470.

[24] Y. DU, W. WANG, AND L. WANG, *Hierarchical recurrent neural network for skeleton based action recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1110–1118.

[25] A. A. EFROS, A. C. BERG, G. MORI, AND J. MALIK, *Recognizing action at a distance*, in Ninth International Conference on Computer Vision, Nice, France, IEEE, 2003, p. 726.

[26] A. EWEIWI, S. CHEEMA, C. THURAU, AND C. BAUCKHAGE, *Temporal key poses for human action recognition*, in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 1310–1317.

[27] R.-E. FAN, P.-H. CHEN, AND C.-J. LIN, *Working set selection using second order information for training support vector machines*, Journal of machine learning research, 6 (2005), pp. 1889–1918.

[28] A. FATHI AND G. MORI, *Action recognition by learning mid-level motion features*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[29] P. FOGGIA, A. SAGGESE, N. STRISCIUGLIO, AND M. VENTO, *Exploiting the deep learning paradigm for recognizing human actions*, in Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on, IEEE, 2014, pp. 93–98.

[30] W. FÖRSTNER, *A framework for low level feature extraction*, in European Conference on Computer Vision, Springer, 1994, pp. 383–394.

[31] W. FÖRSTNER AND E. GÜLCH, *A fast operator for detection and precise location of distinct points, corners and centres of circular features*, in Proc. ISPRS intercommission conference on fast processing of photogrammetric data, 1987, pp. 281–305.

[32] A. GILBERT, J. ILLINGWORTH, AND R. BOWDEN, *Action recognition using mined hierarchical compound features*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2011), pp. 883–897.

[33] L. GORELICK, M. BLANK, E. SHECHTMAN, M. IRANI, AND R. BASRI, *Actions as space-time shapes*, IEEE transactions on pattern analysis and machine intelligence, 29 (2007), pp. 2247–2253.

[34] T. GUHA AND R. K. WARD, *Learning sparse representations for human action recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 1576–1588.

[35] C. HARRIS AND M. STEPHENS, *A combined corner and edge detector.*, in Alvey vision conference, vol. 15, Citeseer, 1988, pp. 10–5244.

[36] M. HASAN AND A. K. ROY-CHOWDHURY, *Continuous learning of human activity models using deep nets*, in European Conference on Computer Vision, Springer, 2014, pp. 705–720.

[37] G. E. HINTON, S. OSINDERO, AND Y.-W. TEH, *A fast learning algorithm for deep belief nets*, Neural computation, 18 (2006), pp. 1527–1554.

[38] G. E. HINTON AND R. R. SALAKHUTDINOV, *Reducing the dimensionality of data with neural networks*, science, 313 (2006), pp. 504–507.

[39] B. K. HORN AND B. G. SCHUNCK, *Determining optical flow*, Artificial intelligence, 17 (1981), pp. 185–203.

[40] F. HU, L. LUO, F. ZHANG, AND J. LIU, *Action recognition using hybrid spatio-temporal bag-of-features*, in Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on, IEEE, 2010, pp. 812–815.

[41] N. IKIZLER-CINBIS AND S. SCLAROFF, *Object, scene and actions: Combining multiple features for human action recognition*, in European conference on computer vision, Springer, 2010, pp. 494–507.

[42] A. G. IVAKHNENKO, *Polynomial theory of complex systems*, IEEE transactions on Systems, Man, and Cybernetics, (1971), pp. 364–378.

[43] H. JHUANG, T. SERRE, L. WOLF, AND T. POGGIO, *A biologically inspired system for action recognition*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, Ieee, 2007, pp. 1–8.

[44] 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 (2013), pp. 221–231.

[45] Z. JIANG, Z. LIN, AND L. DAVIS, *Recognizing human actions by learning and matching shape-motion prototype trees*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 533–547.

[46] I. N. JUNEJO, K. N. JUNEJO, AND Z. AL AGHBARI, *Silhouette-based human action recognition using sax-shapes*, The Visual Computer, 30 (2014), pp. 259–269.

[47] P. KAEWTRAKULPONG AND R. BOWDEN, *An improved adaptive background mixture model for real-time tracking with shadow detection*, in Video-based surveillance systems, Springer, 2002, pp. 135–144.

[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, 2014, pp. 1725–1732.

[49] L. KITCHEN AND A. ROSENFELD, *Gray-level corner detection*, Pattern recognition letters, 1 (1982), pp. 95–102.

