Title	監視カメラ映像を対象とした特徴抽出と人物間動作の認識
Author(s)	NGUYEN, NGOC THUY
Citation	
Issue Date	2016-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/13828
Rights	
Description	Supervisor:吉高　淳夫，情報科学研究科，博士

Doctoral Dissertation

Feature Extraction and Human-Human Interaction Recognition for Video Surveillance

NGUYEN, Thuy Ngoc

Supervisor: **Associate Professor Atsuo Yoshitaka**

School of Information Science
Japan Advanced Institute of Science and Technology

June, 2016

Abstract

Human interaction recognition has been widely studied because it has great scientific importance and many potential practical applications. However, this problem is very challenging especially in realistic environments where background is dynamic and has varying lighting conditions. This dissertation addresses human activity recognition, especially human-human interactions in realistic video material, such as movies, surveillance videos. For classification problem, most existing methods rely on either spatio-temporal local features (i.e. SIFT) or human poses, or human joints to model human interactions. As a result, they are not fully unsupervised processes because they require either hand-designed features or human detection results. Motivated by the recent success of deep learning networks, we investigate a three-layer convolutional network which uses the Independent Subspace Analysis (ISA) algorithm to learn hierarchical invariant features. The ISA algorithm is a generalization of the Independent Component Analysis (ICA), which is very well-known in natural image statistics. Compared to the ICA algorithm, the most notable advantage of the ISA is that it can learn features which are invariant to phase while being selective to orientation and frequency. However, the ISA algorithm becomes slow when applying it on video data. In order to solve this computational problem, we combine the idea of convolutional neural network with the ISA algorithm. Specifically, instead of training the ISA algorithm directly on raw video data, we first train it on small video blocks extracted by our procedure. The obtained features are then convolved with larger video blocks. The outputs of this convolution step are fed into the next layer, which is implemented by another ISA algorithm. This organization enables the three-layer convolutional ISA network to learn hierarchical invariant features. Furthermore, we introduce a pooling layer to reduce the contributions of features learned in lower layers while still achieving translation invariant. Using the invariant features learned by the three-layer convolutional ISA network, we build a bag-of-features representation for videos. Finally, we apply Support Vector Machine (SVM) to classify human interactions. For temporal localization, we slide temporal detection windows with different durations over a continuous video sequence with a stride of 10 frames. For each temporal window, our convolutional ISA network extracts hierarchical invariant features on a dense grid. After scoring the temporal detection windows, a non-maximum suppression is applied to enforce that non of the retained windows are overlapping.

In two cases, we conduced thorough experiments on realistic videos from challenging benchmarks used by activity recognition community. We show that our three-layer convolutional ISA network is effective to represent complex activities such as human interactions in realistic environments. Besides, we believe that our temporal localization method is the first work which reports experimental results on the continuous video sequences of human interactions. Although temporal localization results are insufficient for real applications, it is a first step for further research in localization of human interactions.

Keywords: Temporal localization, Classification, Independent subspace analysis, Human-human interactions, Convolutional neural network, Pooling.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Videos help to improve quality of contents and communication by combining visual, audio, and textual information in multiple data streams. Over the past years, with the rapid development of high technology and faster internet access, video data has become superfluous both in off-line storage and on the internet. Specifically, electronic devices such as computers, tablets, mobile phones are used almost everywhere, and people can record, store, and share videos easily. According to statistics in 2016 [20] from the most popular sharing website - YouTube:

- 5 billion videos are watched on the YouTube every single day.
- 300 hours of video are uploaded to it every minute.
- Total number of people who use YouTube is 1.3 billion.

This shows that videos as well as video cameras have become an inseparable part of our lives. With a fast growing number of videos and with such widespread popularity of watching videos and movies on the Internet, there is an urgent need for advanced video analysis techniques that can systematically interpret and understand the semantics of video contents.

Automatic understanding the content of a video is a long-standing goal of computer vision and it is interesting to identify which objects are the most important in videos. Ivan Laptev [51] performed an experiment and found out that about 35% of screen pixels in movies, TV programs and YouTube videos belong to people. Besides, images on Flickr.com contain about 25% of person pixels. These numbers imply that the visual data we tend to produce, share and consume is strongly biased toward people. However, the percentage of person pixels in a first-view wearable camera dataset [39] is only about 4%. This further indicates that the strong person bias in consumer videos is not natural and is created with the intention of video maker or editor. With the strong bias of video toward people, understanding the semantics of video contents requires the need of automatic methods that interpret person pixels in videos.

There are several levels of interpreting person pixels which can lead to understanding of video contents including people detection, people tracking and analyzing their activities. People detection is the process of verifying the presence of a person (or people) in image sequences and identifying their positions precisely. People tracking is to determine the location of a moving person (or people) during a video sequence. Generally, people detection and tracking are closely related because people tracking usually starts with detecting people. Although people detection and tracking have a wide range of applications, these two processes can only answer simple questions like who are in a video and their trajectories. However, the more interesting information is people behaviors, i.e. what are they doing in a video. Thus, the natural step is to recognize human activities occurred in videos.

Human activity is defined as a collection of human movements with a particular semantic meaning. Recognizing human activities in video is of high interest because of many applications such as entertainment, education, security and surveillance. This research aims at recognizing human activities in movie videos and surveillance videos, hence we then analyze two main applications which receive benefits from our research.

Content-based analysis. As mentioned previously, there are a huge amount of video data uploaded on the Internet every second, and there is a widespread popularity of watching movie videos on the Internet. Users may want retrieve clips with activities of interest, e.g. kiss, flash mob from lot of videos, and it is called content-based search. The traditional methods for content-based search have relied on text, such as those extracted from closed captions or manual annotation. Annotating the unlabeled videos is labor-intensive, and since videos grow explosively, it is impractical to do manual annotation on all activities occurred in a movie video. Therefore, it has become necessary to design an efficient-content based search algorithm and the ability to analyze activities plays an important role for this application. The Hollywood2 dataset [62] is built to test the capacity of activity recognition for content-based search application in movie videos. It consists of 12 activities such as kissing, answering phone, driving a car and so on, which are collected from 69 different Hollywood movies. Besides, it exhibits several challenging factors such as scene and viewpoint variations.

Video surveillance. Nowadays, video surveillance cameras are widely accepted by society and used almost everywhere such as at airport, subways, train stations, bus terminals, shopping malls, banks, post offices and parking lots. Hence, video surveillance cameras are part of our lives. According to BBC statistics in 2009, there are about 1 million surveillance cameras which are installed in the United Kingdom. Hence, constructing automated surveillance systems is one of urgent issues, and such systems require the capacity of detecting abnormal and suspicious activities. As a result, the demand for activity recognition systems increases as well.

People are much better than computers at recognizing human activities. Hence, we believe that mimicking the behavior of human brain can improve the performance of activity recognition systems. Most of perceptual processes are carried out by the neocortex, and the primary visual cortex is the part of the neocortex that receives visual input from the retina. Independent Subspace Analysis (ISA) algorithm has long been studied by researchers in the field of natural image statistics. This algorithm produces outputs very similar to those of complex cells in the primary visual cortex [36]. Besides, deep learning is also inspired by how the the human brain works, and has produced extremely promising results in computer vision and natural language processing. Therefore, we expect that our activity recognition method based on independent subspace analysis and deep learning techniques can obtain better performance compared to other methods.

1.2 Problem statement

There are various types of human activities. Aggarwal et al. [1] conceptually categorize human activities into four levels according to their complexity:

- Gestures are defined as elementary movements of a person's body part, and are atomic components which actions are built. Waving a hand and raising a leg are good examples of gestures.

- Actions are activities that are performed by a single person, and are composed of multiple gestures organized temporally. Figure 1.1 shows sample actions such as walking, jogging, which are extracted from the well-known KTH dataset [84].

- Interactions are classified into two sub-categories: human-human interactions, and human-object interactions. The term human-human interaction is used to describe

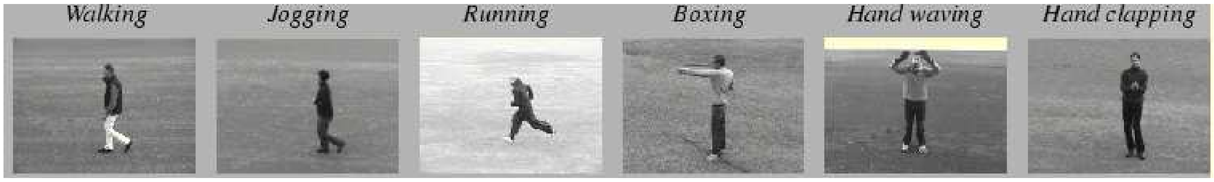


Figure 1.1: Sample actions in the KTH dataset [84]

activities that involve at-least two persons, e.g. two-person fighting. Besides human-human interactions, recognizing human-object interactions is also important, especially in airport or train stations. A person stealing a suitcase from another is an example of human-object interaction.

- Group activities are characterized by movements of members in groups, for instance, a group of persons crossing, a group of persons waiting, and a group of persons queuing.

In addition to activity's categories, the type of video data also affects the complexity of activity recognition. We broadly classify video data into two categories:

- Unrealistic video data is recorded in simplified settings, e.g. a single person fully visible or favorable lighting conditions. KTH [84] and Weizmann [6] are two typical examples, and have been extensively used to report action recognition performance by many researchers [19, 40, 41, 43, 48, 68, 76]. These datasets are captured with a fixed camera in controlled conditions in which only a single person appears in videos taken from a fixed point and with a homogeneous background. They enables to explore the classification ability of systems with variations in actors and actions.

- Realistic video data is characterized by a great variability and the lack of available prior knowledge applicable to (i) the scene (e.g. indoors or outdoors, lighting conditions), (ii) the record setting (e.g. the viewpoint, fixed camera or not, video quality). Potential sources of realistic video data include amateur videos, sports broadcasts, movies, surveillance videos. The most popular datasets are UCF Sports [78], Hollywood2 [62], UT-Interaction [81]. Figure 1.2 shows sample frames of realistic video data, which are extracted from the UCF Sports dataset. Although these datasets are new, they have attracted a lot of attention (see, for example, [18, 61, 82, 87]).

This dissertation focuses on the problem of *human activity recognition*, especially *human-human interactions* in *realistic video material*, such as movies, surveillance videos. The goal of human activity recognition is to analyze activities from an unknown video

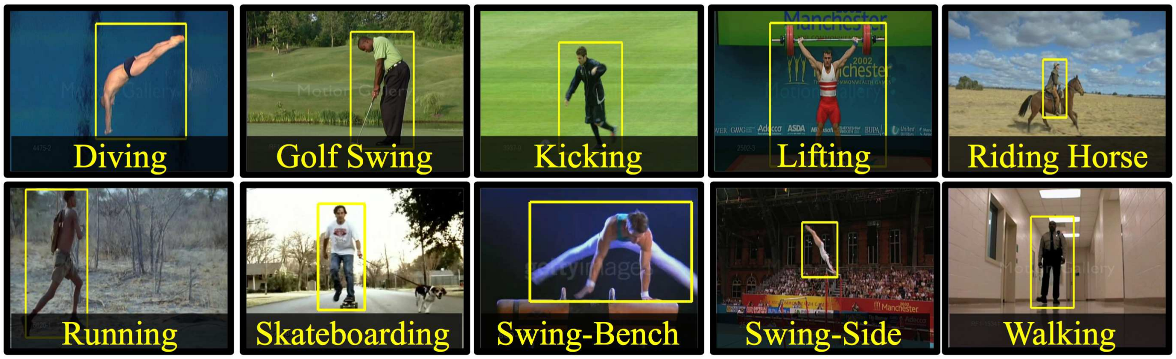


Figure 1.2: Sample frames of realistic video data - UCF Sports [78]

automatically. This dissertation solves two main tasks: *classification* and *localization*. The objective of activity localization is to identify not only which type of activities occurs but also where it occurs in a video. The capacity for activity localization is especially essential in video surveillance systems. Since activity localization is challenging, most of recent methods have only concentrated on activity classification, which is a simplified version of activity localization. Activity classification task is based on the assumption that videos are segmented temporally and/or spatially to contain only one execution of human activity, then its objective is to label each video with its activity category correctly.

There are two types of activity localization: temporal localization and spatio-temporal localization. Temporal localization detects starting and ending frames of the activity. Spatio-temporal localization is more complex because it requires to identify starting, ending frames, and spatial bounding box of the activity. In practice, classification and localization tasks are not separable because most of localization methods often apply a classifier at multiple candidate locations to identify the bounding box that encloses the region of interest.

In summary, in this dissertation, we address the issue of automatic *classification* and *temporal localization* of human-human interactions. Our methods are evaluated on *realistic video data*, which is from various sources of videos, e.g. surveillance videos (the UT-Interaction dataset [81]), movies (the Hollywood2 dataset [62]) for different types of applications, e.g. surveillance, video indexing.

1.3 Challenges with human activity recognition

Human activity recognition is an important and challenging research topic. One of the main challenges is that the same activity can be performed in many different ways, even by the same person. In the following sections, we will analyze research challenges in more detail.

1.3.1 Intra-class and inter-class variations

An issue of human activity recognition is variations of activities. We classify variations of activities into two types: intra-class variations, and inter-class variations.

- Intra-class variations: Variations in the same activity class are referred to as intra-class variations which are the consequences of differences in anthropometry, appearance of actors, and execution rate. Execution rate refers to the speed of performing an activity. For example, there are several versions of a waving a hand gesture. A person might move his hand above his head and then wave his hand; but another person might not move his hand above his head and would just wave from a shoulder height. In addition, people perform a waving hand gesture at different speed and/or duration.

- Inter-class variations: Other variations of activities are inter-class variations. For example, since punch and push interactions share similar movements, differentiating between the two interaction classes is based on the fact that people use two hands to perform push interactions and only one hand for punch interaction. This fact can be clearly recognized by human; however it is difficult to be recognized by computers. In summary, such above variations have to be taken into account in an activity recognition system.

1.3.2 Environmental parameters

The environment in which activities take place is an important factor to consider when researchers record datasets. There are several environmental parameters which affect recognition results: lighting conditions, cluttered backgrounds, occlusions, camera motion, and variations in viewpoint.

- Lighting conditions: The appearances of people in videos change significantly when lighting conditions vary from indoor environments to outdoor ones. In addition, even in outdoor environments, the moment that activities are recorded also influences video quality. Hence, lighting conditions are currently controlled for most of the datasets.

- Cluttered backgrounds: Cluttered backgrounds refer to the presence of other objects or people in the video frame. Hence, cluttered backgrounds make activity recognition even harder as they produce ambiguous information.
- Occlusions: In surveillance videos, there are a lot of people in the view which often cause occlusions. Occlusions can be classified into two types: self-occlusions and occlusions which created by other objects in the scene. These issues should be addressed explicitly in activity recognition systems.
- Camera motion: Most recognition methods require that activities are captured by a fixed camera. One of the reasons is that motion features are affected severely by moving cameras. Motion features characterize activities, and they are one of the most reliable features for activity recognition. Therefore, in unconstrained environments where camera motion exists, recognizing activities typically requires some techniques to remove camera motion components, or features that are invariant to camera motion.
- Variations in viewpoint: Another challenge in activity recognition is variations in viewpoint. The same activity which is captured from different viewpoints can produce different image observations and different motion patterns. Therefore, most methods simplify recognition problem by assuming that activities are recorded from a fixed viewpoint. However, multiple camera viewpoints would be beneficial to alleviate the issues of occlusion.

1.4 Contributions

The goal of this dissertation is to recognize human interactions in realistic video data. The first part of our work is based on local features, which are employed for interaction classification. For this, we investigate existing methods based on local features, and then we develop our new method. The second part of this work introduces our approach for interaction localization in videos. Experimental evaluation is performed on various datasets including the UT-Interaction dataset [81] and the Hollywood2 dataset [62].

To summarize, we provide the following main contributions:

- Previous approaches in activity recognition rely on human poses, human joints, and 3-dimensional local spatio-temporal features. However, it is difficult and time consuming to apply human poses and human joints to different datasets. In detail, human poses and human joints are easily extracted from the UT-Interaction dataset because people in the video scenes are visible, i.e., the average height of person is about 200 pixels. However, in the Hollywood2 dataset, people appear in the video scenes with different scales and different poses (full-body person or the upper-body person); hence it is impossible to extract these features. 3-dimensional local spatio-temporal features usually have two stages: a feature detection stage followed by a feature description stage. Well-known feature detector methods are Harris3D [52], Cuboids [19], and Hessian [107]. Popular feature description methods are Cuboids, HOG/HOF [53], HOG3D [44], and Extended SURF [107]. Wang et al. [32] combined various feature detector methods and feature descriptor methods and evaluated these combination on KTH and Hollywood2 dataset. One of their interesting findings is that there is no universally best 3-dimensional local spatio-temporal feature method for all datasets (KTH and Hollywood2). This finding suggests that learning features directly from the raw data may be more advantageous. This dissertation focuses on developing an unsupervised feature learning method which learns features directly from realistic video data and achieves good balanced results on the UT-Interaction and Hollywood2 dataset. Motivated by the Independent Subspace Analysis (ISA) [34] and deep learning, we introduce a three-layer convolutional ISA network to learn hierarchical invariant features. The ISA is an interesting generalization of Independent Component Analysis (ICA), which is very well-known in natural image statistics. An advantage of ISA, compared to ICA is that it can learn features that are robust to local translation while being selective to frequency, rotation and velocity. However, the ISA algorithm becomes slow if we apply it on video data. Besides, deep learning methods have shown promising results in computer vision and natural language processing. Therefore, in this dissertation, we combine the idea of convolutional neural network with the ISA algorithm to improve the computational time and classification performance on the UT-Interaction and Hollywood2 dataset. Particularly, instead of training the ISA algorithm directly on raw video data, we sample small video blocks from the raw video and train on these blocks. The procedure of extracting video blocks is proposed. We also introduce an organization of three-layer convolutional ISA network, which is able to capture hierarchical representation for videos. Besides, the pooling layer is also presented to concatenate all responses from three layers to construct local features. Experimental results show that no single method achieve higher results on the UT-Interaction and Hollywood2 dataset compared to our method.

- To the best of my knowledge, no attempt has been made for temporal localization of human interactions. Therefore, we develop an approach to localize human interactions temporally in the UT-Interaction dataset [81]. Our localization method is based on the sliding window technique, which slides a window over the entire video and selects the temporal detection window with maximum classifier score. For video representation, we use hierarchical invariant features which are extracted from our three-layer convolutional ISA network. Our localization method is evaluated on the continuous video sequences of the UT-Interaction dataset.

1.5 Dissertation organization

Structure of this dissertation is as follows:

- **Chapter 1: Introduction**

  In this chapter, we analyze two important factors including enormous video data, and applications which explain the importance of human activity analysis. Besides, we also present and categorize challenges of activity recognition.

- **Chapter 2: Related work**

  This chapter describes a literature review on related studies with discussion on their advantages and drawbacks.

- **Chapter 3: Datasets**

  The existing datasets are presented to give a historical overview of development of datasets on activity recognition problem. We also present and analyze the characteristics of the UT-Interaction and Hollywood2 dataset which are used to evaluate our proposed methods.

- **Chapter 4: Interaction Recognition using Hierarchical Invariant Features**

  This chapter is devoted to describe our three-layer convolutional ISA network which is designed to learn hierarchical invariant features. First, we introduce our procedure to extract video blocks, which are the inputs for the three-layer convolutional ISA network. The organization of our convolutional ISA network is also presented in this chapter. Besides, we also present our pooling layer to concatenate and reduce the contributions of simple features. Finally, we evaluate our classification method on datasets, which are described in Chapter 3.

- **Chapter 5: Interaction Temporal Localization based on Sliding Window Approach**

  In this chapter, we present our localization method based on the sliding window technique. We describe temporal sliding window, features for representation videos, classification, and post-processing method in detail. Then, our temporal localization is evaluated on the continuous video sequences of the UT-Interaction dataset.

- **Chapter 6: Conclusion and Perspective**

  Finally, a summary of presented methods will be shown in Chapter 6. Besides, we also discuss possible improvement and extension for human interaction recognition.

Chapter 2

Related Work

There is a large amount of papers published each year in the literature on activity recognition. To provide the context of our work in the domain of activity recognition, we begin by reviewing the existing papers on activity recognition for video data. We provide an overview, describe the most relevant state-of-the-art techniques, and also discuss their advantages and disadvantages. In this chapter, we present recent work, and general survey papers [1, 12, 29, 31, 66, 73, 101, 106] are suggested for further reading to get broader historical development overview of activity recognition.

2.1 Local feature methods

Local features capture appearance and motion information of small video volumes, and are briefly classified into two types: spatio-temporal features and trajectory features. They aim to provide independent representation of activity videos with regard to their spatio-temporal shifts and scales. Activity recognition based on local features is one of the most active research topics. There are several reasons for their popularity. First, local features make no assumptions on the global structure of activities. Second, they are extracted directly from the video, therefore it is able to avoid possible failures of pre-processing steps such as motion segmentation, human detection and tracking. Third, local features are also robust under uncontrolled settings, e.g. background clutter. The effectiveness of the local features have been evaluated on various sources of video data.

