Human Action and Interaction Recognition in surveillance videos

Master of Science Thesis

By
DELIANG WU

Utrecht University

Department of Information and Computing Science
UNIVERSITY UTRECHT
Supervisor: dr. ir. R.W. Poppe

Wuhan University

School of Remote Sensing and Information Engineering
WUHAN UNIVERSITY
Supervisor: Prof. dr. Zhenzhong Chen

A dissertation submitted to the University of Utrecht in accordance with the requirements of the degree of MASTER OF SCIENCE in Artificial Intelligence .

JANUARY 2017

Word count: ******

Abstract

Here goes the abstract

Dedication and acknowledgements

Here goes the dedication.

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

The technique of automatic video content analysis (VCA) is one of the most important areas of Computer Vision and Artificial Intelligence. With this technique, machines can recognize objects, human activities and events in videos. Thus, it can be widely used in many domains, such as human-computer interaction, video classification, entertainment, self-driving, public safety and security, home automation, etc. Action and interaction analysis is one of the most common uses of VCA which focus on human activities analysis, including action recognition/detection of a single person or interaction recognition/detection of two or more people.

1.1 Background

When we talk about human action, we usually mean the activity of a single person. While human interaction is more complex which generally contains three cases: human-human interaction among two or more people like hand-shaking between two people, human-object-human interaction like one person pass an object to another one, and human-object interaction like one person push a table.

The goal of **human action/interaction recognition** in a video is to determine the action/interaction label for a video that is what are the people doing in the video. And the goal of **human action/interaction detection** in a video is to localize a specific action/interaction spatially and temporally in a video.

Although a lot of excellent methods and datasets were published in the last decade in the area of action recognition, it is still challenging and far from being solved. Comparatively, the related work in the domain of interaction recognition and detection is relatively scarce. This is because the job of interaction recognition and detection is even more challenging than that of action.

The main challenges of action and interaction recognition and detections in real scenes mainly include:

1. Various camera views, the videos which will be analysed could be taken from different viewpoints which have never been seen in training data. For example, Figure 1.1 (a) illustrates four videos of a same biking little girl being filmed from four different camera views. The features extracted from these videos can be various, then it becomes hard for the classifier to learn to discriminate them as same activity.

2. Complex background, the background of the interested action and interaction could be various and even totally different. For example, Figure 1.1 (b) illustrates four diving videos taken from totally different background with large camera motion. For those feature descriptors which extract features from not only the segmented person but also the background. Then the features extracted from background would largely confuse the classification results.

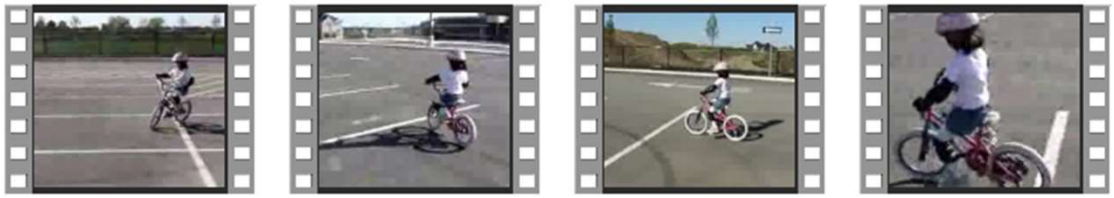
3. Usually, a single action video clip contains hundreds and even more frames, therefore there might be many irrelevant frames which would confuse the analysis. For instance, a video clip of playing basketball may contain frames of commentators and audiences.

4. It is hard to get a decent performance on a small training set. But sometimes we only have small scale target dataset with very limited training set. Such situation is even worse for the task of interaction video analysis since the related dataset is scare compared with action video analysis.

5. Compared with action video analysis, extra features like the relative position and orientation between people involved in the interaction need to be taken into consideration, which makes the feature representation more complex.

Some of interaction analysis related works are [23], [30], [19] and [2]. Where the works of Patron-Perez et al. [23] and Gemeren et al. [30] focus on relative information between people involved in the interaction, such as relative position and orientation in relevant body parts, then use hand-crafted features descriptors, like histograms of orientated gradients (HOG [3]) and histograms of optical flow (HOF [5]), to describe those interaction features. The work of Narayan et al. [19] combines improved trajectory features and foreground motion map features to describe interaction features. Differently, Choi et al. [2] introduce a hierarchical method, which first detects and tracks the location of each person, then learns to analysis the atomic action for each person, and last use the atomic action to infer the collective interaction label. The atomic action is also represented by hand-crafted feature descriptors, such as HOG and bag of video words (BoV [6]).