[50] A. KLASER, M. MARSZAŁEK, AND C. SCHMID, *A spatio-temporal descriptor based on 3d-gradients*, in BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association, 2008, pp. 275–1.

[51] O. KLIPER-GROSS, Y. GUROVICH, T. HASSNER, AND L. WOLF, *Motion interchange patterns for action recognition in unconstrained videos*, in European Conference on Computer Vision, Springer, 2012, pp. 256–269.

[52] J. J. KOENDERINK, *The structure of images*, Biological cybernetics, 50 (1984), pp. 363–370.

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

[54] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with deep convolutional neural networks*, in Advances in neural information processing systems, 2012, pp. 1097–1105.

[55] R. LAGANIERE, *A morphological operator for corner detection*, Pattern Recognition, 31 (1998), pp. 1643–1652.

[56] I. LAPTEV, *On space-time interest points*, International journal of computer vision, 64 (2005), pp. 107–123.

[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 Conference on, IEEE, 2008, pp. 1–8.

[58] Q. V. LE, W. Y. ZOU, S. Y. YEUNG, AND A. Y. NG, *Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3361–3368.

[59] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, nature, 521 (2015), p. 436.

[60] Y. LECUN, B. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. HUBBARD, AND L. D. JACKEL, *Backpropagation applied to handwritten zip code recognition*, Neural computation, 1 (1989), pp. 541–551.

[61] H. LEE, J. YOO, AND S. CHOI, *Semi-supervised nonnegative matrix factorization*, IEEE Signal Processing Letters, 17 (2010), pp. 4–7.

[62] H. LI AND M. GREENSPAN, *Multi-scale gesture recognition from time-varying contours*, in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, vol. 1, IEEE, 2005, pp. 236–243.

[63] T. LINDEBERG, *Scale-space theory: A basic tool for analyzing structures at different scales*, Journal of applied statistics, 21 (1994), pp. 225–270.

[64] J. LIU, J. LUO, AND M. SHAH, *Recognizing realistic actions from videos in the wild*, in Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on, IEEE, 2009, pp. 1996–2003.

[65] J. LIU AND M. SHAH, *Learning human actions via information maximization*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[66] J. LIU, J. YANG, Y. ZHANG, AND X. HE, *Action recognition by multiple features and hyper-sphere multi-class svm*, in Pattern Recognition (ICPR), 2010 20th International Conference on, IEEE, 2010, pp. 3744–3747.

[67] D. G. LOWE, *Object recognition from local scale-invariant features*, in Computer vision, 1999. The proceedings of the seventh IEEE international conference on, vol. 2, Ieee, 1999, pp. 1150–1157.

[68] B. D. LUCAS, T. KANADE, ET AL., *An iterative image registration technique with an application to stereo vision*, (1981).

[69] K. MIKOLAJCZYK AND C. SCHMID, *A performance evaluation of local descriptors*, IEEE transactions on pattern analysis and machine intelligence, 27 (2005), pp. 1615–1630.

[70] H. P. MORAVEC, *Obstacle avoidance and navigation in the real world by a seeing robot rover.*, tech. rep., STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.

[71] V. F. MOTA, J. I. SOUZA, A. D. A. ARAÚJO, AND M. B. VIEIRA, *Combining orientation tensors for human action recognition*, in Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on, IEEE, 2013, pp. 328–333.

[72] F. MURTAZA, M. H. YOUSAF, AND S. A. VELASTIN, *Multi-view human action recognition using 2d motion templates based on mhis and their hog description*, Iet Computer Vision, 10 (2016), pp. 758–767.

[73] J. C. NIEBLES, H. WANG, AND L. FEI-FEI, *Unsupervised learning of human action categories using spatial-temporal words*, International journal of computer vision, 79 (2008), pp. 299–318.

[74] L. PATINO, T. CANE, A. VALLEE, AND J. FERRYMAN, *Pets 2016: Dataset and challenge*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 1–8.

[75] S. PEHLIVAN AND D. A. FORSYTH, *Recognizing activities in multiple views with fusion of frame judgments*, Image and Vision Computing, 32 (2014), pp. 237–249.

[76] X. PENG, L. WANG, X. WANG, AND Y. QIAO, *Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice*, Computer Vision and Image Understanding, 150 (2016), pp. 109–125.

[77] X. PENG, C. ZOU, Y. QIAO, AND Q. PENG, *Action recognition with stacked fisher vectors*, in European Conference on Computer Vision, Springer, 2014, pp. 581–595.