2.1.1 Spatio-temporal features

Local spatio-temporal features aim to capture small three dimensional spatio-temporal volumes of the video, and are usually extracted at precise locations and scales. One of the first work on local spatio-temporal feature detectors is of Laptev et al. [52]. They proposed the Harris3D interest point detector, which is an extension of the Harris interest point detector [33] in image domain. Interest points are those points with a significant local variation of image intensities, and such points are attractive due to their high information contents. Similarly, in video domain, the Harris3D interest points are the locations where image values have significant local variations in both space and time domain. It is assumed that the interest points often correspond to interest events in the video, thus they are informative to construct video representation.

To detect Harris3D interest points, Laptev et al. [52] compute a spatio-temporal second-moment matrix at each video point, and redefine the Harris corner function H in the spatio-temporal domain. Positive local maxima of H correspond to points with high variation of the image values in both space and time. Thus, Harris3D interest points can be found by detecting local maxima of H. Figure 2.1 shows Harris3D interest points in an outdoor image sequence of a person walking.

Dollar et al. [19] observed that in some cases, Harris3D corners are quite rare even when an interesting motion occurs. Therefore, they proposed an alternative interest point detector which yields denser results than the Harris3D. In detail, they employ a set of spatial Gaussian kernels and temporal Gabor filters. Similar to the work of Laptev et al. [52], the final spatio-temporal interest points are detected by finding local maxima of the defined response function.

Willems et al. [107] introduced the Hessian3D detector, which is a spatio-temporal extension of the Hessian saliency measure applied for blob detection in images [56]. The Hessian3D detector computes the Hessian matrix, and measures saliency using the determinant of the Hessian matrix. Besides, an integral video structure is used to speed up the detector by approximating derivatives with box-filter derivations. A non-maximum suppression algorithm is applied to select joint extrema over space, time, and different scales. Generally, the detected Hessian3D interest points are denser than those extracted from the Harris3D detector, but not as dense as those from the work of Dollar et al. [19].

Previous techniques detect spatio-temporal interest points by using local information (local neighborhood). Wong et al. [109] proposed an interest point detector by considering global information. The global information, i.e. the organization of pixels in a whole video sequence, is obtained by applying non-negative matrix factorization. The detector

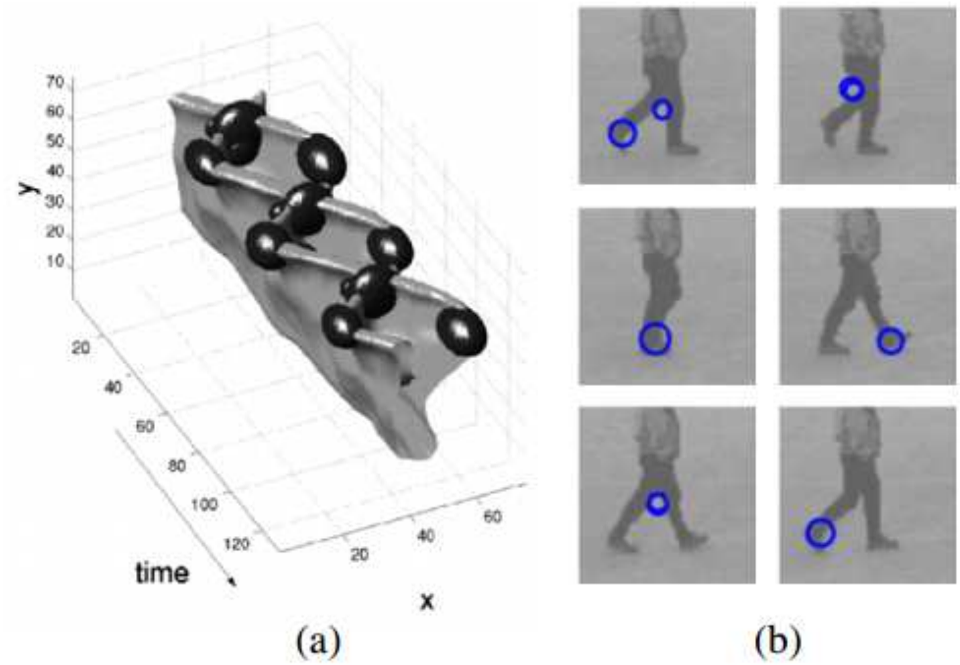


Figure 2.1: Harris3D interest points for the motion of the legs of a walking person. Left image is a 3D plot with a threshold level surface of a leg pattern (upside down) and the detected points. Right image shows the detected interest points [52].

extracts the location of moving parts in a video, and searches for the regions which have a large probability of containing the relevant motion.

Wang et al. [32] have evaluated previous detectors, and the comparison was done on three datasets: KTH [84], UCF [57], and Hollywood2 dataset [62]. According to their evaluation, there is no single detector which achieves the best results. However, compared to other detectors, the Harris3D detector usually obtains good results.

2.1.2 Trajectory features

Trajectory features are usually extracted by detecting spatio-temporal interest points and tracking them in time. Compared to spatio-temporal features, trajectory features encode information about local motion patterns of neighborhood of detected interest points. Several researchers [63,65,92] proposed several ways to detect trajectory features.

In literature, one of the best-known feature tracking methods is the KLT tracker [24,59]. Matikainen et al. [63] extract trajectories of fixed length using the KLT tracker, and then cluster the trajectories. An affine transformation matrix is calculated for each cluster, and the elements of the matrix are then used to represent the trajectories. Figure

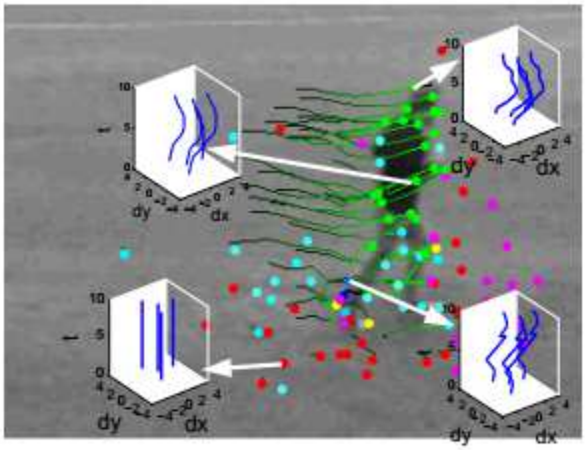


Figure 2.2: Feature trajectories are obtained by detecting and tracking interest points. Trajectories are clustered into a library of trajectons [63].

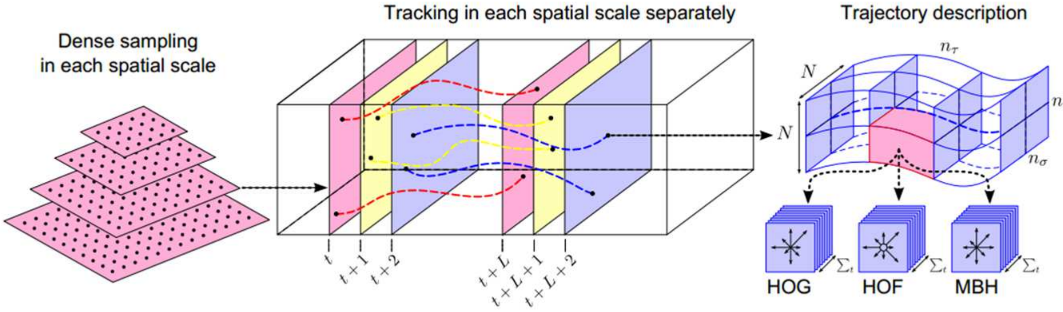


Figure 2.3: Overview of the dense trajectories [103].

2.2 illustrates the extracted feature trajectories. Messing et al. [65] extract interest points by the Harris3D detector and track these points by the KLT tracker. Then, the trajectories with varied lengths are represented as sequences of log-polar quantized velocities.

Different from the above methods, Sun et al. [92] proposed to detect trajectories by matching SIFT descriptors over consecutive frames. The SIFT descriptor is used because it's scale-invariant properties are better compared to the Harris and KLT based feature trackers. Wang et al. [103] also proposed a method to extract dense trajectories. They apply dense sampling to extract interest points, and track them by using a dense optical flow field (illustrated in Figure 2.3). Then, the trajectory shape, histogram of oriented gradients (HOG), histogram of optical flow (HOF), and motion boundary histogram (MBH) are used to capture appearance and motion information of trajectories. This method showed that it outperforms other trajectory-based methods.

2.1.3 Feature descriptors

Feature descriptors characterize shape and motion information in a local neighborhood surrounding interest points and trajectories. Dollar et al. [19] introduced several local feature descriptors based on brightness, gradient, and optical flow information. They investigate three methods to create a feature vector: a simple concatenation of pixel values by flattening, a grid of local histograms, and a single global histogram. Then, principal component analysis is applied to reduce the dimension of each descriptor. Finally, their experimental results show that gradient information yields best performance.

Laptev et al. [53] capture local motion and appearance by combining histograms of oriented gradients (HOG) and histogram of optical flow (HOF). In detail, the local neighborhood surrounding of each detected interest point is divided into a $N \times N \times M$ grid of cells. Then, for each cell, they compute 4-bin HOG histogram and 5-bin HOF histogram. These cell histograms are normalized and concatenated into a final descriptor. Scovanner et al. [85] introduced an extension of the SIFT (scale invariant feature transform) descriptor [58] from the image domain to the video domain. This descriptor is developed based on the spatio-temporal grid idea and spatio-temporal gradients. Each pixel is weighted by a Gaussian centered on the given position, and votes into a grid of histograms of oriented gradients. The Gaussian weighting is applied to assign less weights (importance) of gradients which are far away from the center of local feature. Besides, the dominant operation is used to achieve rotation-invariance.

An extension of the HOG image descriptor [16] to 3D (called HOG3D) was proposed by Klaser et al. [44]. The proposed descriptor is based on the spatio-temporal grid idea and histograms of 3D gradient orientations. Gradients are computed based on convex regular polyhedrons, and by using an integral video representation to speed up computation. Similarly, Willems et al. [107] also proposed the extended SURF (ESURF) descriptor, which is an extension of the image SURF (speeded up robust features) descriptor [5]. In detail, the ESURF method divides the local neighborhood surrounding a local feature into a spatio-temporal grid, and each cell is represented by a vector of weighted sums of uniformly sampled responses of Haar-wavelets along three x, y, t axes.

2.1.4 Encoding methods

The goal of encoding step is to aggregate local features into a global vector representation. Representing a video as a fixed-size vector will leverage standard classification algorithms, such as logistic regression and support vector machine. Besides, the encoding step represents videos with global vectors which are usually smaller than extracted local features. Encoding methods usually consist of three main stages:

1. Find the most representative cluster centers in the feature space.

2. Assign extracted local features to the selected cluster centers.

3. Model the statistics of the assigned features.

One of the most well-known encoding methods is the bag-of-features (BOF) model, which was originally proposed for document retrieval in natural language processing. Then, it has become popular in computer vision, for example [14, 15, 54, 70, 88, 89]. The BOF model encodes global statistics of local features by computing a histogram of occurrences of local features in a video sequence. Firstly, clusters are created by using unsupervised learning method over local features extracted from training videos. The learning is typically done with k-mean clustering algorithm. Note that each cluster center represents a feature or a visual word. Secondly, local features are quantized by assigning to their closest visual words, which is called hard quantization. Finally, a video is represented as a frequency histogram over the visual words (vector of counts), whose size is equal to the number of centroids. L1 and L2 norm are two popular metrics in the BOF encoding, and there are no clear answer which norm is the best.

The BOF encoding uses hard quantization of local features, i.e. histogram encoding. Recent approaches replace the hard quantization with soft-assignment encoding techniques, such as Kernel codebook encoding [99], Fisher vector encoding [71, 83], and Bossa encoding [4]. The Fisher vector encoding models feature space by taking richer statistics into account: the mean, the variance of the assigned features in addition to the sum of posterior probabilities. Instead of using k-mean clustering, the Fisher vector encoding uses Gaussian Mixture Model to construct visual words.

2.2 Global representation methods

Global representation methods recognize activities by employing appearance and motion information either of the whole body structure or of a region of interest which encloses a subject tightly. Global representations are typically derived from silhouette extraction. Global representation methods are widely used in activity recognition because they do not rely on detection and tracking of individual body parts. This property is important especially for realistic videos in which background clutter and occlusion result in identification of body parts particularly difficult. In general, global representation approaches

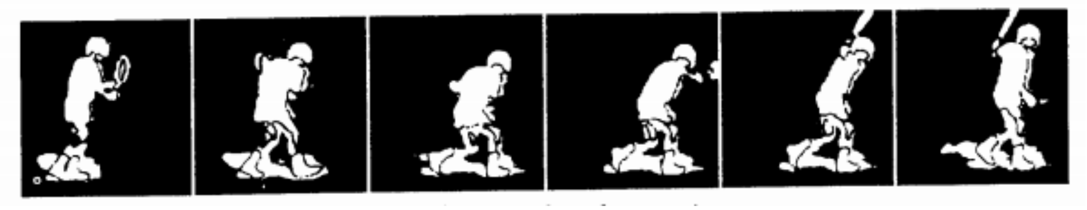


Figure 2.4: The sample shape masks for the backhand stroke activity from the tennis activity [110].

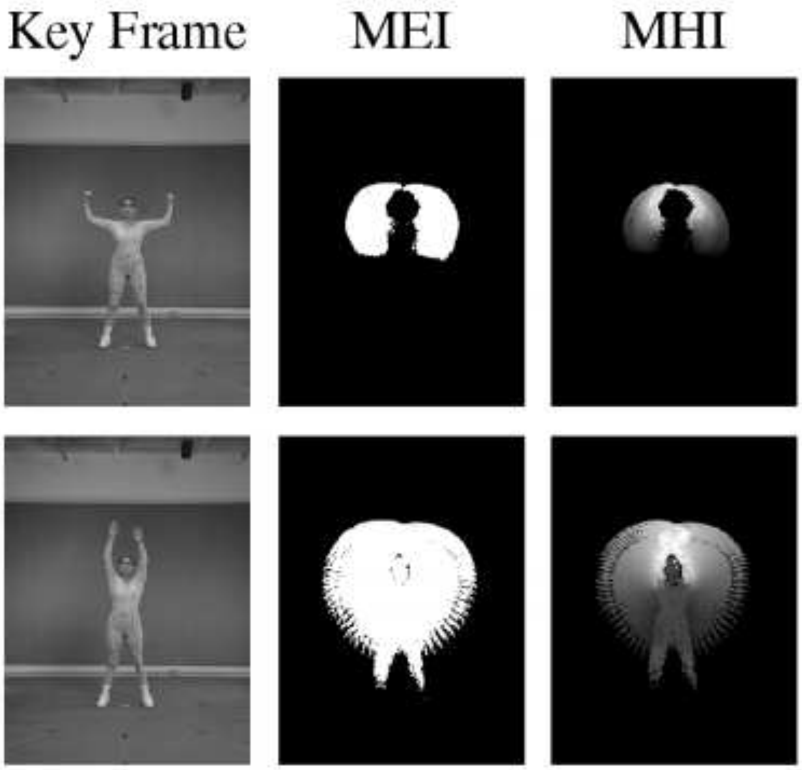


Figure 2.5: MEI and MHI representation for two sample movements. [8].

can be roughly divided into two categories: shape mask and silhouette based methods, optical flow and shape based methods, and body part based methods.

2.2.1 Shape mask and silhouette based methods

Several approaches for activity recognition represent the human body and its dynamics by using shape masks and silhouette information. One of the first methods using silhouette is by Yamato et al. [110] (see Figure 2.4). Firstly, they extract a human shape mask for each frame, compute a grid over the silhouette, and also calculate the ratio of foreground

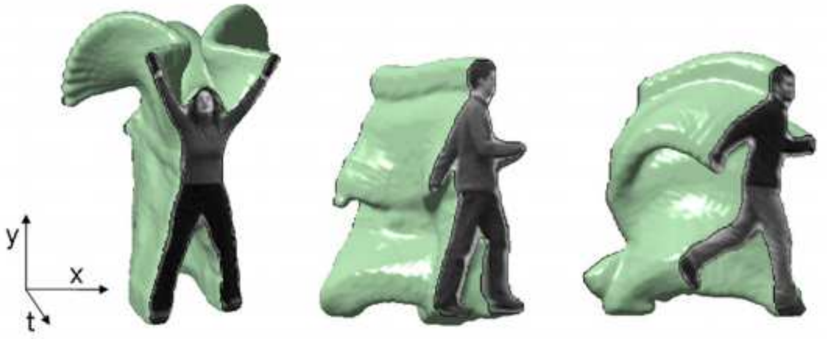


Figure 2.6: Space-time shapes of ‘jumping-jack’, ‘walking’, ‘running’ [6].

to background pixels for each cell. Secondly, the grid representations are quantized into a vocabulary, and the Hidden Markov Models (HMMs) [74] is applied to learn human activities.

Bobick et al. [8] introduced the idea of temporal templates for activity recognition. The silhouettes are extracted from images and the differences of the silhouettes between subsequent frames of the video are aggregated to construct binary motion-energy images (MEI) and motion-history images (MHI) (see Figure 2.5). The MEI images are binary masks which indicate regions of motion. One the other hand, the MHI images weight the motion regions as a function over time (the more recent the higher the function is). Each activity is represented with a temporal template which is composed of the MEI and MHI image. Then, they develop a recognition method by matching temporal templates against stored instances of activities.

Blank et al. [6] proposed an method which represents activities as three-dimensional shapes. In detail, a silhouette is extracted for each frame using background subtraction, and space-time shapes are constructed by stacking a sequence of silhouette images (see Figure 2.6). Then, the properties of the solution to the Poisson equation are explored to extract features such as local saliency, action dynamics, shape structure and orientation. The weighted moments over these features are calculated and are used to represent each sequence of an activity. Finally, a simple nearest neighbor classification with Euclidean distance is applied to recognize activities.

Weinland et al. [105] introduced a compact representation for activity recognition using a set of discriminative silhouette exemplars without modeling any temporal ordering. Activity sequences are then represented as vectors of minimum distance between silhouettes in the set of exemplars and in the sequence. Finally, Bayes classifier with Gaussians is applied to recognize activities.

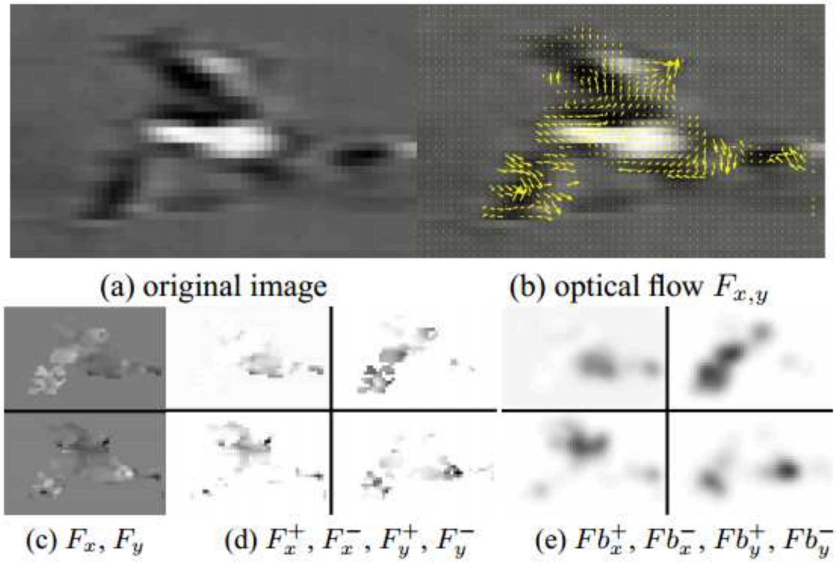

(a) original image
(b) optical flow $F_{x,y}$
(c) F_x, F_y
(d) $F_x^+, F_x^-, F_y^+, F_y^-$
(e) $Fb_x^+, Fb_x^-, Fb_y^+, Fb_y^-$

Figure 2.7: Motion descriptor using optical flow (a) Original image (b) Optical flow $F_{x,y}$(c) Separate the x and y components of optical flow vectors F_x, F_y (d) Half-wave rectification of each component to produce 4 separate channels F_x^+, F_x^-, F_y^+, and F_y^-, (e) Final blurry motion channels Fb_x^+, Fb_x^-, Fb_y^+, and Fb_y^- [23].

Generally, silhouette information is very useful for activity recognition. However, silhouettes are difficult to be extracted when background clutter and camera motion are present. Furthermore, they only describe the outer contours of a person, therefore, silhouette-based methods may not recognize activities which contain self-occlusions.