In this thesis project, we adopt the similar hierarchical architecture with the work of Choi et al. with two main changes: 1). we use deep learning feature descriptors to represent the atomic



(a). A same biking girl filmed in 4 different views

(b). 4 diving videos with totally different background

Figure 1.1: Illustration of some challenges of video analysis.

action for each person involved in the interaction instead of hand-crafted feature descriptors; 2). Besides the networks learning atomic features for each person, we add an additional deep learning network to learn the relative position and orientation features.

1.2 Project Goals

In this thesis project, we will focus on two-person interaction recognition and detection. The goals of this thesis project including:

1. Do interaction recognition based on the target dataset. The inputs are the segmented specific interaction videos and the outputs are the predicted class labels.

2. Do interaction detection based on the target dataset. The input is the un-segmented videos and the output is the spatial and temporal location of specified classes.

3. Construct a hierarchical multi-level network to learn interaction features. Hierarchical network means we first learn atomic action features for each person involved in interaction, then combine these features to learn interaction features. Multi-level network means we have global first level network which learns global interaction features, such as relative position and orientation between people involved in the interaction and second level network which learns atomic actions for each person.

1.3 Contributions

We mainly have following contributions on interaction video analysis:

1. We introduce deep learning feature descriptors to address interaction recognition and detection with only small scale target interaction video dataset available.

2. We introduce a hierarchical multi-level framework for interaction video analysis tasks.

1.4 Outline

In the next chapter, we will introduce the action and interaction video analysis related works including methods and datasets. In Chapter 3, we will introduce our overall architecture, training strategies and the metric method of this project. In Chapter 4, we will descriptor the low-level design of this project in detail. In Chapter 5, we will introduce the training process and experiment results, and analyse the results. At last, we will give conclusions and possible future works of this project.

CHAPTER 2

RELATED WORK

Due to the large potential practical values in many domains and the big technical challenges, video analysis is a very hot research topic in both academic and industrial communities in recent years. Since the related works of interaction video analysis [23] [30] [19] [2] are relatively scarce compared with that in the closely related areas, like action video analysis [10] [20] [29] [12] [7] [11] [27], we will both introduce the architectures used in interaction and action video analysis related works in Section 2.1. The general architecture of a video recognizer usually consists of a feature descriptor and a classifier. Video feature description research can be mainly divided into two directions: hand-crafted feature descriptors and learning based feature descriptors. We will introduce these two types separately in Section 2.2 and 2.3. Another very important factor determines the performance of video analysis is the training datasets. Without rich and nicely annotated training dataset, it is hard to get a decent performance even for the best-built algorithm. So, we will introduce popular publicly available action and interaction video datasets in Section 2.4.

2.1 Architectures of Interaction video analysis related works

Choi et al. [2] introduce a hierarchical network to recognize the collective activity of a group of people. The hierarchical model is illustrated in Figure 2.1 (b). O_i and O_j are the features for each individual in the video, and O_c is crowd context feature which represents the overall information of the video. A is the atomic action for each individual, I is the interaction between two individuals and C is the collective activity of the group of people. For example, the collective activity "$gathering$", illustrated in Figure 2.1 (a), is characterized as a collection of interactions (such as "$approaching$") between individuals. Each interaction is described as pairs of atomic activities (for example "$facing-right$" and "$facing-left$"). Each atomic activity is associated