[78] S. RAHMAN, S.-Y. CHO, AND M. LEUNG, *Recognising human actions by analysing negative spaces*, IET computer vision, 6 (2012), pp. 197–213.

[79] S. A. RAHMAN, I. SONG, M. K. LEUNG, I. LEE, AND K. LEE, *Fast action recognition using negative space features*, Expert Systems with Applications, 41 (2014), pp. 574–587.

[80] U. RAMER, *An iterative procedure for the polygonal approximation of plane curves*, Computer graphics and image processing, 1 (1972), pp. 244–256.

[81] M. RAVANBAKHSH, H. MOUSAVI, M. RASTEGARI, V. MURINO, AND L. S. DAVIS, *Action recognition with image based cnn features*, arXiv preprint arXiv:1512.03980, (2015).

[82] D. REISFELD, *The constrained phase congruency feature detector: simultaneous localization, classification and scale determination*, Pattern Recognition Letters, 17 (1996), pp. 1161–1169.

[83] D. RO AND H. PE, *Pattern classification and scene analysis*, (1973).

[84] L. ROSENTHALER, F. HEITGER, O. KÜBLER, AND R. VON DER HEYDT, *Detection of general edges and keypoints*, in European Conference on Computer Vision, Springer, 1992, pp. 78–86.

[85] S. SAMANTA AND B. CHANDA, *Fastip: a new method for detection and description of space-time interest points for human activity classification*, in Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, ACM, 2012, p. 8.

[86] M. SAPIENZA, F. CUZZOLIN, AND P. H. TORR, *Learning discriminative space–time action parts from weakly labelled videos*, International journal of computer vision, 110 (2014), pp. 30–47.

[87] C. SCHMID, R. MOHR, AND C. BAUCKHAGE, *Evaluation of interest point detectors*, International Journal of computer vision, 37 (2000), pp. 151–172.

[88] B. SCHÖLKOPF, A. SMOLA, AND K.-R. MÜLLER, *Kernel principal component analysis*, in International Conference on Artificial Neural Networks, Springer, 1997, pp. 583–588.

[89] C. SCHULDT, I. LAPTEV, AND B. CAPUTO, *Recognizing human actions: a local svm approach*, in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3, IEEE, 2004, pp. 32–36.

[90] P. SCOVANNER, S. ALI, AND M. SHAH, *A 3-dimensional sift descriptor and its application to action recognition*, in Proceedings of the 15th ACM international conference on Multimedia, ACM, 2007, pp. 357–360.

[91] Y. SHI, Y. TIAN, Y. WANG, AND T. HUANG, *Sequential deep trajectory descriptor for action recognition with three-stream cnn*, IEEE Transactions on Multimedia, 19 (2017), pp. 1510–1520.

[92] K. SIMONYAN AND A. ZISERMAN, *Very deep convolutional networks for large scale image recognition*, arXiv preprint arXiv:1409.1556, (2014).

[93] K. SIMONYAN AND A. ZISSERMAN, *Two-stream convolutional networks for action recognition in videos*, in Advances in neural information processing systems, 2014, pp. 568–576.

[94] S. M. SMITH AND J. M. BRADY, *Susan: a new approach to low level image processing*, International journal of computer vision, 23 (1997), pp. 45–78.

[95] P. SMOLENSKY, *Information processing in dynamical systems: Foundations of harmony theory*, tech. rep., COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.

[96] K. SOOMRO, A. R. ZAMIR, AND M. SHAH, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, arXiv preprint arXiv:1212.0402, (2012).

[97] L. SUN, K. JIA, T.-H. CHAN, Y. FANG, G. WANG, AND S. YAN, *Dl-sfa: deeply-learned slow feature analysis for action recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2625–2632.

[98] L. SUN, K. JIA, D.-Y. YEUNG, AND B. E. SHI, *Human action recognition using factorized spatio-temporal convolutional networks*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4597–4605.

[99] X. SUN, M. CHEN, AND A. HAUPTMANN, *Action recognition via local descriptors and holistic features*, in Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, IEEE, 2009, pp. 58–65.

[100] C. THURAU AND V. HLAVÁC, *Pose primitive based human action recognition in videos or still images*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[101] C. TOMASI AND T. KANADE, *Detection and tracking of point features*, (1991).

[102] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, AND M. PALURI, *Learning spatiotemporal features with 3d convolutional networks*, in Computer Vision (ICCV), 2015 IEEE International Conference on, IEEE, 2015, pp. 4489–4497.