2.2.2 Optical flow and shape based methods

Another type of global representation methods is to use dense optical flow information for activity recognition. Efros et al. [23] introduced a novel motion descriptor based on optical flow measurements in a space-time volume for each person. In detail, they track soccer players in videos, and compute a descriptor on the tracks using blurred optical flow. Figure 2.7 shows motion descriptor using blurred optical flow. To classify the activity being performed by a human figure in a query sequence, they retrieve nearest neighbors from an annotated video sequences. Ahad et al. [2] use these four flow channels to compute motion templates. This method has proven that it can solve the motion overwriting of self-occlusion in a MHI approach [8].

Several approaches [17,96] build a grid-based representation of optical flows for activity recognition. For example, Danafar et al. [17] adapt the work of Efros et al. [23] by dividing

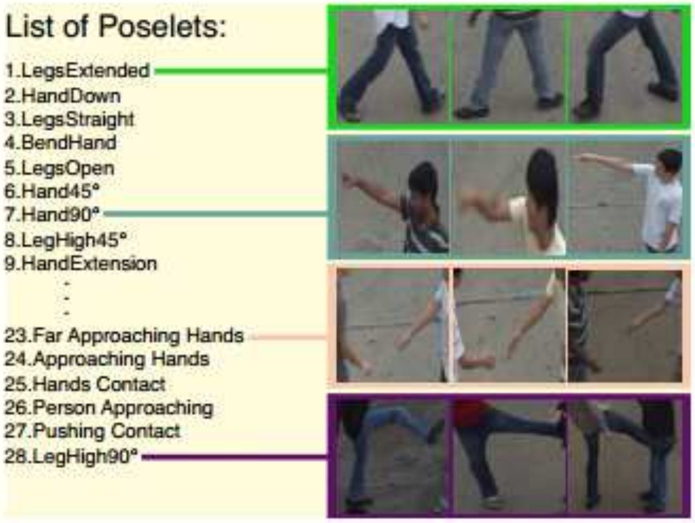


Figure 2.8: Examples of annotated poselets [75].

human figure into horizontal slices which approximately contain head, body and legs. Tran et al. [96] build rectangular grids of silhouettes and optical flows.

2.2.3 Body part based methods

Body part based approaches utilize information such as body part positions and movements, and build the relationship between body parts. Raptis et al. [75] represent an activity as a sparse sequence of discriminative key frames which is a collection of partial key-poses of the subjects depicting key states in the activity sequence. This method relies on a collection of poselets to characterize video frames. Figure 2.8 shows several examples of annotated poselets. The key frames are inferred by a max-margin discriminative framework where key frames are treated as latent variables.

Another method is of Kong et al. [46], which also captures inter-dependencies at action level and body part level to distinguish activities instead of inferring key poses. Firstly, they apply a pedestrian detector and tracker to obtain subject trajectories. They combine a large scale global feature and local features of body parts to represent the action of each subject. Then, activities are predicted by the co-occurrence of individual actions, e.g. activity = { action, action }. Generally, body part based approaches require a method for localizing persons, thus they intrinsically rely on the quality of human detection and tracking methods.

Chapter 3

Datasets

3.1 Overview of activity recognition datasets

Public dataset provide common criterion to measure and compare accuracies of proposed approaches. Therefore, a construction of a dataset containing videos of human activities plays a vital role in the advancement of human activity recognition research. In this section, we give an overview of human activity datasets which are currently available, and discuss the characteristics of the datasets. As previously explained, we classify video data material into two categories: unrealistic video data, and realistic video data.

For unrealistic video data, KTH [84] and Weizmann [6] are two typical examples, and are designed to report activity recognition performance. The KTH dataset contains 6 actions: walking, jogging, running, boxing, hand waving, and hand clapping which are performed by 25 subjects. Similarly, the Weizmann dataset consists of 10 relatively simple actions: walking, running, jumping, galloping sideways, bending, one-hand waving, two-hands waving, jumping in place, jumping jack, and skipping. These datasets are recorded in simplified conditions in which only one single person appears in videos taken from a fixed point and with a homogeneous background.

There is a growing need for designing new datasets which capture a wider range of actions in more complex background. Attempts have been made to record video clips in more realistic conditions such as MSR Action [112] and UT-Tower [13]. The UT-Tower dataset is designed to explore recognition techniques which address the issues of classifying human actions in low-resolution videos and from a distance view. It contains 9 categories of human actions: pointing, standing, digging, walking, carrying, runing, waving 1, waving 2, and jumping; and faces several challenges such as low resolution, illumination conditions, and shadows.

Furthermore, realistic video data can be gathered directly from TV, movies, and webs. Datasets like Hollywood2 [62], UCF Sports [78], UCF50 [77], UCF11 [57], Ollympic Sports [69], HMDB51 [49] belong to this category. For example, the UCF Sports dataset is composed of 10 actions collected from various sports: diving, golf swinging, kicking, lifting, horse-back riding, running, skating, swinging, and walking. This dataset is challenging because it contains various background scenes, and viewpoints.

In addition to human actions, researchers also design datasets to explore human-human or human-object interactions in realistic conditions such as TV Human Interaction [72] and UT-Interaction [81]. Table 3.1 shows an overview of activity recognition datasets.

Table 3.1: Development of activity recognition datasets.

Complexity	Type of video data	Type of activity	Source	Dataset
Low	Unrealistic	Action	Recorded videos (indoor/outdoor)	Weizmann [6] KTH [84]
	Realistic	Action	Recorded videos (indoor/outdoor)	UT-Tower [13] MSR Action [112]
⇓			Videos from web (indoor/outdoor)	Hollywood2 [62] UCF Sports [78] UCF50 [77] UCF11 [57] Ollympic Sports [69] HMDB51 [49]
High		Interaction	Recorded videos and TV shows	UT-Interaction [81] TV Human Interaction [72]

3.2 Datasets for experimental evaluation

As previously mentioned, this dissertation addresses the issue of automatic *classification* and *temporal localization* of human-human interactions, which is the most challenging topic in activity recognition. For human-human interaction recognition, there are two

Table 3.2: Summary of the statistics of the UT-Interaction dataset.

Number of classes	6
Number of video sequences	20
Resolution	720 × 480 pixels
Frame rate	30 fps
Average duration	1 min.
Average execution per video sequence	8
Number of subjects per video sequence	$2 \sim 4$
Average height of subject	200 pixels

popular datasets: UT-Interaction [81] and TV Human Interaction [72]. We focus on surveillance application, hence, the UT-Interaction dataset is chosen to evaluate the effectiveness of our classification and temporal localization method. Besides, we also find out how well our methods work on another dataset, for example Hollywood2 dataset [62].

3.2.1 The UT-Interaction dataset

Description

The UT-Interaction dataset [81] is designed to encourage researchers to explore recognition of complex human activities, e.g. human-human interactions, from videos taken in realistic settings. It includes videos of continuous executions of six classes of two-person interactions: shake-hands, hug, kick, point, punch and push. It contains 20 video sequences, whose lengths are around 1 minute. The dataset is recorded with the resolution of 720 × 480, 30 fps, and the height of a person in each video sequence is about 200 pixels. Besides, several subjects with more than 15 different clothing conditions appear in video sequences. There is at least one execution per interaction in each video sequence, which provides 8 executions of human interactions per video on average. Time intervals, bounding boxes, and ground truth labels of all interaction executions are provided for evaluation of classification and localization methods. Table 3.2 shows the characteristics of this dataset.

The dataset is divided into two sets: Set 1 and Set 2. Each set is composed of 10 video sequences. The videos of the Set 1 are recorded on a parking lot with slightly different zoom rate, and little camera jitter. Generally, their backgrounds are mostly static. Figure 3.1 shows example snapshots of two-person interactions in Set 1. From video sequences 1 to 4 of the Set 1, there are only two interacting subjects which appear in these scenes.

However, from video sequences 5 to 8, interacting subjects and pedestrians are present in the scene. Video sequence 9 and 10 are more complex because there are two pairs of interacting subjects performing interactions simultaneously.

Similarly, Set 2 also contains 10 video sequences (e.g. from sequence 11 to 20), which are taken at a lawn on a windy day. Figure 3.2 shows some example snapshots in Set 2. There are only two subjects which perform interactions from video sequences 11 to 13. In video sequences 18, 19, and 20, two pairs of interacting subjects perform interactions concurrently. Table 3.3 describes the characteristics of video sequences in detail.

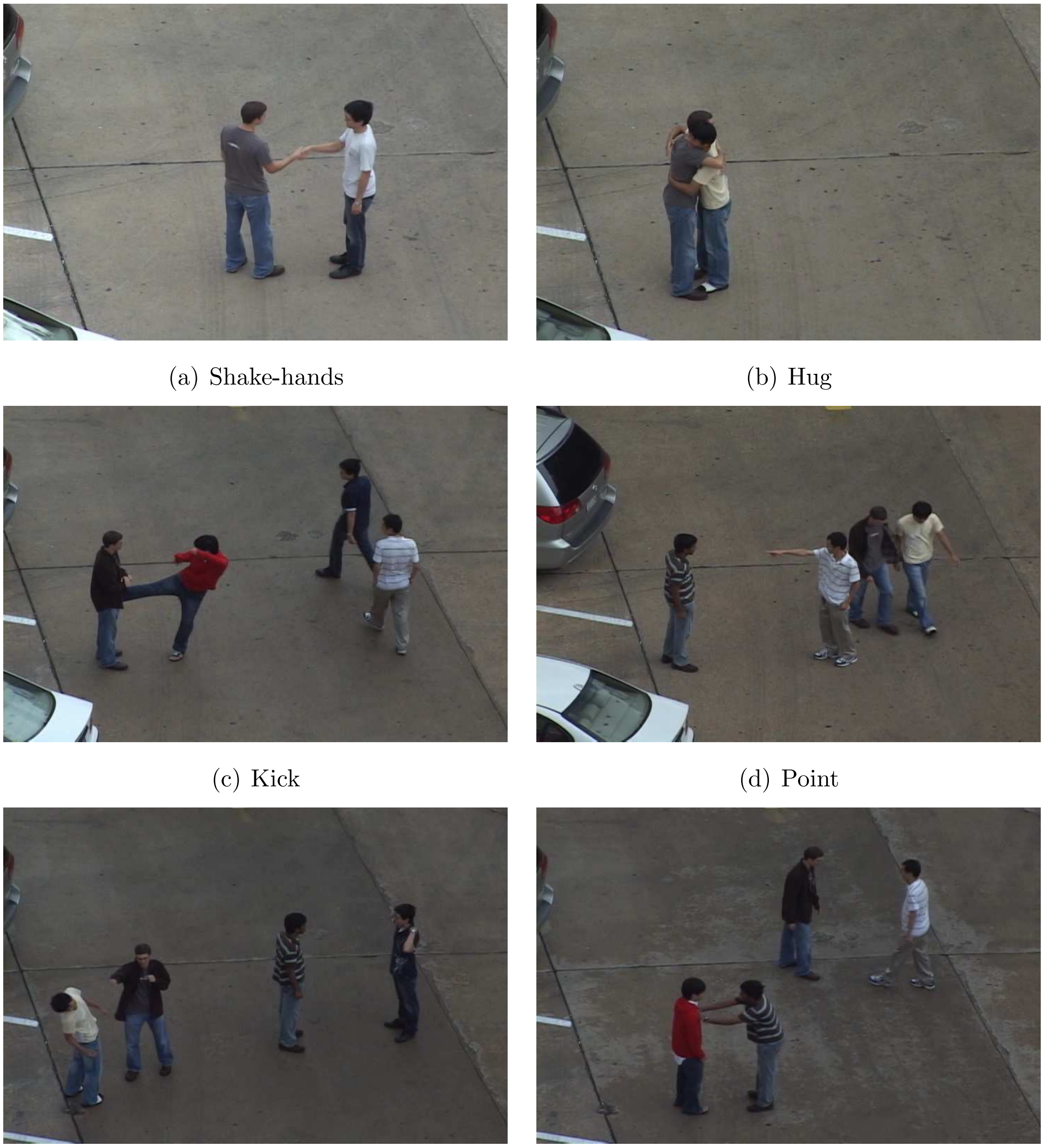
Table 3.3: Description of the UT-Interaction dataset. 'Pedestrian' indicates whether the scene of video sequence contains irrelevant pedestrians. 'Simultaneous interactions' describes whether the video sequence consists of two pairs of simultaneous executions.

	Sequences	Number of subjects	Pedestrian	Simultaneous interactions
Set 1	1 - 4	2	×	×
	5 - 8	2	○	×
	9 - 10	4	×	○
Set 2	11 - 13	2	×	×
	14 - 17	2	○	×
	18 - 20	4	×	○

Evaluation metrics

We perform experimental evaluation on two types of tasks: the classification task and temporal localization task. For the classification task, we selected 120 interaction executions (i.e. 60 executions for each set). The interaction executions are extracted by segmenting the video sequences spatially and temporally based on provided bounding boxes and ground truth time intervals. Finally, 120 video segments are obtained and used for the training and testing (i.e. 60 video segments for each set). We followed the classification settings described for the ICPR 2010 contest [81], where each set is evaluated separately. We performed 10-fold leave-one-out cross validation for each set. It means that for each round, we leaved 6 video segments for the testing, and use the other 54 video segments for the training. The performance is evaluated in terms of Accuracy, Precision, and Recall [90], which are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3.1}$$



(a) Shake-hands

(b) Hug

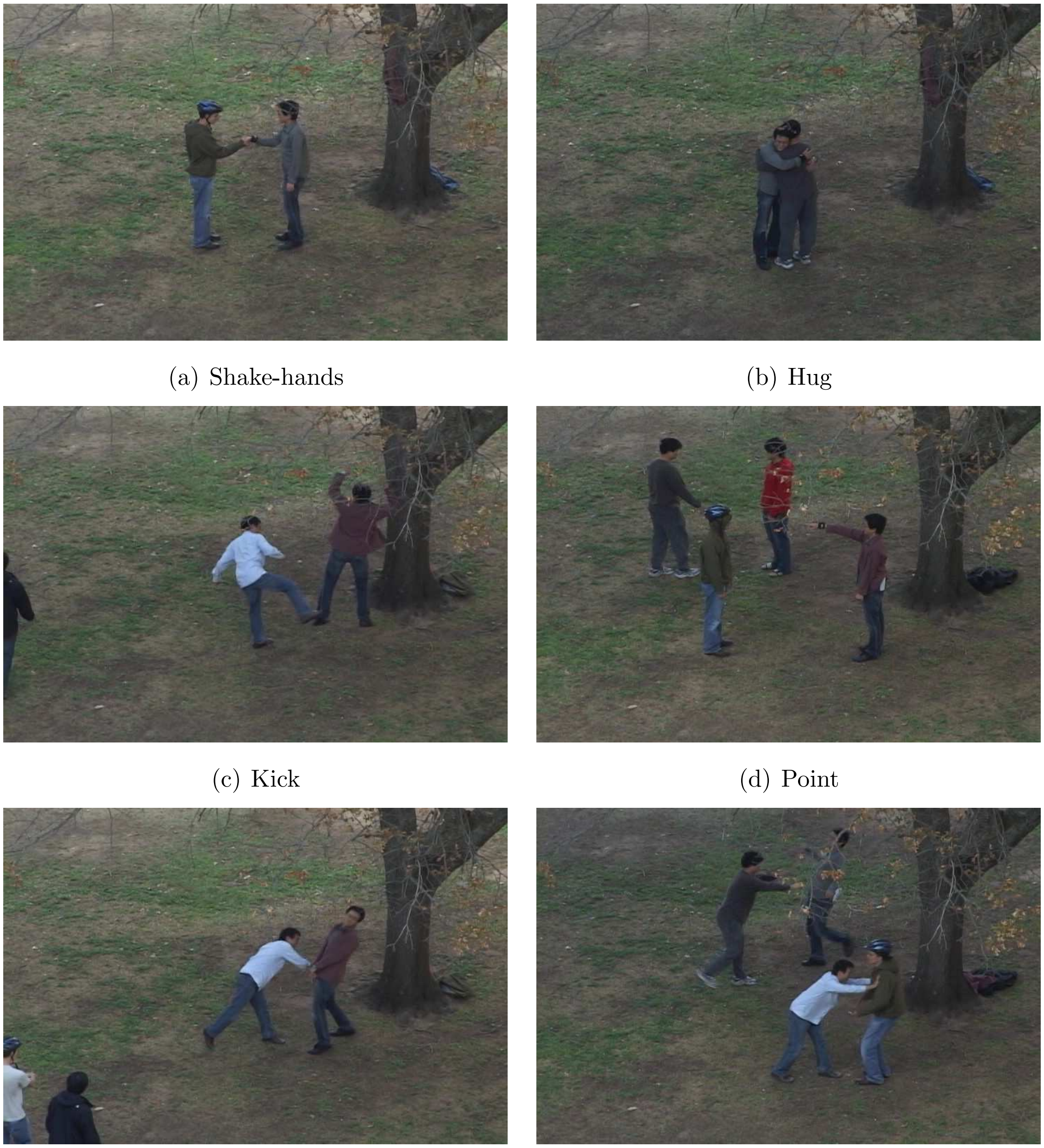
(c) Kick

(d) Point

(e) Punch

(f) Push

Figure 3.1: Example snapshots of six classes of two-person interactions in Set 1.



(a) Shake-hands

(b) Hug

(c) Kick

(d) Point

(e) Punch

(f) Push

Figure 3.2: Example snapshots of six classes of two-person interactions in Set 2.

Table 3.4: Summary of the statistics of the Hollywood2 dataset

Number of action classes	12
Number of videos for training	823
Number of videos for testing	884
Resolution	Min: 224 × 528, max: 576 × 720
Frame rate	25 fps
Number of scenes	10

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

where TP, TN, FP, and FN stand for true positive, true negative, false positive and false negative respectively.

Similarly, we also followed 10-fold leave-one-out cross validation per set to evaluate our localization method. However, for localization task, we used the video sequences instead of video segments. Hence, for each round, we leaved one among 10 sequences for the testing and used the other 9 for the training. The objective of temporal localization is to classify an occurring activity's class and annotate it's time interval correctly. If the annotation overlaps with the ground truth time intervals more than 50%, we treat it as a true positive. Otherwise, it is considered as false positive. We also report our experimental results in terms of precision and recall.

3.2.2 The Hollywood2 dataset

Description

The Hollywood2 dataset [62] is constructed by collecting realistic videos from 69 different Hollywood movies. It consists of 12 classes of human actions including answer phone, drive a car, eat, fight person, get out of a car, shake hands, hug, kiss, run, sit down, sit up, and stand up. Figure 3.3 shows sample frames of these action classes. Table 3.4 provides some properties of the Hollywood2 dataset. There are 10 video scenes including house, road, bedroom, car, hotel, kitchen, living room, office, restaurant, and shop. The dataset is built to encourage the development of recognition systems that can recognize various actions under noise, viewpoint changes. Table 3.5 shows the distributions of class instances in the training and test set. Figure 3.3 and Figure 3.4 show several sample frames for each action class of this dataset.

Table 3.5: Distributions of class instances of the Hollywood2 dataset.