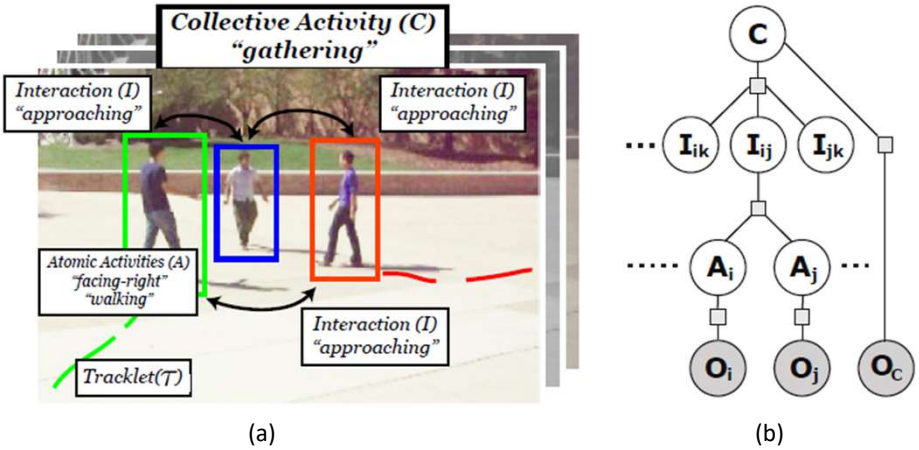


Figure 2.1: The hierarchical activity recognition model and an example. Reprinted from [2].

with a spatial-temporal trajectory. [2] use HOG to represent appearance features and use bag of video words (BoV [6]) to represent spatial-temporal features. Compared with the work of Choi et al. which use hierarchical model to recognize group activities, Gemeren et al. [30] extract interaction features from local body parts of interacting people and focus on those parts of videos that characterize the interaction. This is helpful to distinguish between interactions that differ only slightly. HOG (appearance) and HOF (movement) are combined as feature descriptors in this work.

The work of Patron-Perez et al. [23] is similar with [30]. Patron-Perez et al. detect and track the upper body for each person in the interaction video and crop the tracklet to generate a person-centred descriptor with HOG and HOF. Besides, the head orientation is also concerned in the feature descriptor because head orientation is also an important cue for interaction video analysis.

There are also some methods which use bag of local features to describe the features of interaction, like the work of Yimeng et al. [36] and Yu et al. [13]. Yimeng et al. propose the concept of spatio-temporal phrase which is a combination of local features in a certain spatial and temporal structure including their order and relative positions, then the video is represented by a bag of spatio-temporal phrases. Yu et al. propose to use a set of attributes and the pair-wise co-occurrence relationship of two attributes to describe the videos. An attribute is a binary representation of a body part, like 'the torso is still'. A co-occurrence relationship of two attributes is a binary representation of the association between two attributes, for example, the co-occurrence relationships between attributes "torso bending" and "still leg" in the activity "bow".

2.2 Hand-crafted Feature Descriptor

Scale-invariant feature transform (SIFT [16] [17]) and histogram of oriented gradients (HOG [3]) feature descriptors achieve great successes and are widely applied in the image content analysis.

SIFT is a local feature describing algorithm which detects interest points in an image and computes the gradients for each interest point to construct features of an image. SIFT algorithm adopts difference of Gaussian (DoG) algorithm to detect the interest points by constructing an image pyramid with several scales and blurring images with several different Gaussian filters for each scale. The interest points are those points with maximum values compared with their neighbor pixels in the image pyramid. Thus, those points with edges and corners which represent the main feature of an object are most likely to be found as interest points. Because interest points are detected in different scales, SIFT feature descriptor is scale invariant. Gradient computing is applied for the interest point centered blocks for each interest point. The computed histogram of gradient features are rotated to the main orientation for each interest point, thus it is rotation invariant. At last, the features are normalized to further eliminate the effect of different lighting.

For HOG algorithm, the image is divided into several small cells, histogram of gradient (orientation and magnitude) are computed for each cell. This is because the appearance and shape of an object can be nicely described by the gradient. HOG feature descriptor is somewhat transform invariant because the features are computed in local cells. And because the features are normalized over sliding overlap blocks which contain several cells in a block. So, it is lighting invariant.

Though SIFT and HOG can efficiently describe appearance and shape features for images, but they can not be directly used for video analysis task, because the video analysis requires not only appearance and shape description but also motion description. Thus, they are also extended to 3D-SIFT[7] [26] and 3D-HOG[12] to describe the video features. Among all current hand-crafted feature descriptors, improved Dense Trajectories(iDT)[32][33] have been shown to perform best on a variety of datasets.