[103] J. TURNER, *Maxwell on the method of physical analogy*, The British journal for the philosophy of science, 6 (1955), pp. 226–238.

[104] D. K. VISHWAKARMA AND R. KAPOOR, *Hybrid classifier based human activity recognition using the silhouette and cells*, Expert Systems with Applications, 42 (2015), pp. 6957–6965.

[105] D. K. VISHWAKARMA, R. KAPOOR, AND A. DHIMAN, *A proposed unified framework for the recognition of human activity by exploiting the characteristics of action dynamics*, Robotics and Autonomous Systems, 77 (2016), pp. 25–38.

[106] H. WANG, A. KLÄSER, C. SCHMID, AND C. LIU, *Dense trajectories and motion boundary descriptors for action recognition*, International journal of computer vision, 103 (2013), pp. 60–79.

[107] H. WANG, A. KLÄSER, C. SCHMID, AND C.-L. LIU, *Action recognition by dense trajectories*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3169–3176.

[108] H. WANG, M. M. ULLAH, A. KLASER, I. LAPTEV, AND C. SCHMID, *Evaluation of local spatio-temporal features for action recognition*, in BMVC 2009-British Machine Vision Conference, BMVA Press, 2009, pp. 124–1.

[109] H. WANG, C. YUAN, W. HU, AND C. SUN, *Supervised class-specific dictionary learning for sparse modeling in action recognition*, Pattern Recognition, 45 (2012), pp. 3902–3911.

[110] L. WANG, Y. QIAO, AND X. TANG, *Action recognition with trajectory-pooled deep-convolutional descriptors*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4305–4314.

[111] L. WANG AND D. SUTER, *Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model*, in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[112] G. WILLEMS, T. TUYTELAARS, AND L. VAN GOOL, *An efficient dense and scale-invariant spatio-temporal interest point detector*, in European conference on computer vision, Springer, 2008, pp. 650–663.

[113] L. WISKOTT AND T. J. SEJNOWSKI, *Slow feature analysis: Unsupervised learning of invariances*, Neural computation, 14 (2002), pp. 715–770.

[114] S.-F. WONG AND R. CIPOLLA, *Extracting spatiotemporal interest points using global information*, in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007, pp. 1–8.

[115] J. WRIGHT, A. Y. YANG, A. GANESH, S. S. SASTRY, AND Y. MA, *Robust face recognition via sparse representation*, IEEE transactions on pattern analysis and machine intelligence, 31 (2009), pp. 210–227.

[116] Y. YACOOB AND L. S. DAVIS, *Recognizing human facial expressions from long image sequences using optical flow*, IEEE Transactions on pattern analysis and machine intelligence, 18 (1996), pp. 636–642.

[117] G. K. YADAV AND A. SETHI, *A flow-based interest point detector for action recognition in videos*, in Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing, ACM, 2014, p. 41.

[118] G. K. YADAV, P. SHUKLA, AND A. SETHFI, *Action recognition using interest points capturing differential motion information*, in Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 1881–1885.

[119] W. YANG, Y. WANG, AND G. MORI, *Human action recognition from a single clip per action*, in Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 482–489.

[120] M. D. ZEILER AND R. FERGUS, *Visualizing and understanding convolutional networks*, in European conference on computer vision, Springer, 2014, pp. 818–833.

[121] M. D. ZEILER, G. W. TAYLOR, AND R. FERGUS, *Adaptive deconvolutional networks for mid and high level feature learning*, in Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 2018–2025.

[122] Z. ZHANG AND D. TAO, *Slow feature analysis for human action recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 436–450.

[123] J. ZHENG, Z. JIANG, AND R. CHELLAPPA, *Cross-view action recognition via transferable dictionary learning*, IEEE Transactions on Image Processing, 25 (2016), pp. 2542–2556.

[124] J. ZHENG, Z. JIANG, P. J. PHILLIPS, AND R. CHELLAPPA, *Cross-view action recognition via a transferable dictionary pair.*, in bmvc, vol. 1, 2012, p. 7.

[125] F. ZHU AND L. SHAO, *Correspondence-free dictionary learning for cross-view action recognition*, in Pattern Recognition (ICPR), 2014 22nd International Conference on, IEEE, 2014, pp. 4525–4530.

[126] J. ZHU, B. WANG, X. YANG, W. ZHANG, AND Z. TU, *Action recognition with actons*, in Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3559–3566.