	Training set	Test set
Answer phone	66	64
Drive a car	85	102
Eat	40	33
Fight person	54	70
Get out car	51	57
Shake hands	32	45
Hug person	64	66
Kiss	114	103
Run	135	141
Sit down	104	108
Sit up	24	37
Stand up	132	146
All samples	823	884

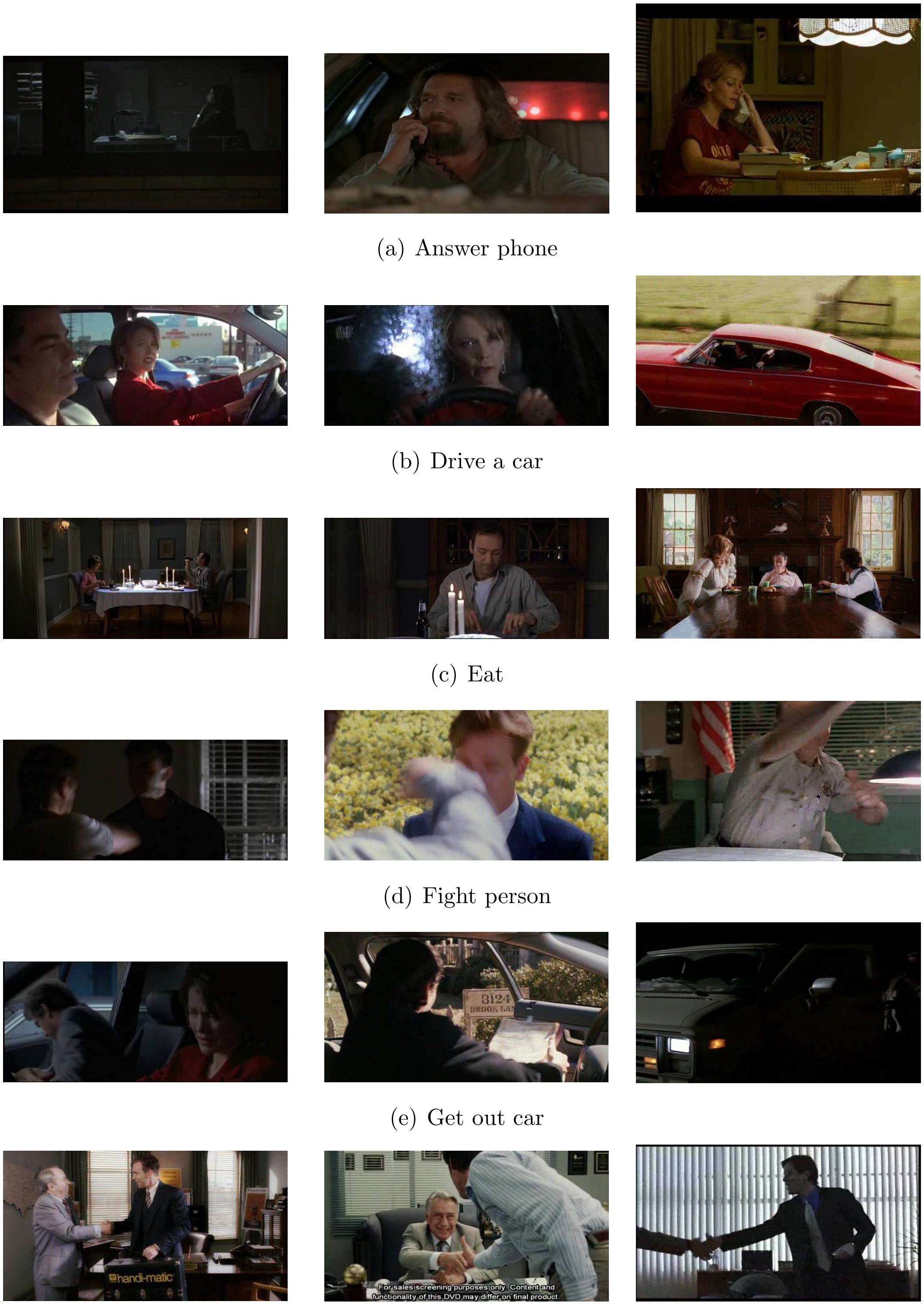
Evaluation metrics

In our experiments, we used the clean training dataset which has 823 training samples while the test set has 884 samples. The performance for the Hollywood2 is evaluated as suggested in [62], i.e., by computing the average precision (AP) for each of the action classes and reporting the mean AP over all classes (mAP). The average precision (AP) is defined as follows:

$$\mathrm{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1.0\}} P(r) \tag{3.4}$$

$$P(r) = \max_{\tilde{r}:\, \tilde{r} \geq r} P(\tilde{r}) \tag{3.5}$$

where $P(r)$ is interpolated precision that takes the maximum precision over all recalls which are greater than r.


handi-matic
3124
BROOK LA
For sales screening purposes only. Content and functionality of this DVD may differ on final product

(a) Answer phone

(b) Drive a car

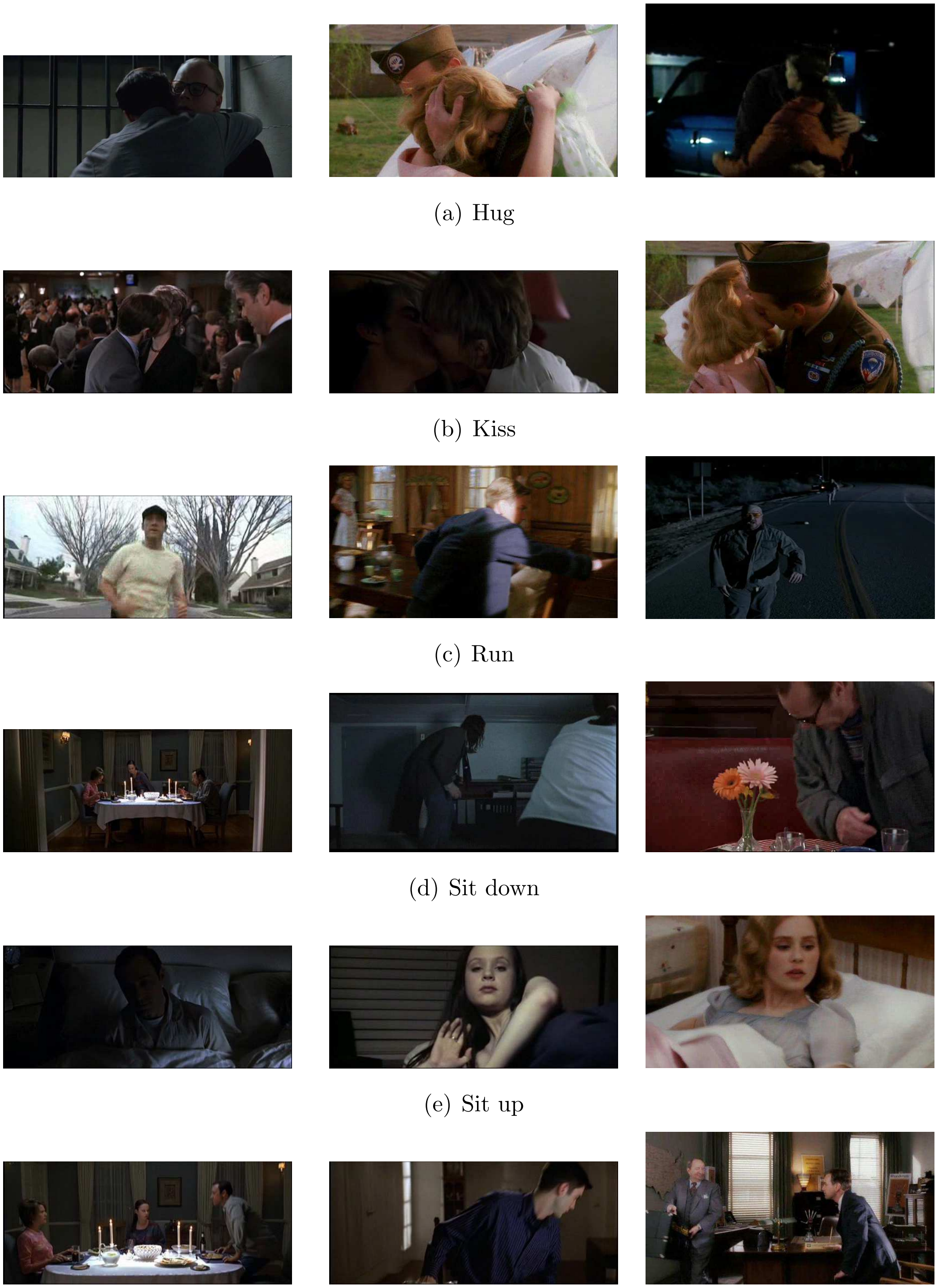
(c) Eat

(d) Fight person

(e) Get out car

(f) Shake hands

Figure 3.3: Sample frames for the Hollywood2 action dataset. Three samples are given for each of the twelve action classes.



(a) Hug

(b) Kiss

(c) Run

(d) Sit down

(e) Sit up

(f) Stand up

Figure 3.4: Sample frames for the Hollywood2 action dataset. Three samples are given for each of the twelve action classes (continue from Figure 3.3).

Chapter 4

Interaction Recognition using Hierarchical Invariant Features

4.1 Introduction

Action recognition in simple videos, such as KTH dataset [84] and Weizmann dataset [6] has shown promising results [10, 42, 87]. Recent efforts have been put in place to analyze activities with more complex structures, e.g. human-human interactions. Human-human interactions are more complicated compared with simple actions because of several reasons. One reason is that the causal relationships between two persons are complicated. For example, in a 'punch' interaction, one person moves to attack, and the other reacts. Another reason is that individual movements in different interaction classes could be similar and thus difficult to be discriminated. In this chapter, we focus on *human-human interactions* and address the problem of *classification*.

Many state-of-the-art activity models are based on 3-dimensional local spatio-temporal features [9, 19, 47, 52, 107] such as SIFT [85], and HOF [53] descriptors to model human interactions. These features are robust to noise, small camera jitters, and sudden changes in lighting conditions. Ryoo and Aggarwal [80] presented a kernel function which is designed to measure the structural similarity between sets of local features extracted from two videos. They considered temporal relations (e.g. equal, before, meet) and spatial relations (e.g. near, far) of these local features to evaluate the similarity between the structures of two videos. Similar to [80], Gaur et al. [28] also extracted local features and used them to build their model. They represented videos as graphs of these local features which respect their spatio-temporal relations. Hence, the problem of evaluating the similarity of two videos is equivalent to find correspondences between the two graphs.

Other approaches [3,21,46,75,86,98,102] focused on representing interactions in terms of atomic-level actions and analyzing contextual information of these actions, such as mutual dependencies of atomic actions and inter-dependencies between body parts. For example, an interaction ‘shake-hands’ can be recognized if the atomic actions of two persons are correctly classified as ‘stretch hand’. Vahdat et al. [98] represented an activity as a sequence of key poses which captures important atomic-level actions of two individuals, and formulated temporal orderings and spatial relations among the locations of these key poses. They presented an efficient dynamic programming algorithm to infer the key poses, and learned parameters for their model by using a max-margin criterion. Kong et al. [46] presented a hierarchical model which captures inter-dependencies at action level and body part level to distinguish various human interactions. They combined a large scale global feature and local features of body parts to represent the action of each individual. Then, human interactions are predicted by the co-occurrence of individual actions. In general, these approaches are effective to represent complex human interactions, and improve classification accuracy. However, one of the biggest limitations of the approaches is that they depend on human detection results and tracking algorithm, which are also challenging issues in realistic settings. Furthermore, atomic actions also have to be defined manually, and they are different for video sources.

Additionally, skeleton-based approaches have been considered by several researchers, e.g. [64], [113]. Meng et al. [64] introduced a discriminative function based on appearance features and spatial relations within each individual and between two persons. These spatial relations are calculated by analyzing the pairwise relative locations among extracted joints. Similarly, Kiwon et al. [113] explored geometric relational features including joint, plane and velocity features. Then, a Multiple Instance Learning-based classifier is applied to recognize human interactions. Body joints are detected by training manually annotated joints as in [64], or by using Kinect sensors as in [113].

This chapter focuses on finding a representation for video sequences of interactions and actions recorded in realistic settings. Inspired by recent success in deep learning networks, we introduce a three-layer convolutional network which uses the Independent Subspace Analysis (ISA) to learn hierarchical invariant features. The ISA algorithm is a generalization of the Independent Component Analysis (ICA), which is very well-known in natural image statistics. Compared to the ICA algorithm, the most notable advantage of the ISA is that it can learn features which are invariant to phase while being selective to orientation and frequency. However, the ISA algorithm will become slow if the dimension of input data is large. In order to solve this computational problem, we combine the idea of convolutional neural network with the ISA algorithm. Specifically, instead of training the ISA algorithm directly on raw video data, we first train it on small video blocks. The obtained features are then convolved with larger video blocks. The outputs of this convolution step are fed into the next layer, which is implemented by another ISA algorithm. This organization enables the three-layer convolutional ISA network to learn hierarchical representation for video data.

The main contributions of this proposed method are as follows:

(i) Devise a procedure for video block extraction to enhance foreground information of sampled video blocks.

(ii) Build a three-layer convolutional ISA network to learn hierarchical invariant features for videos by unsupervised learning.

(iii) Introduce a pooling layer to reduce the contribution of features in lower layers while still achieving translation invariant.

Outline

First, in Section 4.2, we present an overview of the Independent Component Analysis and Independent Subspace Analysis for image data. Second, Section 4.3 describes our approach in more detail including the specifics of video block extraction, the three-layer convolutional ISA network and the pooling layer. The interaction recognition model based on bag-of-features is presented in Section 4.4. Finally, we present parameter settings and experimental results on the public UT-Interaction dataset [81], and the Hollywood2 dataset [62] in Section 5.3. In addition, we show the comparison results on the performance of our method and the other methods, and investigate the importance of different components of our method.

4.2 Independent Subspace Analysis (ISA) for image data

Definition of the ISA and it's algorithm

Independent Component Analysis (ICA) [37] is a statistical model, which is defined by a linear transformation of latent independent variables. In particular, let $\mathbf{x}^t$ denote the grey-scale values in a small image patch, the ICA model expresses $\mathbf{x}^t$ as a linear superposition of some features **A**:

$$\mathbf{x}^t = \mathbf{A}\mathbf{s} \tag{4.1}$$

where **s** is a vector whose elements are components (or coefficients). Note that **s** is different from patch to patch. The matrix **A** is the same for all patches.

The basic assumption in the ICA model is that the components **s** are nongaussian and statistically independent. Given a sufficient number of observations of image patches, the problem is then to estimate the values of **A** without knowing the values of latent components **s**. This problem is restricted to the basic case where **A** is an invertible matrix. Hence, estimation of **A** in Eq. (4.1) is equivalent to determining the values of **W** in Eq. (4.2):

$$\mathbf{s} = \mathbf{W}\mathbf{x}^t \tag{4.2}$$

where **W** is obtained by inverting the matrix **A**.

Independent Subspace Analysis (ISA) [35] is an interesting generalization of the basic ICA, and has the same model as in Eq. (4.1). In contrast to the ICA, the components **s** are not assumed to be statistically independent. In the ISA model, **s** can be divided into couple, triplet, or in general κ-tuples where κ is the dimension of subspace. The ISA model assumes that the components inside a given κ-tuple may be dependent on each other, but dependencies among different κ-tuples are not allowed.

Figure 4.1 represents the ISA as a two-layer network, where the elements of the matrix **W** in Eq. (4.2) are weights in the first layer. In this figure, the dimension of subspace is 2 ($\kappa = 2$). The objective of the ISA is to learn the weights **W** while the weights **V** in the second layer are fixed to represent the subspace structure of the units in the first layer.

Let $\mathbf{x}^t \in \mathbb{R}^{n\times 1}$ again denote the input patch, the response of $l-$th unit in the first layer is defined by Eq. (4.3):

$$e_l = (\sum_{j=1}^{n} \mathbf{W}_{lj}\mathbf{x}_j{}^t)^2 \tag{4.3}$$

where $\mathbf{W} \in \mathbb{R}^{k\times n}$ is the connection weights of the first layer; n and k are the input dimension and number of units in the first layer.

As illustrated in Figure 4.1, each unit of the second layer pools over a small neighborhood of adjacent first layer units. Hence, the response of each second layer unit is defined by Eq. (4.4):

$$f_i(\mathbf{x}^t; \mathbf{W}, \mathbf{V}) = \sqrt{\sum_{l=1}^{k} \mathbf{V}_{il} e_l} = \sqrt{\sum_{l=1}^{k} \mathbf{V}_{il} (\sum_{j=1}^{n} \mathbf{W}_{lj}\mathbf{x}_j{}^t)^2} \tag{4.4}$$

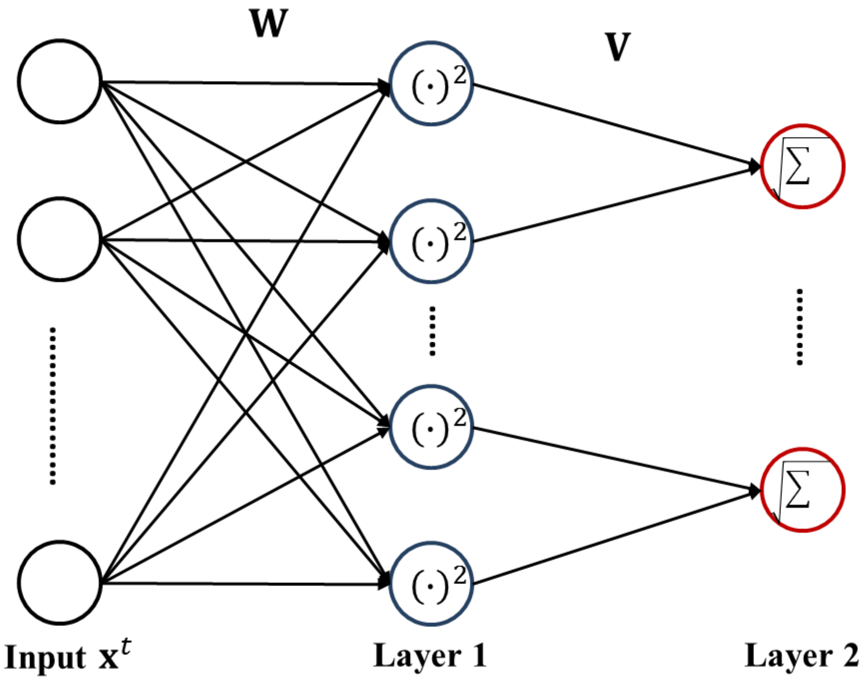


Figure 4.1: The neural network architecture of an ISA network. The blue and red bubbles represent units in the first and second layer respectively. In this figure, the dimension of subspace is 2: each red bubble looks at 2 blue bubbles.

where $\mathbf{V} \in \mathbb{R}^{m \times k}$ is the weights connecting units of the first layer to units of the second layer, and m is the number of units in the second layer. The matrix $\mathbf{V}$ represents the subspace structure of the units in the first layer, and is defined by Eq. (4.5):

$$\mathbf{V}_{il} = \begin{cases} 1, & \text{if } (i-1)\kappa + 1 \leq l \leq (i-1)\kappa + \kappa \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

where V_{il} represents the weight connecting from l-th unit in the first layer to i-th unit in the second layer, and κ is the dimension of subspace. It is important to note that the number of units in the first layer is divisible by the dimension of subspace ($k = \kappa m$).

Given T image patches $\mathbf{x}^t, t = 1, ..., T$ (or $\{\mathbf{x}^t\}_{t=1}^T$), the ISA algorithm learns the weights $\mathbf{W}$ through finding sparse feature representations in the second layer by solving:

$$\begin{aligned} \underset{\mathbf{W}}{\text{minimize}} \quad & \sum_{t=1}^{T} \sum_{i=1}^{m} f_i(\mathbf{x}^t; \mathbf{W}, \mathbf{V}), \\ \text{subject to} \quad & \mathbf{W}\mathbf{W}^T = \mathbf{I} \end{aligned} \tag{4.6}$$

where $\{\mathbf{x}^t\}_{t=1}^T$ are input patches which are whitened by linearly transforming to have zero mean and identity covariance.

Algorithm 1 Batch projected gradient descent

```
Input: ${\mathbf{x}^t}_{t=1}^{T}, \mathbf{V}, n, k, m$
Output: $\mathbf{W}$
Randomize initial values of weights $\mathbf{W}$
$\mathbf{W} \leftarrow \text{proj}_U \mathbf{W}$, where $U$ is the space of matrices satisfying $\mathbf{W}\mathbf{W}^T = I$
repeat
    $\nabla \mathbf{W} \leftarrow \frac{\partial \sum_{t=1}^{T} \sum_{i=1}^{m} f_i(\mathbf{x}^t;\mathbf{W},\mathbf{V})}{\partial \mathbf{W}}$
    $\mathbf{W} \leftarrow \mathbf{W} - \alpha \nabla \mathbf{W}$, where $\alpha$ is learning rate
    $\mathbf{W} \leftarrow \text{proj}_U \mathbf{W}$
until convergence
```

Learning the weights $\mathbf{W}$, which is based on batch projected gradient descent on the objective function Eq. (4.6), is shown in Algorithm 1. The gradient of the objective function is obtained according to Eq. (4.7):

$$\frac{\partial f_i(\mathbf{x};\mathbf{W},\mathbf{V})}{\partial \mathbf{W}} = (\sum_{l=1}^{k} \mathbf{V}_{il}(\sum_{j=1}^{n} \mathbf{W}_{lj}\mathbf{x}_j{}^t)^2)^{\frac{-1}{2}} \sum_{l=1}^{k} \mathbf{V}_{il}(\sum_{j=1}^{n} \mathbf{W}_{lj}\mathbf{x}_j{}^t)\mathbf{x}_j{}^t \tag{4.7}$$

The ISA algorithm requires orthonormal constraint; hence the projection step proj_U is called during optimization as shown in Algorithm 1. The projection step proj_U is achieved by Eq. (4.12):

Preprocessing for the ISA algorithm

Before applying the ISA on the data, it is very useful to do some preprocessing techniques including *centering* and *whitening* to make the problem of ISA estimation simpler and better conditioned. The most basic necessary technique is to center image data by removing its DC component. The DC component refers to the mean grey-scale value of an image patch, and it is often assumed that the DC component does not contain interesting information. Hence, the DC component is often removed from the image to simplify solely further analysis of the ISA algorithm.

Another useful preprocessing strategy in the ISA is to whiten data. It is intuitively rather clear that raw images typically redundant in the sense that adjacent pixel values are highly correlated (i.e. pixel values of two nearby pixels are very similar). The objective of whitening is to make the raw input images less redundant. Or to phrase it another way, let $\mathbf{x}$ denote a vector, the whitening technique transforms $\mathbf{x}$ into a new vector $\tilde{\mathbf{x}}$, which is white, i.e. its components are uncorrelated and their variances equal unity.

One popular method for whitening is to use the eigenvalue decomposition (EVD) of the covariance matrix $\Sigma = \mathrm{E}\{\mathbf{x}\mathbf{x}^T\}$. Then, let us compute the eigenvectors of Σ, and stack the eigenvectors in column to form the matrix $\mathbf{U}$:

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix} \tag{4.8}$$

where $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_n$ are eigenvectors corresponding to eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$ (λ_1 is the largest eigenvalue). Let $\mathbf{D}$ be the diagonal matrix of the eigenvalues:

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \tag{4.9}$$

Whitening can now be done by

$$\tilde{\mathbf{x}} = \mathbf{U}\mathbf{D}^{\frac{-1}{2}}\mathbf{U}^T\mathbf{x} \tag{4.10}$$

where the matrix $\mathbf{D}^{\frac{-1}{2}}$ is obtained by a simple component-wise operation as:

$$\mathbf{D}^{\frac{-1}{2}} = \begin{bmatrix} \lambda_1^{\frac{-1}{2}} & 0 & \dots & 0 \\ 0 & \lambda_2^{\frac{-1}{2}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n^{\frac{-1}{2}} \end{bmatrix} \tag{4.11}$$

$$\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}}\mathbf{W} \tag{4.12}$$

Analysis of the ISA

Figure 4.2 shows typical filters learned from the ISA algorithm on 50000 image patches of size 32 × 32 when subspace dimension is chosen to be 2 ($\kappa = 2$). The patches are obtained at random locations from grayscale images, and converted into column vectors of length 1024. Before applying the ISA algorithm, these vectors are then whitened by linearly transforming to have zero mean and identity covariance.

In Figure 4.2, we visualize six groups of filters (12 filters in total), each row consists of three groups of filters. Eeach group of two filters spans a feature subspace. It is clearly seen that the ISA algorithm is able to learn Gabor filters (edge detectors) with many frequencies and orientations. In addition, filters in the same feature subspace have similar orientations and frequencies. Their locations are not identical, but near to each other as well. The ISA algorithm assembles similar filters into a feature subspace and thereby achieves invariant property.

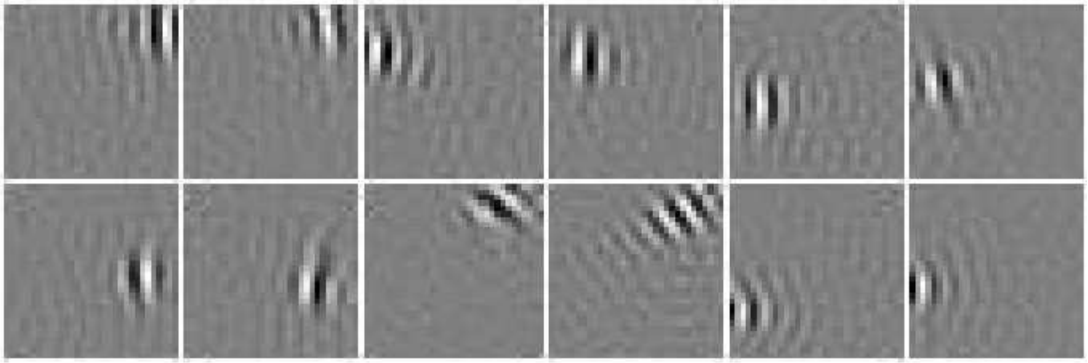


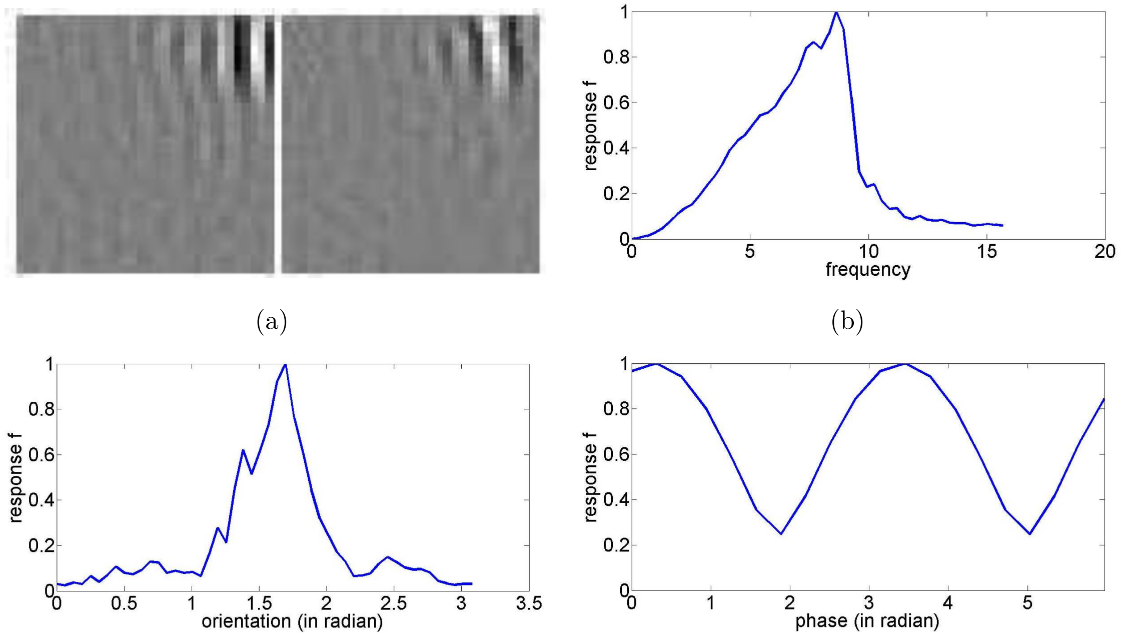
Figure 4.2: Six groups of filters (produced by $\mathbf{W}$) learned from the ISA algorithm when trained on images patches. Each row contains three groups of filters. The grey-scale value of a pixel means the value of coefficient (weight) at that pixel. Grey pixels mean zero coefficients. Each group of two filters represents a feature subspace (or a neuron in the second layer).

To analyze the properties of the ISA algorithm, we look at the response f_i of a representative feature subspace in different stimulus configurations. First, we find the optimal stimulus for the feature subspace by fitting a parametric Gabor function [34]. The stimulus parameters are phase, orientation, and frequency. Then, we vary one of the stimulus parameters to see how the response changes while holding other stimulus parameters constantly at the optimal values. Figure 4.3 shows the analysis for a particular feature subspace. We can clearly see that the feature subspace has phase invariance while it is selective to orientation and frequency. The combination of selectivity to orientation and frequency with phase invariance makes feature subspaces be good candidates for representing image data.

4.3 Three-layer convolutional ISA network

The ISA training algorithm requires orthonormal constraint, hence during optimization the weights $\mathbf{W}$ are projected to the constraint set by computing $(\mathbf{W}\mathbf{W}^T)^{-\frac{1}{2}}\mathbf{W}$. The inverse square root of the matrix involves solving an eigenvector problem, therefore, its computational complexity grows as a cubic function of the input dimension. Therefore, the ISA training algorithm becomes slow when the dimension of the input data is large, such as video data.

In order to solve this computational problem, we combine the convolutional neural network with the ISA algorithm. Specifically, instead of training the ISA algorithm di-



(a)

(b)

(c)

(d)

Figure 4.3: Responses f_i of a feature subspace i. (a) Two underlying filters in a feature subspace. (b) Effect of varying frequency. (c) Effect of varying orientation. (d) Effect of varying phase. The response values are normalized so that the maximum response is equal to 1.

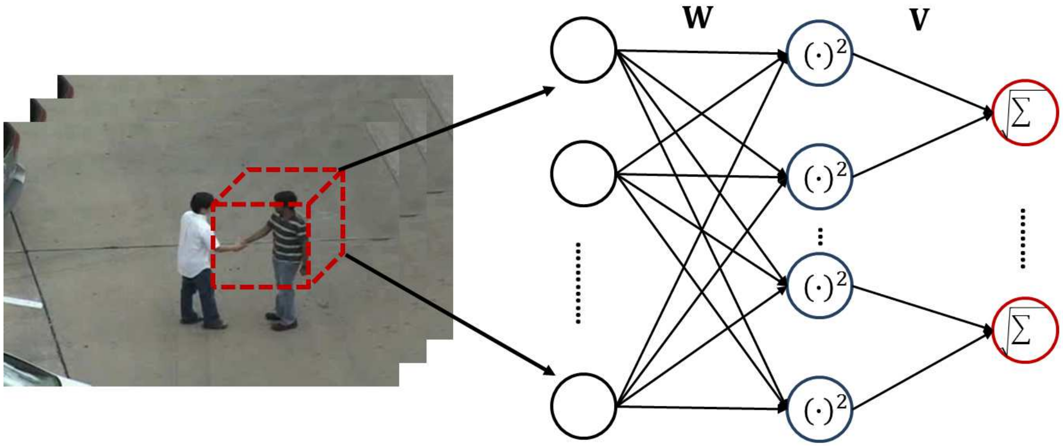


Figure 4.4: Graphical depiction of applying the ISA algorithm to video data.

rectly on video data, we first train it on small video blocks. The obtained filters are then convolved with a larger region of video data. The outputs of this convolution step are fed into the next layer, which is implemented by another ISA algorithm. By this model, we can learn a hierarchical representation of video data.

Figure 4.4 shows the major steps to apply the ISA algorithm to video data in our approach. Firstly, we extract video blocks instead of image patches and then transform them into vectors. For example, if a video block $\mathbf{B}$ is of size $w \times h \times t$, it is converted into a column vector $\mathbf{x}^t$ as follows:

$$\mathbf{B} \rightarrow \mathbf{x}^t = \begin{bmatrix} \mathbf{B}_{111} \\ \cdots \\ \mathbf{B}_{w11} \\ \cdots \\ \mathbf{B}_{1h1} \\ \cdots \\ \mathbf{B}_{wh1} \\ \cdots \\ \mathbf{B}_{1ht} \\ \vdots \\ \mathbf{B}_{wht} \end{bmatrix} \tag{4.13}$$

These vectors are the inputs to the ISA algorithm. Secondly, we also employ following preprocessings to those inputs to simplify and speed up the ISA. The first preprocessing step is removing the DC component, which is the mean grey-scale value of the pixels in that video block. After that, principal component analysis (PCA) is used to whiten and reduce the dimension of the inputs. Then, the ISA learns the weights $\mathbf{W}$ with batch projected gradient descent.

4.3.1 Video block extraction

The ISA algorithm is often trained on randomly sampled video blocks. However, the background often occupies large areas in every single frame. This fact usually results in a high number of video blocks containing background information in sampled data. On the other hand, video blocks with foreground information play a far more important role in disambiguating similar human interactions. Therefore, it's desirable to have a high number of sampled foreground video blocks. To achieve that property, we devise a procedure to extract video blocks based on the frame differencing method.

First, for each training video we compute frame differences to detect all moving pixels. Then N video blocks are randomly sampled and sorted by their energies, which is the sum of intensities of all pixels within that block. Finally, we only keep M highest energy video blocks. The value of N is set to 500 in our experiments.

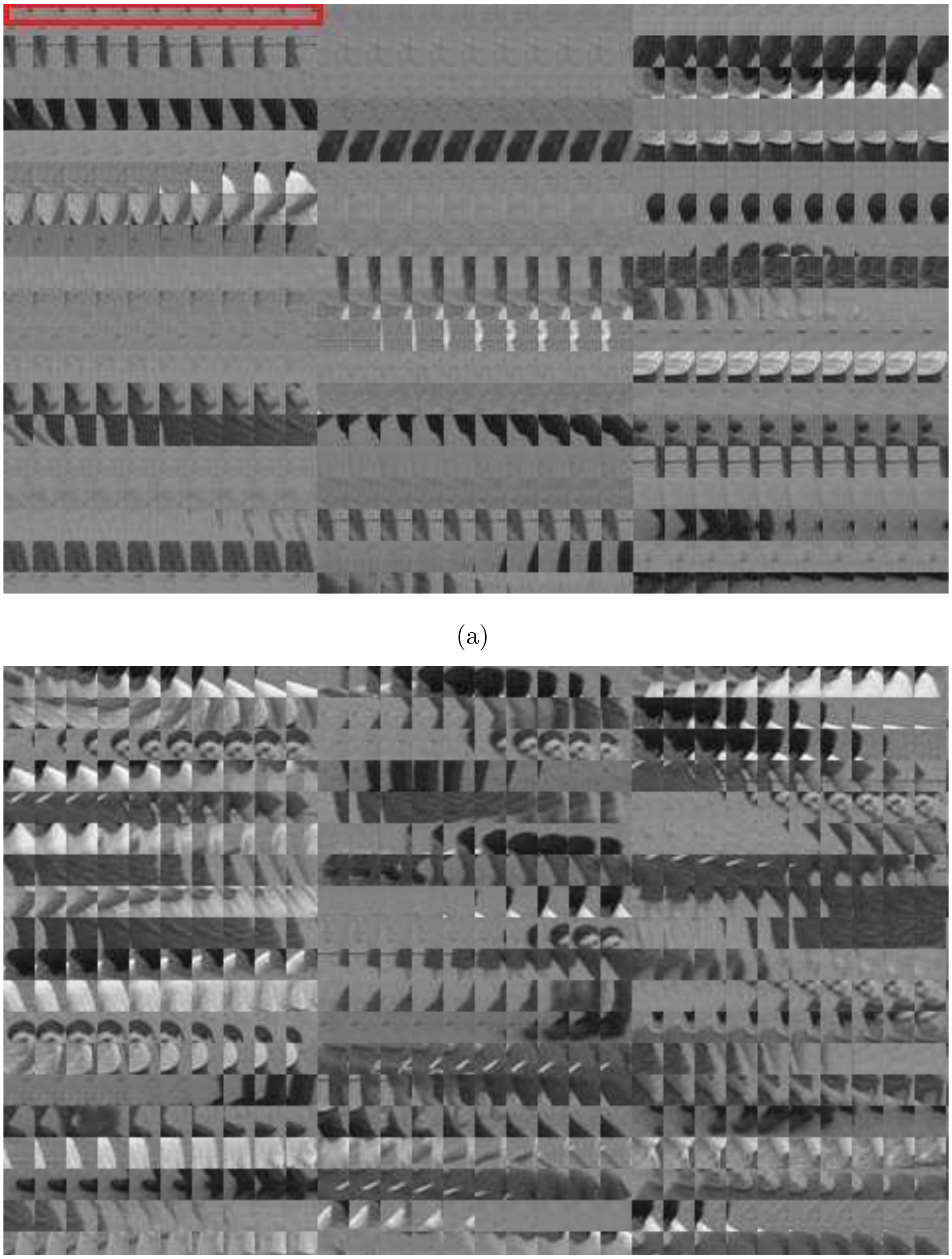
Figure 4.5 compares the foreground information in extracted video blocks with and without applying frame differencing. It is clear that the extracted video blocks in Figure 4.5b have more foreground information than the ones in Figure 4.5a, thus they support differentiating human interactions better.

4.3.2 Hierarchical invariant features

To learn high-level concepts and solve the computational problem of the ISA algorithm when trained on video data, we combine the convolution technique with the standard ISA algorithm to design a three-layer convolutional ISA network. This convolutional network uses PCA and ISA as sub-units to learn hierarchical invariant features from video data.

In particular, we extract video blocks of size $w_1 \times h_1$ (spatial dimensions) and t_1 (temporal dimension) in the first layer. They are fed into the ISA algorithm in the first layer which we call ISA1. The output of this layer is the weights W_1 and the subspace structure V_1.

Similarly, in the second layer, we extract video blocks of size $w_2 \times h_2 \times t_2$, which are independent of the ones in the first layer. In order to find hierarchical features, the dimensions of the video blocks in the second layer are set to be larger than the ones in the first layer. As a result, each block in the second layer can be seen as a collection of m overlapping video blocks of size $w_1 \times h_1 \times t_1$. Figure 4.6 illustrates this composition in detail. In the bottom of Figure 4.6, there is the biggest cube, which corresponds to a video block of size $w_2 \times h_2 \times t_2$ in the second layer. Inner blocks, such as the yellow, blue


(a)

(b)

Figure 4.5: Representation of extracted video blocks. Each video block is a sequence of frames. The red rectangle corresponds to a complete video block. The video blocks in (a) are randomly extracted from videos without applying frame differencing. The ones in (b) are obtained as a result of applying frame differencing.

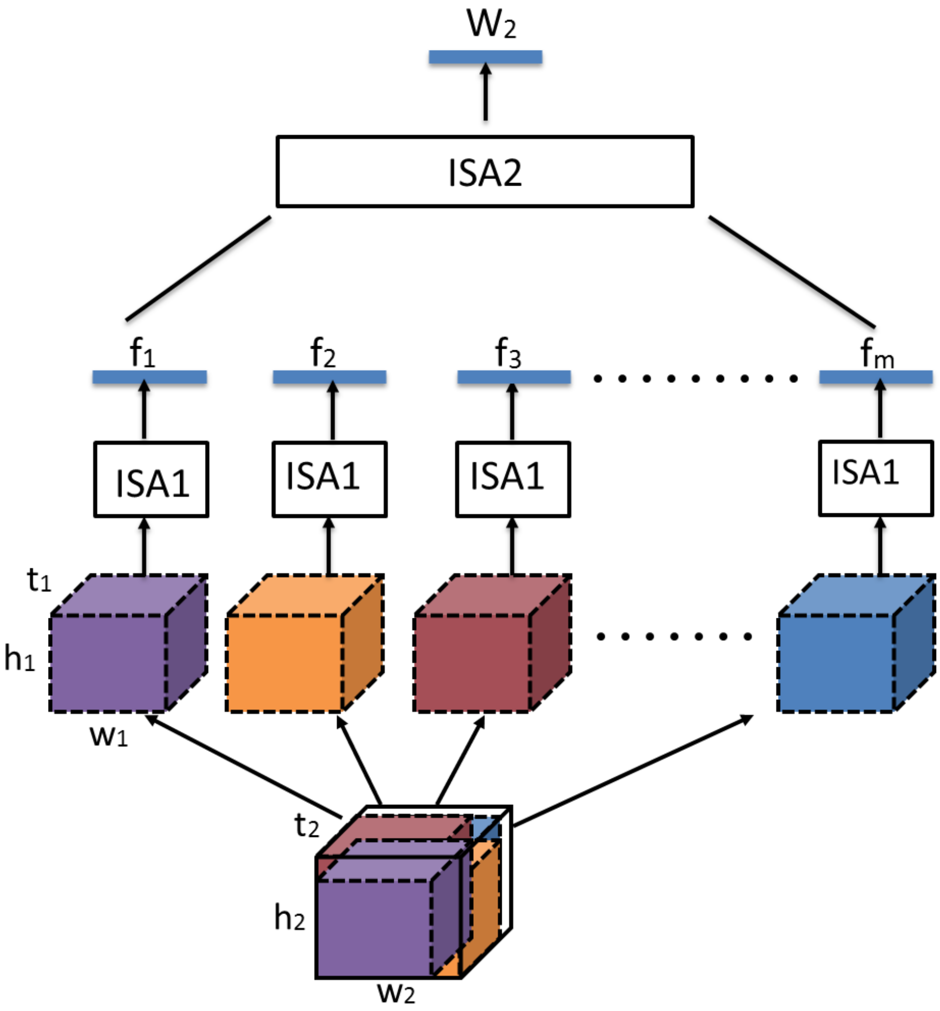


Figure 4.6: An illustration of the convolution step in the three-layer convolutional ISA network.

ones are of size $w_1 \times h_1 \times t_1$. These inner blocks are convolved with the weights W_1, V_1 of the ISA1 learned from the first layer. The responses f_1, f_2,..,f_m are concatenated to form the inputs to train ISA weights at the second layer. In the same way, we call the ISA algorithm in the second layer ISA2, and the learned weights are W_2, V_2. As mentioned above, before feeding the inputs into the ISA2, the PCA is performed to whiten and reduce the dimension of the input to speed up the time computation. Furthermore, we also use a greedy layered-wise approach to reduce the training time of the convolutional ISA network.

The same procedure is repeated at the third layer. For each video block of size $w_3 \times h_3 \times t_3$, we make a collection of overlapping video blocks of size $w_2 \times h_2 \times t_2$, which follow the same processing in the second layer. That means we further represent each video block of $w_2 \times h_2 \times t_2$ as a collection of video blocks of $w_1 \times h_1 \times t_1$. The smallest video blocks are convolved with W_1, V_1. The responses are then convolved with W_2, V_2 and are finally combined to form the inputs to the ISA algorithm in the third layer. Figure 4.7 shows a typical example of filters, which are learned by training the convolutional ISA network on the video blocks of size $16 \times 16 \times 10$.

In summary, the convolutional ISA network learns filters with small video blocks. The learned filters are then convolved with larger video blocks, and the outputs (or obtained features) are concatenated to form the inputs to the next layer. By doing this, we can learn a hierarchical representation of the data. In addition, the ISA algorithm is able to learn features which are invariant to phase, and selective to orientation and frequency. This will enable our convolutional ISA network to learn hierarchical invariant features.

4.3.3 Pooling

The convolutional ISA network learns simple features in the first layer, and more complex features in subsequent layers. However, simple features are not as significant for representing a video as the more complex ones. Hence, we try to reduce the contribution of features learned from the first layer by performing following steps. First, these features are processed by *mean spatial temporal pooling*. It is based on the stationary property of images. This property implies that features are useful in one region are also likely to be useful in other regions. Pooling operation aggregates statistics of features at various locations, and thereby achieves translation invariant (less-over fitting). Second, PCA is then applied to further reduce the number of features in the first layer.

Other more complex features learned from subsequent layers do not require much post-processing as the features in the first layer. Specifically, we adopt the *max pooling operation* on features of the second layer, and do nothing for features of the third layer.

In our experiments, after the pooling stage, there are 50 features in the first layer, and 100 features in the second layer. 50 features of the third layer are kept in the same. The features from three layers are concatenated to construct local features, which are inputs to the bag-of-features model.

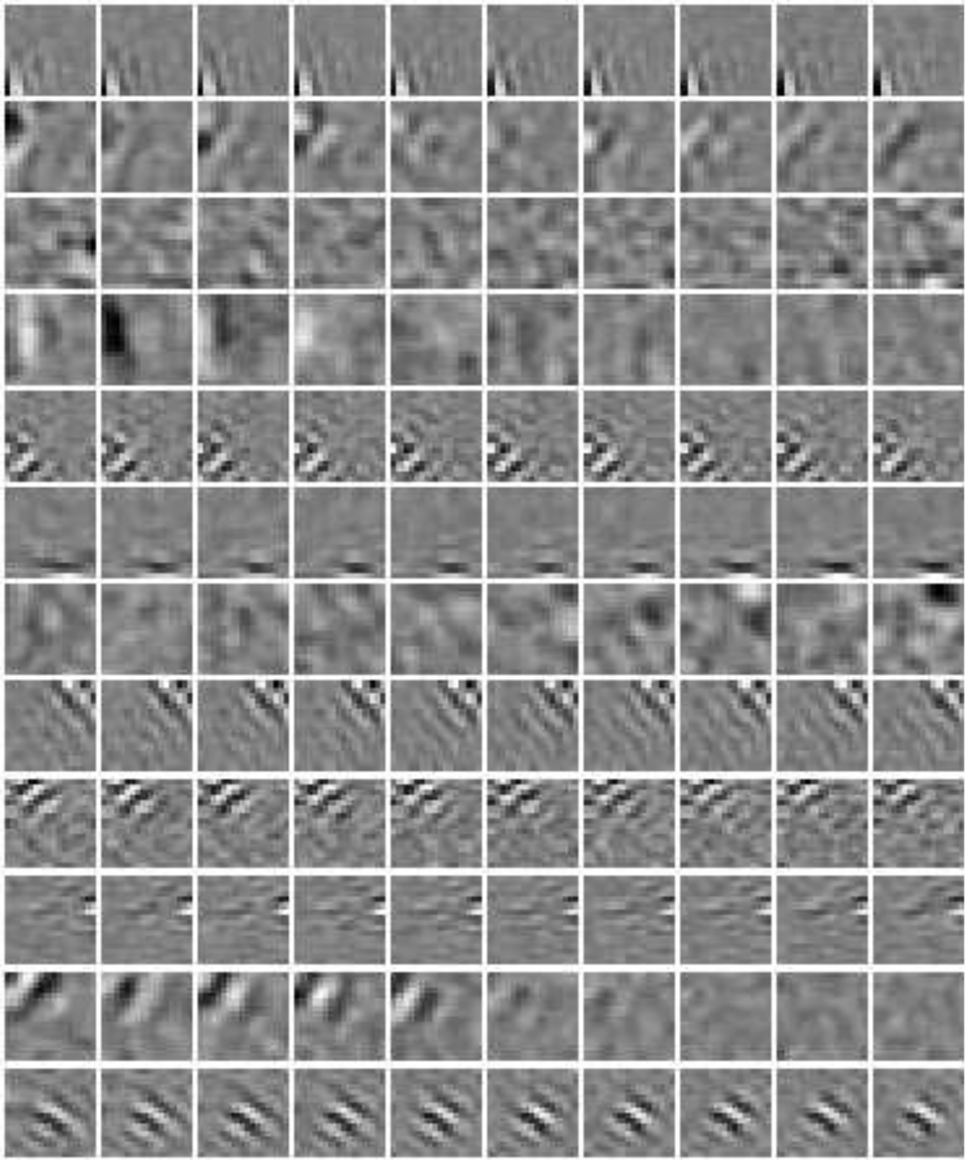


Figure 4.7: Typical filters learned by the convolutional ISA network on 16 × 16 × 10 video blocks. Each row is a filter in 3D (drawn from a row of the matrix W). It can be clearly seen that frames in a row resemble each other closely while their differences are still recognizable.

4.4 Classification

4.4.1 Bag-of-features representation

Our recognition model is based on the bag-of-features (BOF), which is a widely used technique in the literature. In particular, the convolutional ISA network learns hierarchical invariant features from videos on a dense grid in which video blocks overlap 50% in x, y and t dimensions. The learned features are quantized into visual words by k-means clustering; hence a video is then represented as a frequency histogram over visual words. Note that we build the multi-resolution histogram or "pyramid" [30], where the height of pyramid is set to be 3. The pyramid approach will result in a higher-dimensional representation which preserves more information.

4.4.2 Support vector machine

We apply $\mathcal{X}^2$ Support Vector Machine (SVM) [11] to classify human interactions. The $\mathcal{X}^2$ kernel is defined by Eq. (4.14):

$$K(H_i, H_j) = exp(-\frac{1}{2A}\sum_{n=1}^{N_V}\frac{(H_{in} - H_{jn})^2}{H_{in} + H_{jn}}) \tag{4.14}$$

where H_i and H_j are the frequency histograms of the visual words, and N_V is the vocabulary size. A is the mean value of distances between all training samples.

For multi-class classification, we apply the one-against-rest approach and select the class with the highest score. In our experiments, we set the number of visual words N_V to 3000.

4.5 Experimental results

Experimental evaluation of the proposed approach for the UT-Interaction [81] and Hollywood2 dataset [62] is discussed in this section. Our main objective is to evaluate the effectiveness of hierarchical invariant features which are learned from our three-layer convolutional ISA network. We also investigate the behavior of our method with regard to a number of factors including execution time, preprocessing step to extract video blocks, sizes of video blocks at each layer.

Table 4.1: Classification performance for the UT-Interaction dataset.

	Accuracy	Precision	Recall
Set 1	95.6%	86.8%	86.7%
Set 2	96.1%	90.1%	88.3%

4.5.1 Experimental setup

Datasets

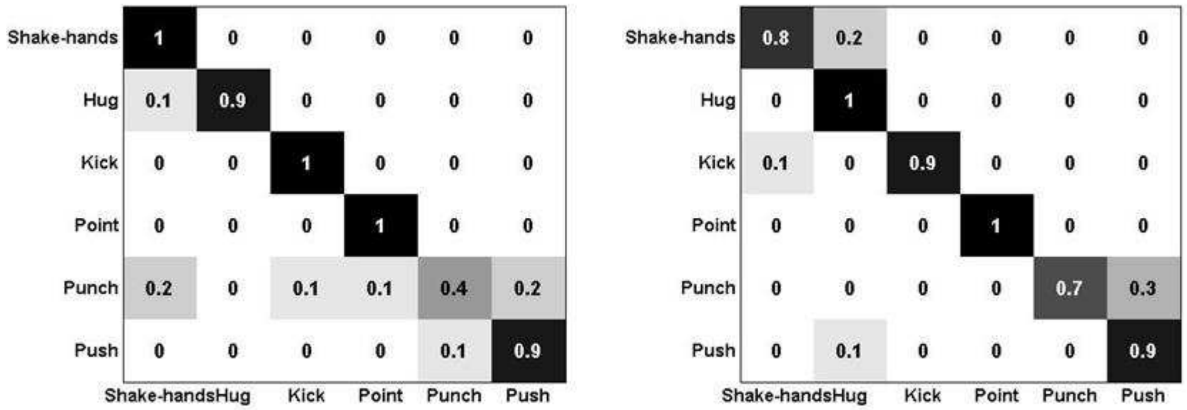
For a better insight of the performance of our method, we employ different datasets (including UT-Interaction and Hollywood2 dataset) in our experiments. We investigate the behavior of our method and perform parameter optimization for the UT-Interaction dataset. Accuracy, precision, and recall are defined in Chapter 3.

Parameter settings

We train the ISA1 on video blocks of size $16 \times 16 \times 10$. The algorithm learns 300 features. The dimension of the subspace is set to 1 ($\kappa = 1$), and the ISA becomes the ICA model. This value of subspace size is acceptable because we aim to learn simple features in the first layer and more complex features from subsequent layers.

The inputs to the ISA2 are of size $20 \times 20 \times 14$. The convolution step is performed in the ISA2 with a convolution stride of 4. At this phase, the ISA2 learns 200 features when the dimension of the subspace is 2. By the same token, the inputs to the ISA3 are of size $20 \times 20 \times 20$. Note that we do not increase the dimensions of video blocks in the third layer because large video blocks are not adequate configuration in order to capture small human movements. This phase learns 100 features with the dimension of the subspace of 2.

We sample 250 video blocks for each training video to form the inputs to each layer. Furthermore, the obtained responses from three layers are fed into the pooling layer to construct local features. The pooling layer keeps 50, 100, 50 features for the first, the second, and the third layer, respectively. In summary, each local feature is represented as a 200-dimensional vector. To speed up the computation, we down-sample original videos to half spatial resolution in all our experiments.



	Shake-hands	Hug	Kick	Point	Punch	Push
Shake-hands	1	0	0	0	0	0
Hug	0.1	0.9	0	0	0	0
Kick	0	0	1	0	0	0
Point	0	0	0	1	0	0
Punch	0.2	0	0.1	0.1	0.4	0.2
Push	0	0	0	0	0.1	0.9

	Shake-hands	Hug	Kick	Point	Punch	Push
Shake-hands	0.8	0.2	0	0	0	0
Hug	0	1	0	0	0	0
Kick	0.1	0	0.9	0	0	0
Point	0	0	0	1	0	0
Punch	0	0	0	0	0.7	0.3
Push	0	0.1	0	0	0	0.9

Figure 4.8: Confusion matrices of per-clip classification results on Set 1 and Set 2 of the UT-Interaction dataset. Horizontal rows of the matrices represent ground truths, and vertical columns are predictions.

4.5.2 Classification results

UT-Interaction dataset

The performance of our method for the UT-Interaction dataset is provided in Table 4.1. The accuracy of our method is 95.6% on Set 1, and 96.1% on Set 2, respectively. In Figure 4.8, we show the confusion matrices of our method on Set 1 and Set 2 of the UT-Interaction dataset.

From those results, we can clearly seen that our method classifies correctly with more than 90% recall for hug, kick, and point interactions. However, it confuses ‘punch’ vs. ‘push’, and ‘shake-hands’ vs. ‘hug’. More specifically, 20% of ‘punch’ interactions on Set 1 and 30% of ‘punch’ interactions on Set 2 are misclassified as ‘push’ interactions because of similar local movements of body part.

Hollywood2 dataset

Table 4.2 shows our classification performance for the Hollywood2 dataset with regard to mean average precision (mAP). The lowest and highest precision are 25.5% for ‘answer phone’ and 79% ‘fight person’, respectively.

Comparison to state-of-the-art

In Table 4.3, we present the comparison between our method and other methods for the UT-Interaction dataset. The first method to compare is the one proposed by Waltisberg et al. [102] which achieved the highest result in the contest at ICPR 2010. It is clearly shown that our method outperforms the method of Waltisberg et al. [102] by 3.4% and 8.3% on Set 1 and Set 2 respectively. On the other hand, although our method and the method by Le et al. [55] are developed from the ISA algorithm, our method achieves higher results than theirs by 6.7% on Set 1 and 8.3% on Set 2. The confusion matrices for the method of Waltisberg et al. [102] and Vahdat et al. [98] are shown in Figure 4.9.

However, our method gets lower result in comparison with the method proposed by Vahdat et al. [98]. This is because their method gets advantages of using a pedestrian detector [16] and a tracker [79] to obtain human positions in frames. However, that fact also comes with some disadvantages. The first disadvantage in their method is that human height affects the result of the pedestrian detector. For example, the pedestrian detector fails to detect people when video scenes only display a part of human body. It means their method is not effective to work on this kind of data. The second disadvantage is that their method could not work when the people, who perform activities, appear in the middle (temporal) of a video. That is because their method depends on a tracker that requires people being tracked has to be in the video from the beginning. Therefore, their method is only applicable to datasets such as UT-Interaction where people appeared from the first frame.

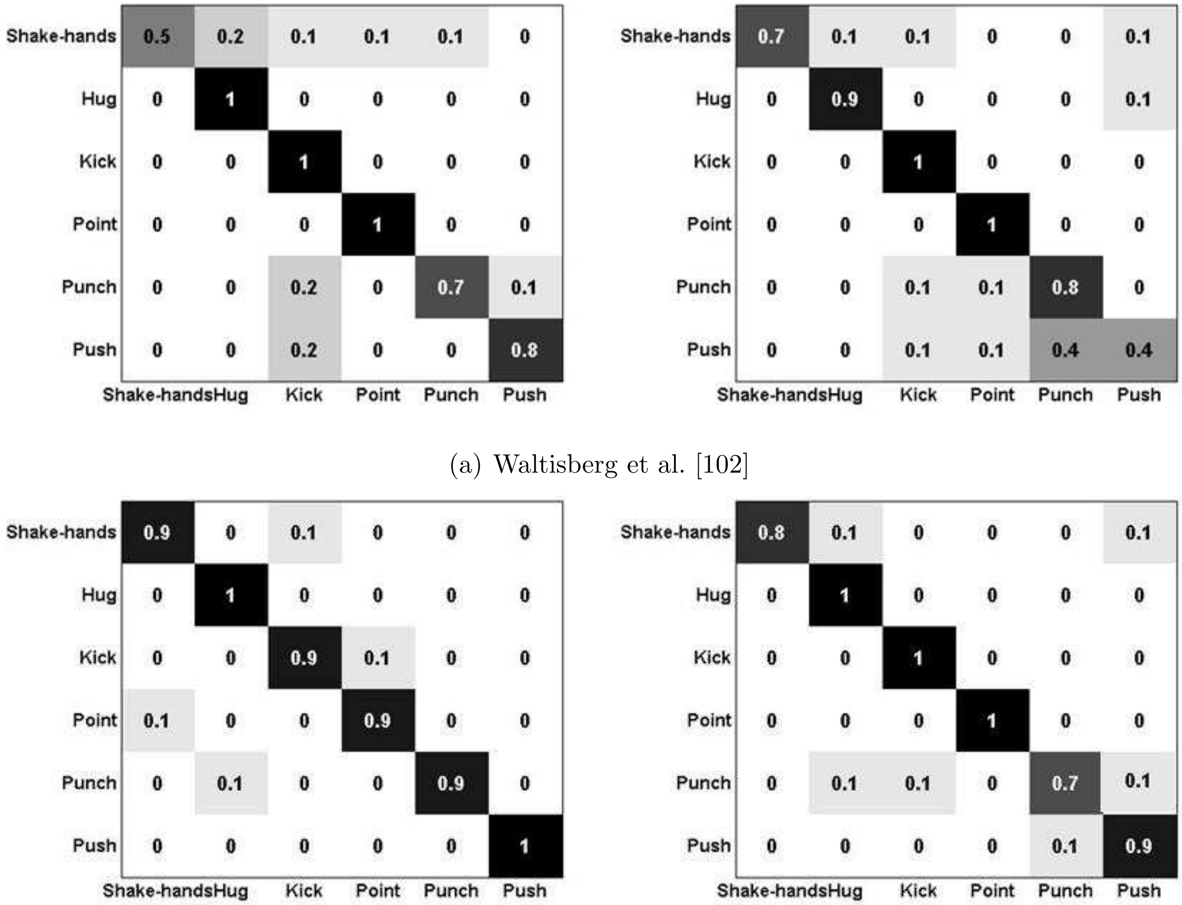
Table 4.2: Classification performance per action class for the Hollywood2 dataset

	mAP
Answer phone	25.5%
Drive a car	87.7%
Eat	61.9%
Fight person	79.0%
Get out car	43.3%
Shake hands	24.1%
Hug person	37.7%
Kiss	59.6%
Run	72.8%
Sit down	59.4%
Sit up	30.9%
Stand up	62.3%
Average	53.7%

Table 4.3: Performance comparison (in terms of recall) for the UT Interaction dataset.

	Set 1	Set 2	Average
Our method	86.7%	88.3%	87.5%
Cuboid + SVM (from [81])	85.0%	70.0%	77.5%
Waltisberg et al. [102]	83.3%	80.0%	81.6%
Le et al. [55]	80.0%	80.0%	80.0%
Yu et al. [111]	-	-	83.3%
Vahdat et al. [98]	93.3%	91.3%	92.3%

For all reasons mentioned above, the method by Vahdat et al. [98] can not work on



	Shake-hands	Hug	Kick	Point	Punch	Push
Shake-hands	0.5	0.2	0.1	0.1	0.1	0
Hug	0	1	0	0	0	0
Kick	0	0	1	0	0	0
Point	0	0	0	1	0	0
Punch	0	0	0.2	0	0.7	0.1
Push	0	0	0.2	0	0	0.8

	Shake-hands	Hug	Kick	Point	Punch	Push
Shake-hands	0.7	0.1	0.1	0	0	0.1
Hug	0	0.9	0	0	0	0.1
Kick	0	0	1	0	0	0
Point	0	0	0	1	0	0
Punch	0	0	0.1	0.1	0.8	0
Push	0	0	0.1	0.1	0.4	0.4

(a) Waltisberg et al. [102]

	Shake-hands	Hug	Kick	Point	Punch	Push
Shake-hands	0.9	0	0.1	0	0	0
Hug	0	1	0	0	0	0
Kick	0	0	0.9	0.1	0	0
Point	0.1	0	0	0.9	0	0
Punch	0	0.1	0	0	0.9	0
Push	0	0	0	0	0	1

	Shake-hands	Hug	Kick	Point	Punch	Push
Shake-hands	0.8	0.1	0	0	0	0.1
Hug	0	1	0	0	0	0
Kick	0	0	1	0	0	0
Point	0	0	0	1	0	0
Punch	0	0.1	0.1	0	0.7	0.1
Push	0	0	0	0	0.1	0.9

(b) Vahdat et al. [98]

Figure 4.9: Confusion matrices for the UT-Interaction dataset of two previous methods (a) Waltisberg et al. [102], (b)Vahdat et al. [98].

Table 4.4: Performance comparison (in terms of mAP) for the Hollywood2 dataset.

Method	mAP
Our method	53.7%
Cuboids [19] + HOG/HOF [52] (from [32])	46.2%
Taylor et al. [94]	46.6%
Le et al. [55]	53.3%
Sun et al. [93]	48.1%
Gaidon et al. [27]	54.4%

Hollywood2 dataset which contains challenging factors such as changes in people height, scene changing. In contrast to their method, our method achieves mAP of 53.7% on Hollywood2 dataset. It is important to emphasize that we use the same parameter settings for UT-Interaction and Hollywood2 dataset. This fact demonstrates that our method is applicable to different types of datasets. In addition, unlike other methods which require labelled data to learn features, our three layered convolutional ISA network is the unsupervised feature learning method because it is able to learn features directly from unlabeled data. Furthermore, the features learned from the network are proven to be phase invariant, which are good features to represent images and videos.

Table 4.4 compares the mean average precision (mAP) between our method and other methods for the Hollywood2 dataset. Achieving the mAP of 53.7%, our approach performs better than other methods [19, 55, 93, 94]. Compared with the method proposed by [32] that is based on the hand-designed features including cuboid detector and HOG/HOF descriptor, our method outperforms by 7.1%. Although the method of Sun et al. [93], the method proposed by Le et al. [55], and our method are based on unsupervised feature learning, the result of our method is higher than theirs. However, in comparison with the method by Gaidon et al. [27], our method is 0.7% lower than their result. This can be explained by the observation that Gaidon et al. [27] organize activities into unordered binary trees of local trajectories, which enable their method to capture complex activities effectively.

In summary, there is no single method which achieves higher results for the UT-Interaction and Hollywood2 dataset than our method. It means that our method achieves balanced results for different types of datasets.

Discussion of independent subspace analysis (ISA) and slow feature analysis (SFA)

Independent subspace analysis is a generalization of independent component analysis. The advantage of the ISA is that it can learn features which are robust to local translation while being selective to orientation and frequency. Slow feature analysis [108] is also unsupervised feature method which learns invariant or slowly varying features from input signals. Blaschke et al. [7] presented an analytical comparison between SFA and ICA. One of their interesting findings is that the SFA and ICA are equivalent in some cases. It may open another issue if we analyze the relation between ISA and SFA. Therefore, in this dissertation, we compare independent subspace analysis with slow feature analysis based on experimental results on the Hollywood2 dataset. In particular, we show the comparison between our method and the method of Sun et al. [93] which is developed by combining slow feature analysis with deep learning techniques.

Our method achieves mAP of 53.7% on the Hollywood2 dataset while the method of Sun et al. [93] achieves 48.1%. It means that our method outperforms the method of Sun et al. [93] by 5.6%. However, in my opinion, this evidence may not be sufficient to conclude which feature is more effective for activity recognition.

4.5.3 Analysis of parameter settings

We investigate the behavior of our method with regard to a number of factors including video block extraction, sizes of video blocks at each layer, and execution time. This analysis is based on the result of our experiments for the UT-Interaction dataset.

Influence of frame differencing in video block extraction

To investigate the influence of frame differencing in the video block extraction step on recognition results, we compared the performance of our method with and without frame differencing, which is shown in Table 4.5. By applying frame differencing in the video block extraction step, the performance of our method is significantly improved. In particular, the recall is improved as much as 5% for Set 1 and 3.3% for Set 2. This is because our devised video block extraction (with frame differencing) extracts video blocks containing foreground information which play an important role in disambiguating similar human interactions.

Table 4.5: Performance comparison with and without applying frame differencing in video block extraction

	Accuracy		Recall	
	Set 1	Set 2	Set 1	Set 2
Video block extraction (with frame differencing)	95.6%	96.1%	86.7%	88.3%
Video block extraction (without frame differencing)	93.9%	95.0%	81.7%	85.0%

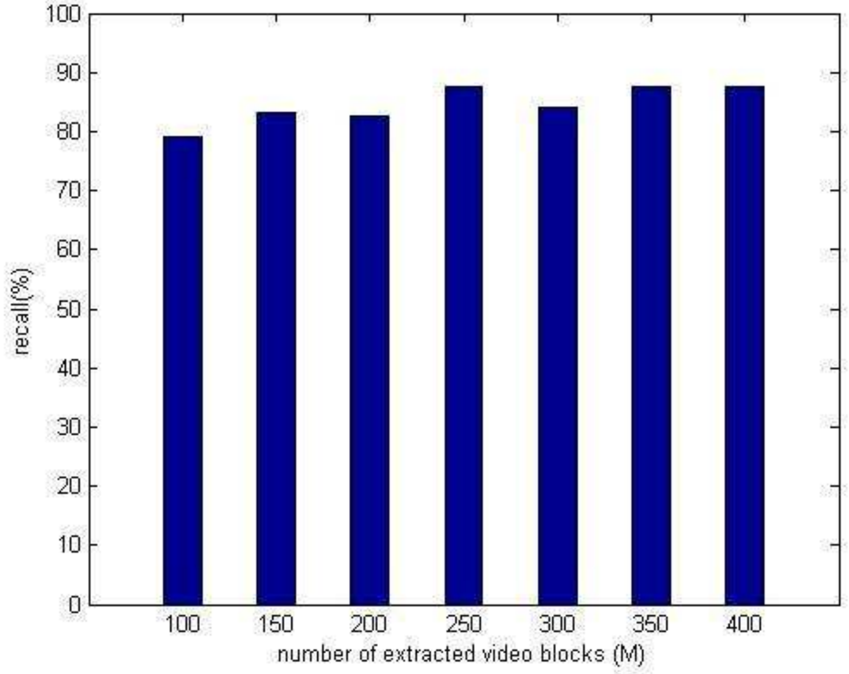


Figure 4.10: Recognition results on the UT-Interaction dataset with changing the number of extracted video blocks.

Influence of number of extracted video blocks

Figure 4.10 shows recognition results in case of changing the number of extracted video blocks. We can see that our method achieves highest recall with these values of M: 250, 350, 400. Therefore, in the experiments, the number of extracted video blocks is set to 250.

Influence of block size parameters

Table 4.6 shows the performance of our method for different choices for the sizes of video blocks at three layers. From the result, we can see that our method performs best when the block sizes are either 12 × 12 × 8 - 16 × 16 × 12 - 20 × 20 × 16 or 16 × 16 × 10 - 20 × 20 × 14 - 20 × 20 × 14 at three layers, respectively, where both accuracy and recall metrics are maximum. Furthermore, larger or smaller video block sizes deteriorates the recall significantly. This result implies that too large or too small video block sizes are not effective to capture human interactions in videos.

Execution time

The parameters for our three-layer convolutional ISA network are set as discussed above. Set 1 of the UT-Interaction dataset has 54 videos for training and 6 videos for testing. For each training video, we extract 250 video blocks. Hence each layer of the three-layer convolutional ISA network is trained on 13500 video blocks. Total time for learning weights of all three layers is 520 seconds. This result is obtained by running our method on the computer with following configuration: Intel(R) Core(TM) i7-2600 CPU @ 3.40 GHz 3.70 GHz, and 8 GB RAM.

We perform feature extraction with dense sampling (50% overlapping in x, y, and t dimensions) on videos in Set 1 with spatial resolution of 173 × 128 (half resolution compared to original resolution to fasten execution time). The total number of frames is 7160. The feature extraction velocity is 47 frames/second. The total execution time

Table 4.6: Recognition results on the UT-Interaction dataset in different choices of block size

Block size	Accuracy	Recall
8 × 8 × 8 12 × 12 × 12 16 × 16 × 16	93.9%	81.7%
12 × 12 × 8 16 × 16 × 12 20 × 20 × 16	95.8%	87.5%
16 × 16 × 10 20 × 20 × 14 20 × 20 × 14	95.8%	87.5%
16 × 16 × 10 20 × 20 × 14 24 × 24 × 14	94.4%	83.3%

including training and testing in bag-of-features model to classify 6 test videos is 523 seconds, which is approximately 8.7 minutes. Generally, our method is fast because it requires only matrix vector product and convolution operations.

4.6 Summary of interaction recognition

In this chapter, we have introduced a three-layer convolutional network which uses the Independent Subspace Analysis (ISA) algorithm to learn hierarchical invariant features from videos. Using the invariant features learned by the ISA, we build a bag-of-features (BOF) model to recognize human interactions. We also evaluate the performance of our approach and the effectiveness of hierarchical invariant features on video sequences of the UT-Interaction and Hollywood2 dataset. Experimental results show that our three-layer convolutional ISA network is able to learn features which are effective to represent complex activities such as human interactions in realistic environments.

Chapter 5

Interaction Temporal Localization based on Sliding Window Approach

5.1 Introduction

Activity localization is an important research topic with a wide range of applications such as video surveillance, automatic understanding of videos, search and annotation applications. It is much more demanding, and is generally a more difficult task compared to activity classification. That is because activity localization requires the activity class to be correctly classified and also its spatial-temporal extents to be identified. Hence, activity classification can be considered as a sub-problem which is required to solve in activity localization. There are two types of localization task: temporal localization [22, 26] and spatio-temporal localization [38, 45, 50, 91, 95, 97, 100, 104].

Fewer research efforts have been made on activity localization compared to activity classification. One of the most straightforward way is to treat the localization task as localized classification. This technique, known as sliding window, slides either a temporal or spatio-temporal window over the entire video, and selects the detection window with the maximum classifier score. If the detection window overlaps with ground truth extents more than a certain percentage and it's label is correctly recognized, it is considered as a correct detection. For example, for temporal localization, Gaidon et al. [26] introduced a Actom Sequence Model (ASM) which represents the temporal structure of activities as a sequence of histograms of actom-anchored visual features. Actoms can be referred to as atomic activity units, whose durations are learned in a non-parametric way.

Several researchers [50, 67, 95] developed structured models for localization inspired by a deformable part model (DPM) [25] in object detection. The DPM is a latent-variable model, which is composed of a series of detectors: a root filter for the entire object and many part filters covering smaller parts of the object. This model extracted histogram of oriented gradients (HOG) features and applied a latent support vector machine. The detectors are combined into a scoring function by considering the maximum individual scores and penalizing the displacement of the parts from an initial configuration. The model was extended to both temporal [67] and spatio-temporal activity localiztion [50,95]. For example, Tian et al. [95] extracted HOG3D features [44] instead of HOG features, and their model also consists of a root filter and many part models. The spatio-temporal sliding window also is the applied, and the detection window with highest score is chosen as the location of the activity.

Gemert et al. [100] introduced unsupervised spatio-temporal proposals which are directly generated from dense trajectories [103] to represent videos for classification and localization. The proposals reduce the video search space to a small set of spatio-temporal tubes, which are likely to contain an activity. Therefore, their method is faster compared to sliding window approaches.

Ma et al. [60] also proposed a new representation, hierarchical space-time segments, for activity localization. This approach has two level hierarchy: first level consists of root space-time segments which may contain a human body, and second level comprises parts of the root. This approach uses an unsupervised method to extract static and non-static segments, and also captures their hierarchical relationships. Thus, the approach yields good classification and localization results comparable to state-of-the-art methods.

Previous localization methods have evaluated and shown their effectiveness on the UCF Sports [78]. The UCF Sports is designed for classification and localization of actions, and in general there is one primary action class shown in each video. Some videos in the UCF Sports may include one or more instances from other action classes. However, the UT-Interaction dataset is recorded for localization of human-human interactions, and contains 20 continuous video sequences. Each video sequence consists of many executions of interactions, which are performed sequentially and/or concurrently. In addition, people enter and exit video scenes at any time of video sequences. That is the reason performing localization on the UT-Interaction dataset is more complicated compared to other datasets, such as UCF-Sports. To the best of my knowledge, most of recent methods have only concentrated on classification, and no attempt has been made for localization of human interactions.

This chapter addresses the problem of *temporal localization* of *human interactions*, i.e., finding if and when an interaction is performed in a database of continuous video

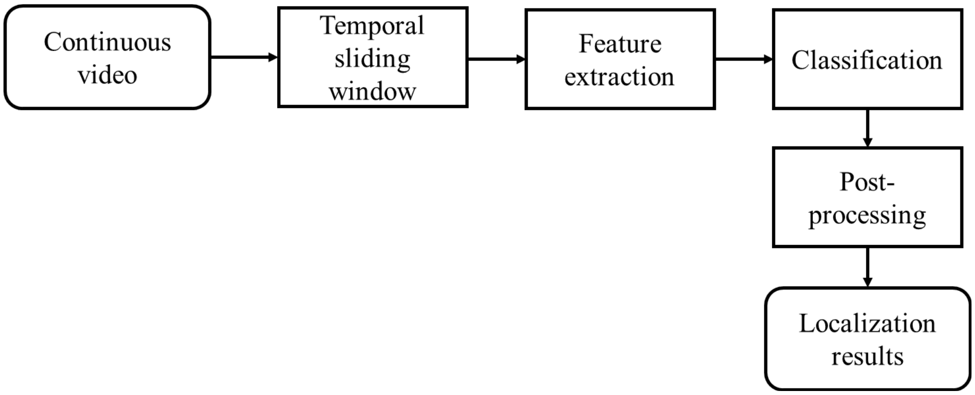


Figure 5.1: The procedure of our localization method on continuous video sequences.

sequences. As discussed in the previous chapter, the proposed three-layer ISA convolutional network has been proven that its extracted hierarchical invariant features can represent complex interactions in realistic video data. Therefore, in localization task, we also learn hierarchical invariant features and employ a sliding window technique over the video sequence, and select the temporal detection window with maximum classifier score. Figure 5.1 shows the procedure of our proposed localization method.

Our main contribution is to introduce an approach for localization of human interactions in realistic video data based on hierarchical invariant features. Although applying the sliding window technique is not a new idea, it is the first work which reports localization results on the UT-Interaction dataset.

Outline

Firstly, in Section 5.2, we present an overview of our temporal localization based on the sliding window technique. Section 5.2.1 describes the temporal sliding window in more detail, and Section 5.2.2 presents the implementation technique for extracting hierarchical invariant features. Secondly, classification and post-processing method are presented in Section 5.2.3 and in Section 5.2.4, respectively. Finally, we present parameter settings and experimental results on the continuous sequences of the UT-Interaction dataset in Section 5.3.2, and conclude our method in Section 5.4.

Table 5.1: Duration (in frames) per interaction class of Set 1 of the UT-Interaction dataset.

	Min	Max	Average
Shake-hands	90	154	112
Hug	103	143	126
Kick	48	125	75
Point	52	117	93
Punch	30	125	72
Push	58	201	103

Table 5.2: Duration (in frames) per interaction class of Set 2 of the UT-Interaction dataset.

	Min	Max	Average
Shake-hands	65	118	95
Hug	90	125	107
Kick	44	95	60
Point	50	121	77
Punch	22	68	54
Push	53	105	70

5.2 Interaction localization based on temporal sliding window

5.2.1 Temporal sliding window

Table 5.1 and 5.2 show minimum, maximum and average duration (in frames) for each interaction class of Set 1 and Set 2 of the UT-Interaction dataset. It is clear to see that minimum duration of interaction execution is 22 frames ('point' interaction) and maximum duration is 201 frames ('push' interaction). Therefore, we use the temporal detection windows with varied durations as 14, 28, 42, 56, 70, 84, 98, 112, 126, 140, and 154 frames.

5.2.2 Extraction of hierarchical invariant features

To learn hierarchical invariant features, we sample 250 video blocks of size 16 × 16 × 10 from ground-truth video segments, and train the ISA1 with the dimension of the subspace of 1. Similarly, video blocks of size 20 × 20 × 14 are inputs to the ISA2. The convolution step is performed in the ISA2 with a convolution stride of 4. In short, we learn 300 features for the ISA1, and 200 features for the ISA2 with the dimension of the subspsce of 2. For the third layer of the three-layer convolutional ISA network, we sample video blocks of 20 × 20 × 14, which are the same size as in the ISA2. Note that we do not increase the dimension of video blocks in the third layer because large video blocks are not effective to capture small movements of body parts. This phase learns 100 features with the dimension of the subspace of 2.

As indicated previously, the convolutional ISA network learns simple features in the first layer, and more complex features in subsequent layers. Therefore, we try to reduce the contribution of features learned from the first layer by mean spatial temporal pooling and PCA. The features learned from the second layer are also processed by max pooling operation. Finally, after the pooling stage, we obtain 50 features for the first layer, and 100 features for the second layer, and 50 features for the third layer.

For each temporal detection window, our convolutional ISA network extracts hierarchical invariant features on a dense grid in which video blocks overlap 50% in x, y and t dimensions. The extracted features are quantized into visual words by k-means clustering, and each temporal detection window is then represented as the frequency histogram over the visual words. In our experiments, we set the number of visual words to 3000.

5.2.3 Classification

In this section, we consider temporal localization in continuous video sequences as a large-scale classification problem. It means that we apply a classifier at multiple temporal locations throughout the continuous video sequence, and select the location with maximum classifier score.

Positive and negative training examples

For each interaction class, we train a χ^2 SVM classifier on the BOF representation of temporal windows. We use time intervals given by ground-truth annotations as positive training examples. As negative training examples, we use:

(i) All segmented videos from other interaction classes.

(ii) Example windows which are randomly sampled around ground-truth intervals, and have an overlap between 20% and 30% with a positive example.

(iii) Example windows which are randomly sampled from continuous video sequences. Note that the overlap of two negative training examples is less than 60% to avoid redundancy.

In continuous video sequences, the number of executions of interactions is rare and most of the scenes are irrelevant actions or just background scenes. Therefore, more negative training examples than positive training examples are necessary. However, when training a SVM classifier for interaction localization, we often have a very large number of negative examples. Therefore, it is not feasible to take all negative examples into account simultaneously. Hence, we will construct training data which consists of positive training examples and hard negative ones by applying a hard negative mining algorithm.

The hard negative mining algorithm is a simple technique which allows to extract a small set of key negative examples. First, we train a classifier model with an initial subset of negative training examples, which are randomly sampled from continuous video sequences. Then, we slide temporal detection windows, and collect example windows which are incorrectly classified (false positives) to form a set of hard negatives. The hard negative examples are added to the training set, and we re-train our classifier model. This process repeats a few times.

Testing

For testing, we slide temporal detection windows over a continuous video sequence with a stride of 10 frames. As previously mentioned, we use the temporal detection windows with a duration of 14, 28, 42, 56, 70, 84, 98, 112, 126, 140, and 154 frames. After scoring the temporal detection windows, a non-maximum suppression is applied to enforce that non of the retained windows are overlapping.

5.2.4 Non-maximum suppression

Sliding window approach typically results in multiple overlapping detection windows for each execution of an interaction. To eliminate the overlapping detection windows, we apply a greedy procedure which is called non-maximum suppression.

In detail, after applying the sliding window method, we have a set of temporal detection windows for a particular execution in video sequence. Each temporal detection window is defined by a time interval with a score. We sort the temporal detection windows by score, and find the highest scoring windows and remove all neighboring positive windows with an overlap greater than 0.3.

Table 5.3: Localization results on the Set 1 of the UT-Interaction dataset.

	Recall	Precision
Shake-hands	66.7%	72.7%
Hug	66.7%	61.5%
Kick	58.3%	87.5%
Point	53.3%	42.1%
Punch	25.0%	60.0%
Push	53.9%	70.0%
Average	53.9%	65.6%

5.3 Experimental results

In this section, we present experimental evaluation of our localization method on the UT-Interaction dataset [81]. Our objective is to evaluate the effectiveness of the sliding window technique, and hierarchical invariant features which are learned from our three-layer convolutional ISA network. We also investigate the influence of some important components of our method.

5.3.1 Experimental setup

We followed 10-fold leave-one-out cross validation per set to evaluate our localization method. Note that in this chapter, we use continuous video sequences instead of video segments as in classification. If the detection window overlaps with the provided ground truth time intervals more than 50%, it is considered as a true positive. Otherwise, it is considered as false positive.

5.3.2 Localization results

Table 5.3 shows precision and recall of our localization method on the Set 1 of the UT-Interaction dataset. Our method obtains a performance of 65.6% precision and 53.9% recall. Specifically, our localization method localizes correctly more 60% precision of shake-hands, hug, kick, punch and push. Point interactions get lowest precision (42.1%) because there are a lot of similar point interactions which occur in the continuous video sequence.

Precision and recall of our localization method on the Set 2 of the UT-Interaction is presented in Table 5.4. Generally, our method achieves 60.5% in terms of precision,

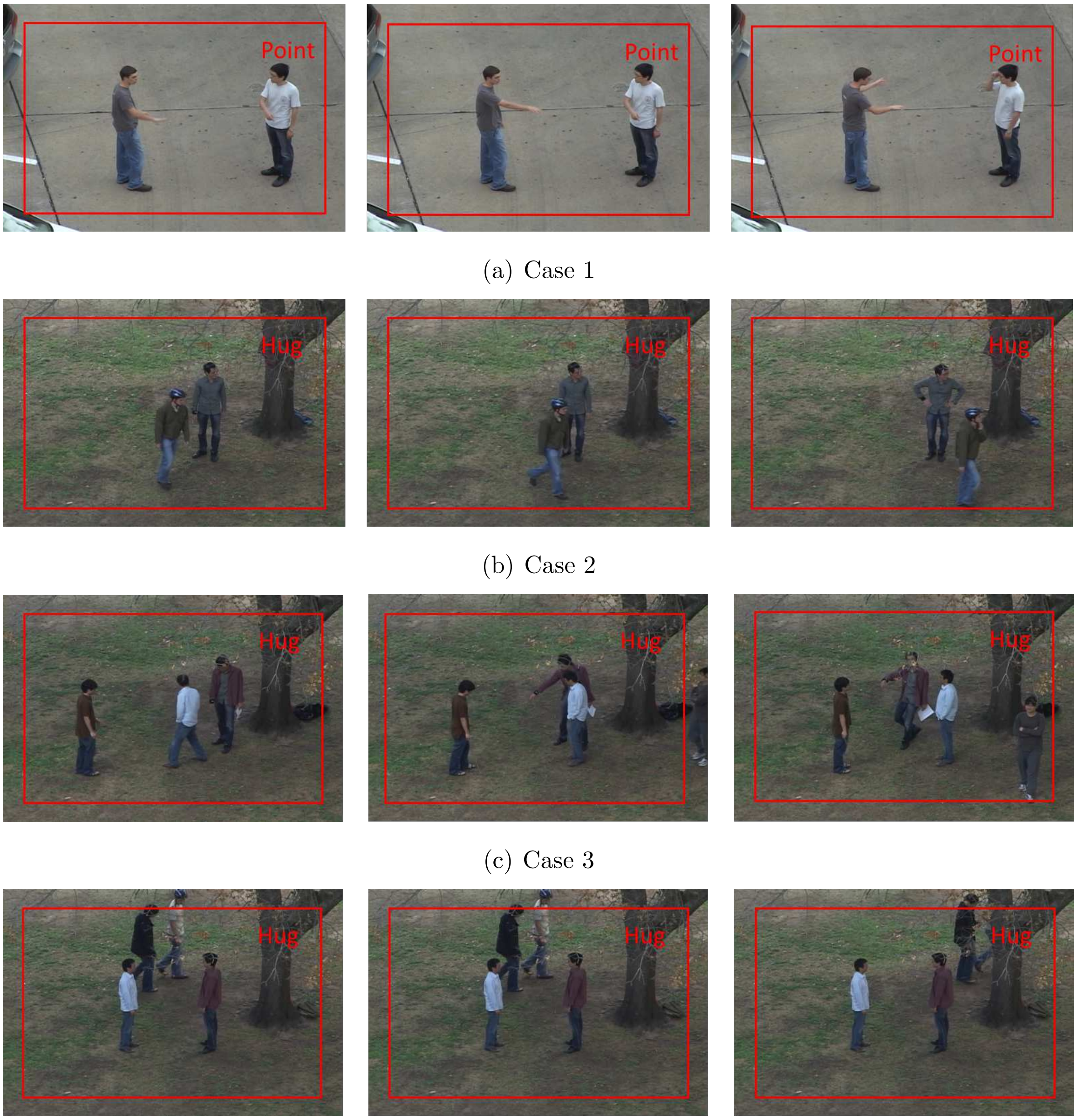
Table 5.4: Localization results on the Set 2 of the UT-Interaction dataset

	Recall	Precision
Shake-hands	50.0%	72.7%
Hug	63.6%	28.0%
Kick	58.3%	100.0%
Point	45.0%	64.3%
Punch	36.4%	44.4%
Push	50.0%	53.3%
Average	50.6%	60.5%

and 50.6% in terms of recall. In Set 2, we obtain more than 60% precision of shake-hands, kick, and point. However, the precision of hug interactions is rather low in Set 2. It may be because there are two pairs of interacting subjects, and our localization method misclassifies such scenes as hug interactions. These results are insufficient for real applications; however they encourage for further research on localization of human interactions. Figure 5.2 shows several wrong localization cases in the UT-Interaction dataset.

5.4 Summary of interaction temporal localization

In this chapter, we have introduced an approach for solving temporal localization on the UT-Interaction dataset, which is one of the hardest datasets for localization. To the best of our knowledge, this is the first work that reports temporal localization results on the UT-Interaction dataset. As discussed in previous chapter, the three-layer convolutional ISA network is able to learn hierarchical invariant features from videos. Our classification method outperforms several state-of-the-art method. We employ a sliding window technique to localize human interactions. Temporal detection windows with different durations are used, and we slide these windows over the continuous video sequence with a stride of 10 frames. A non-maximum suppression is applied to enforce that non of the retained windows could be overlapping. However, experimental results on the UT-Interaction dataset are insufficient for real applications. It means that in the future, it is required to put more effort into localization task, which is more difficult and significant compared to classification task.



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 5.2: Several wrong localization cases of our temporal localization method for the UT-Interaction dataset.

Chapter 6

Conclusion and Perspective

Our work investigated *hierarchical invariant features* in order to represent human interactions in realistic video data. Besides, we also introduced the *temporal localization method based on sliding window technique* to localize human interactions. Our experiments show that classification performance on challenging video sources, such as the UT-Interaction, the Hollywood2 are comparable to state-of-the-art methods. Furthermore, we believe that our localization method is the first work which reports experimental results on the continuous video sequences of the UT-Interaction dataset.

In the following, we summarize our key contributions and the main observation obtained from our experiments. We also point out some potential research directions what we deem interesting for future work on interaction recognition in real-world videos

6.1 Conclusion

Interaction recognition using Hierarchical Invariant Features

Our first contribution is a *unsupervised feature learning method* for representation of activity videos in real-world settings. We proposed a *three-layer convolutional ISA network* which is able to learn hierarchical invariant features automatically. In particular, we introduced a *procedure of video block extraction*, which allows to extract foreground video blocks. Compared with the original ISA algorithm, our convolutional ISA network is trained on video blocks instead of image patches. The obtained features are then convolved with larger video blocks. The outputs of this convolution step are fed into the next layer, which is implemented by another ISA algorithm. This structure enables the three-layer convolutional ISA network to learn hierarchical invariant features. Furthermore, we proposed a *pooling layer* to reduce the contributions of simple features in lower layers while still achieving translation invariance. Our experiments highlighted that hierarchical invariant features are effective to represent video sequences of interactions and actions recorded in realistic settings.

Interaction Temporal Localization based on Sliding Window Approach

Recent methods have concentrated on activity classification problem, and fewer research efforts have been made on activity localization. The UT-Interaction is considered as one of the hardest datasets for localization because each video sequence consists of many executions of interactions, which are performed sequentially and/or concurrently. Besides, people enter and exit video scenes at any time of video sequences. We proposed our temporal localization method based on a sliding window technique and hierarchical invariant features. Specifically, we apply a classifier at multiple temporal locations throughout the continuous video sequence, and select the location based on the maximum classifier score. For each temporal detection window, we extract hierarchical invariant features using the three-layer convolutional ISA network. Finally, the post-processing method is applied to enforce that non of the detection windows are overlapping. Our method obtains a performance of 65.6% mAP on Set 1, and 60.5% mAP on Set 2. It means that in the future, it is required to put more effort into localization to improve localization performance. Besides, in this thesis, we only developed temporal localization method, thus spatio-temporal localization of human interactions also attracts our interest in the future.

6.2 Future work

In this section, we give some possible extensions of our work, which are suggested by our experiments and the recent progress of the computer vision.

Temporal representation for human interactions

Our recognition method is based on the popular bag-of-features (BOF), which suffers from a severe limitation. The BOF ignores the temporal ordering of video frames, which is an important cue in activity recognition. Therefore, capturing temporal ordering of video frames is important to improve the classification and localization performance.

Incorporate motion features

Motion information plays an important role for representation of activities. Thus combining several motion features with our hierarchical invariant features may achieve better results for recognition of human interactions. Several features, such as dense trajectories [103], histogram of optical flows (HOF), motion boundary histogram (MBH) are considered to capture motion information.

Spatio-temporal localization

Our current localization focuses on localizing human interactions temporally. However, since the continuous video sequences of the UT-Interaction dataset consist of many execution of interactions which are performed sequentially and/or concurrently, performing temporal localization is not enough. In the future, we will put more effort into localizing interactions temporally and spatially.

Publications

[1] N. Nguyen and A. Yoshitaka, “Soccer Video Summarization based on Cinematography and Motion Analysis,” Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP), pp. 1-6, 2014.

[2] N. Nguyen and A. Yoshitaka, “Human Interaction Recognition using Independence Subspace Analysis Algorithm,” Proc. IEEE International Symposium on Multimedia (ISM), pp. 40-46, 2014.

[3] N. Nguyen and A. Yoshitaka, “Human Interaction Recognition using Hierarchical Invariant Features,” International Journal of Semantic Computing (IJSC), pp. 169-191, 2015.

[4] N. Nguyen and A. Yoshitaka, “Classification and Temporal Localization of Human-Human Interactions,” Proc. IEEE International Conference on Multimedia Big Data (BigMM), 2016 (presented).

Acknowledgement

First of all, I would like to take this opportunity to express my deep gratitude to my supervisor, Associate Professor Atsuo Yoshitaka for his invaluable guidance. His experience, vision and scientific intuition have been inspired me during these years. He has always encouraged me to consistently develop both technical and scientific skills. Furthermore, he has always been very supportive and patient.

I would also like to give special thanks to Associate Professor Hideaki Kanai, the supervisor of my minor research. His openness and enthusiasm have allowed me to push my boundaries and develop my own ideas.

Special thanks go also to the second supervisor, Associate Professor Kazunori Kotani, and all the committee members for agreeing to evaluate my research and asking me many insightful questions throughout our interactions. My gratitude is furthermore sent to all members of Yoshitaka and Kotani laboratory for their kind support over the past years in both academic and private life. I also pay my acknowledgment to Jaist Doctoral Research Fellow (DRF), which gives me an opportunity to pursue a PhD course.

I also can not express how grateful I am for all support I continuously received from my family, and from all my friends. Finally, the biggest thanks go to my husband, An, for his support, help, and understanding. The love and joys he brings me every day makes life taste better, and anything seems possible with him beside me.

Bibliography

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

[2] A. Ahad, T. Ogata, J. Tan, H. Kim, and S. Ishikawa. Motion recognition approach to solve overwriting in complex actions. In *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition,*, pages 1–6, 2008.

[3] M. Amer and S. Todorovic. A chains model for localizing participants of group activities in videos. In *Proc. IEEE. Int. Conf. Computer Vision (ICCV)*, pages 786–793, 2011.

[4] S. Avila, N. Thome, M. Cord, E. Valle, and D. Araujo. Bossa: Extended bow formalism for image classification. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 2909–2912, 2011.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. European Conference on Computer vision (ECCV)*, pages 404–417. 2006.

[6] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. IEEE. International Conference on Computer Vision (ICCV)*, pages 1395–1402, 2005.

[7] T. Blaschke, P. Berkes, and L. Wiskott. What is the relation between slow feature analysis and independent component analysis? *Neural computation*, pages 2495–2508, 2006.

[8] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 257–267, 2001.

[9] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *Proc. IEEE. Int. Conf. Computer Vision (ICCV)*, pages 778–785, 2011.

[10] A. Castrodad and G. Sapiro. Sparse modeling of human actions from motion imagery. *International Journal of Computer Vision*, pages 1–15, 2012.

[11] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, pages 1–27, 2011.

[12] J. M. Chaquet, E. J. Carmona, and A. Fernndez-Caballero. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633–659, 2013.

[13] C. C. Chen and J. K. Aggarwal. Recognizing human action from a far field of view. In *Workshop on Motion and Video Computing (WMVC)*, pages 1–7, 2009.

[14] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision (ECCV)*, pages 1–2, 2004.

[15] O. Cula and K. Dana. Compact representation of bidirectional texture functions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages I–1041, 2001.

[16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[17] S. Danafar and N. Gheissari. Action recognition for surveillance applications using optic flow and svm. In *Assian Conference on Computer Vision (ACCV)*, pages 457–466, 2007.

[18] K. Derpanis, M. Sizintsev, K. Cannons, and R. Wildes. Action spotting and recognition based on a spatiotemporal orientation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 527–540, 2013.

[19] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. IEEE. Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VSPET)*, pages 65–72, 2005.

[20] D. Donchev. 27 mind blowing youtube facts, figures and statistics 2016. http://fortunelords.com/27-mind-blowing-youtube-facts-figures-and-statistics-backed-by-data/, 2016.

[21] Z. Dong, Y. Kong, C. Liu, H. Li, and Y. Jia. Recognizing human interaction by multiple features. In *Proc. Asian Conf. Pattern Recognition (ACPR)*, pages 77–81, 2011.

[22] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1491–1498, 2009.

[23] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 726–733, 2003.

[24] G. Farneback. Two-frame motion estimation based on polynomial expansion. In *Proc. Scandinavian Conf. on Image analysis*, pages 363–370. 2003.

[25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2010.

[26] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3201–3208, 2011.

[27] A. Gaidon, Z. Harchaoui, and C. Schmid. Activity representation with motion hierarchies. *International Journal of Computer Vision (IJCV)*, pages 219–238, 2014.

[28] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. A ”string of feature graphs” model for recognition of complex activities in natural videos. In *Proc. IEEE. Int. Conf. Computer Vision (ICCV)*, pages 2595–2602, 2011.

[29] D. Gavrila. The visual analysis of human movement. *Computer Vision and Image Understanding*, pages 82–98, 1999.

[30] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1458–1465, 2005.

[31] G. Guangchun Cheng, Y. Wan, A. Saudagar, K. Namuduri, and B. Buckles. Advances in human action recognition: A survey. *arXiv preprint arXiv:1501.05964*, 2015.

[32] H. H. Wang, M. Ullad, K. A., I. Laptev, and C. Schimid. Evaluation of local spatio-temporal features for action recognition. In *Proc. of the British Machine Vision Conference (BMVC)*, pages 1–11, 2010.

[33] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, page 50, 1988.

[34] A. Hyvarinen, J. Hurri, and P. Hoyer. *Natural Image Statistics*. Springer, 2009.

[35] A. Hyvarinen and U. Koster. Fastisa: A fast fixed-point algorithm for independent subspace analysis. In *EsANN*, pages 371–376, 2006.

[36] A. Hyvarinen and U. Koster. Complex cell pooling and the statistics of natural images. *Network: Computation in Neural Systems*, pages 81–100, 2007.

[37] A. Hyvarinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, pages 411 – 430, 2000.

[38] M. Jain, J. Gemert, H. Jegou, P. Bouthemy, and C. Snoek. Action localization with tubelets from motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 740–747, 2014.

[39] N. Jojic, A. Perina, and V. Murino. Structural epitome: a way to summarize ones visual experience. In *Advances in neural information processing systems*, pages 1027–1035, 2010.

[40] I. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *Proc. European Conference on Computer Vision (ECCV)*, pages 293–306. Springer Berlin Heidelberg, 2008.

[41] K. K. Schindler and L. Gool. Action snippets: How many frames does human action recognition require? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[42] H. Kiani, T. Sim, and S. Lucey. Multi-channel correlation filters for human action recognition. In *Proc. IEEE. Int. Conf. on Image Processing (ICIP)*, pages 1485–1489, 2014.

[43] W. Kim, J. Lee, M. Kim, D. Oh, and C. Kim. Human action recognition using ordinary measure of accumulated motion. *EURASIP Journal of Advanced in Signal Processing*, pages 1–11, 2010.

[44] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference (BMVC)*, pages 1–10, 2008.

[45] A. Klaser, M. Marszalek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *Proc. European Conf. on Trends and Topics in Computer Vision (ECCV)*, pages 219–233, 2012.

[46] Y. Kong and Y. Jia. A hierarchical model for human interaction recognition. In *Proc. IEEE. Int. Conf. Multimedia and Expo (ICME)*, pages 1–6, 2012.

[47] Y. Kong, Y. Jia, and Y. Fu. Learning human interaction by interactive phrases. In *Proc. European Conference on Computer Vision (ECCV)*, pages 300–313, 2012.

[48] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2046–2053, 2010.

[49] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Proc. Int. Conference on Computer Vision (ICCV)*, pages 2556–2563, 2011.

[50] T. Lan, Y. Wang, and G. Mori. Discriminative figure-centric models for joint action localization and recognition. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2003–2010, 2011.

[51] I. Laptev. Modeling and visual recognition of human actions and interactions. Master's thesis, 2013.

[52] I. Laptev and T. Lindeberg. Space-time interest points. In *Proc. IEEE. Int. Conf. Computer Vision (ICCV'03)*, pages 432–439, 2003.

[53] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[54] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006.

[55] 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 *Proc. IEEE. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3361–3368, 2011.

[56] T. Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, pages 79–116, 1998.

[57] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *Proc. IEEE. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1996–2003, 2009.

[58] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[59] B. Lucas and b. p. y. Kanade, T. An iterative image registration technique with an application to stereo vision.

[60] S. Ma, J. Zhang, N. Ikizler-Cinbis, and S. Sclaroff. Action recognition and localization by hierarchical space-time segments. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2744–2751, 2013.

[61] B. Mahasseni and S. Todorovic. Latent multitask learning for view-invariant action recognition. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 3128–3135, 2013.

[62] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2929–2936, 2009.

[63] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *Proc. IEEE. Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pages 514–521, 2009.

[64] L. Meng, L. Qing, P. Yang, J. Miao, X. Chen, and D. Metaxas. Activity recognition based on semantic spatial relation. In *Proc. International Conference on Pattern Recognition (ICPR)*, pages 609–612, 2012.

[65] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *Proc. IEEE. Int. Conf. on Computer Vision (ICCV)*, pages 104–111, 2009.

[66] T. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, pages 90 – 126, 2006.

[67] J. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. European Conference on Computer Vision (ECCV)*, pages 392–405, 2010.

[68] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[69] J. C. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc. European Conference on Computer Vision (ECCV)*, pages 392–405, 2010.

[70] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, pages 299–318, 2008.

[71] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1817–1824, 2013.

[72] P. Perez, M. Marszalek, A. Zisserman, and I. Reid. High five: Recognising human interactions in tv shows. In *British Machine Vision Conference (BMVC)*, 2010.

[73] R. Poppe. A survey on vision-based human action recognition. *Image Vision Computing*, pages 976–990, 2010.

[74] L. Rabiner, C. Lee, B. Juang, and J. Wilpon. Hmm clustering for connected word recognition. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 405–408, 1989.

[75] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2650–2657, 2013.

[76] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *Proc. European Conference on Computer Vision (ECCV)*, pages 577–590, 2010.

[77] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, pages 971–981, 2013.

[78] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[79] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision (IJCV)*, pages 125–141, 2008.

[80] M. S. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Proc. IEEE. Int. Conf. Computer Vision (ICCV)*, pages 1593–1600, 2009.

[81] M. S. Ryoo, C.-C. Chen, J. K. Aggarwal, and A. Roy-Chowdhury. An overview of contest on semantic description of human activities (sdha) 2010. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, pages 270–285, 2010.

[82] S. Samanta and B. Chanda. Space-time facet model for human activity classification. *IEEE Transactions on Multimedia*, pages 1525–1535, 2014.

[83] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision (IJCV)*, pages 222–245, 2013.

[84] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proc. International Conf. on Pattern Recognition (ICPR)*, pages 32–36, 2004.

[85] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proc. ACM. Int. Conf. Multimedia (MULTIMEDIA)*, pages 357–360, 2007.

[86] Y. S. Sefidgar, A. Vahdat, S. Se, and G. Mori. Discriminative key-component models for interaction detection and recognition. *Computer Vision and Image Understanding (CVIU)*, pages 16–30, 2015.

[87] L. Shao, L. Liu, and M. Yu. Kernelized multiview projection for robust action recognition. *International Journal of Computer Vision*, pages 1–15, 2015.

[88] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 370–377, 2005.

[89] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1470–1477, 2003.

[90] M. Sokolova and G. Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, pages 427 – 437, 2009.

[91] K. Soomro, H. Idrees, and M. Shah. Action localization in videos through context walk. In *Proc. IEEE Int. Conference on Computer Vision (ICCV)*, pages 3280–3288, 2015.

[92] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2004–2011, 2009.

[93] 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 *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2632, 2014.

[94] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Proc. European Conference on Computer Vision (ECCV)*, pages 140–153, 2010.

[95] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2649, 2013.

[96] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *European Conference on Computer Vision (ECCV)*, pages 548–561. 2008.

[97] D. Tran and J. Yuan. Max-margin structured output regression for spatio-temporal action localization. In *Advances in neural information processing systems*, pages 350–358, 2012.

[98] A. Vahdat, B. Gao, M. Ranjbar, and G. Mori. A discriminative key pose sequence model for recognizing human interactions. In *Proc. IEEE. Int. Conf. Computer Vision Workshops (ICCV Workshops)*, pages 1729–1736, 2011.

[99] J. Van Gemert, J.-M. Geusebroek, C. Veenman, and A. Smeulders. Kernel codebooks for scene categorization. In *Proc. European Conference on Computer Vision (ECCV)*, pages 696–709. 2008.

[100] J. Van Gemert, M. Jain, E. Gati, and C. Snoek. Apt: Action localization proposals from dense trajectories. In *British Machine Vision Conference (BMVC)*, 2015.

[101] S. Vishwakarma and A. Agrawal. A survey on activity recognition and behavior understanding in video surveillance. *The Visual Computer*, pages 1–27, 2012.

[102] D. Waltisberg, A. Yao, J. Gall, and L. Van Gool. Variations of a hough-voting action recognition system. In *Proc. Int. Conf. Pattern Recognition (ICPR)*, pages 306–312, 2010.

[103] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176, 2011.

[104] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *Proc. European Conf. on Computer Vision*, pages 565–580, 2014.

[105] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2008.

[106] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, pages 224–241, 2011.

[107] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. European Conference on Computer Vision (ECCV)*, pages 650–663, 2008.

[108] L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, pages 715–770, 2002.

[109] S.-F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007.

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

[111] T.-H. Yu, T.-K. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forest. In *Proc. of the British Machine Vision Conference (BMVC)*, pages 1–12, 2010.

[112] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 2442–2449, 2009.

[113] K. Yun, J. Honorio, D. Chattopadhyay, T. Berg, and D. Samaras. Two-person interaction detection using body-pose features and multiple instance learning. In *Proc. IEEE. Int. Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 28–35, 2012.