2.2.1 3D-SIFT Feature Descriptor

Compared with 2D-SIFT[16][17] which only considers x and y dimensions, 3D-SIFT[7][26] takes another time(t) dimension into consideration, because the motion information contained in time dimension is an essential cue for video analysis. There are two steps to construct the SIFT feature descriptor. The first step is key points localization and the second step is calculating the sub-histograms for each key point. The method of key points localization is as same as 2D-SIFT. The main differences between 2D-SIFT and 3D-SIFT is calculating the sub-histogram for each key point.

The 2D gradient magnitude and orientation for each pixel are calculated in x and y dimensions while 3D-SIFT calculates gradients in three dimensions x, y and t. Thus, the motion information

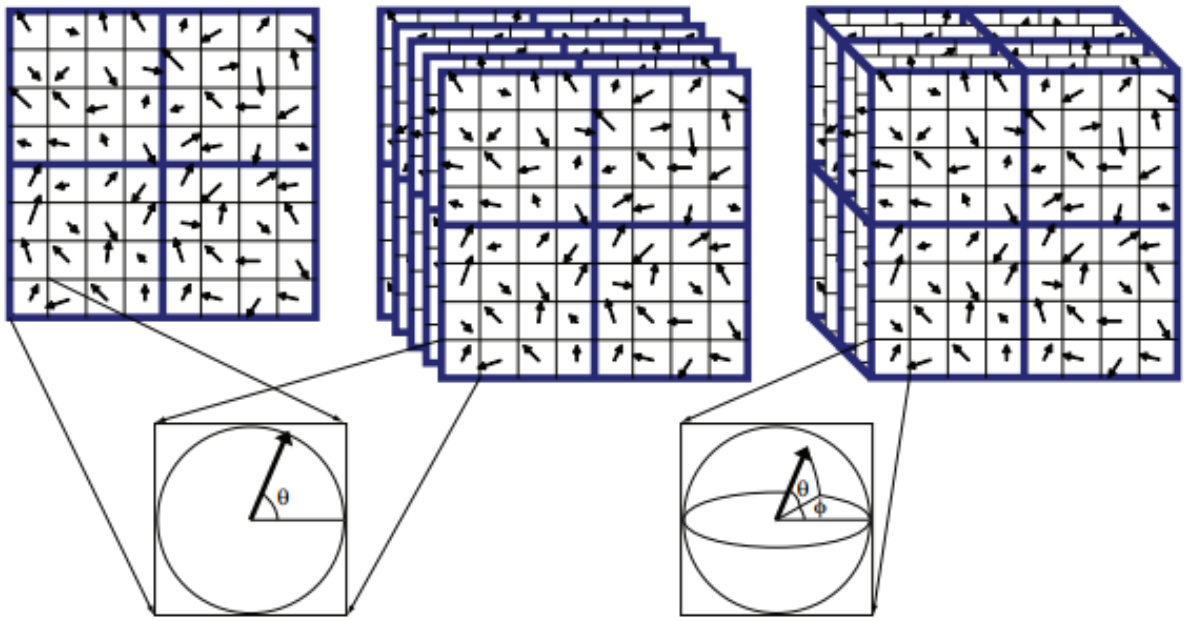


Figure 2.2: The SIFT descriptor. The left image shows the 2D SIFT descriptor. The center image shows how multiple 2D SIFT descriptor could be used on a video without modification to the original method. The right image shows the 3D SIFT descriptor with its 3D sub-volumes, each sub-volume is accumulated into its own sub-histogram. These histograms are what makes up the final descriptor. Reprinted from [7].

along the time dimension is also well presented by the sub-histogram of 3D gradients.

The 3D-SIFT feature descriptor is illustrated in Figure 2.2. After getting the sub-histogram, the orientation of each key point could be fixed. In 3D-SIFT, the dominant orientation of key point could be represented by θ and ϕ. To keep the orientation invariant, all neighbourhood of each key point are rotated to the dominant orientation. 3D-SIFT also inherits the method of key points detection and histogram normalization from 2D-SIFT, thus, it is also scale and lighting invariant.

2.2.2 3D-HOG Feature descriptor

2D-HOG is widely used for the purpose of object detection in static images. 2D-HOG calculates the gradient orientation for each pixel and statics them as histogram of orientated gradient over cells. Klaser et al. extended it to 3D-HOG[12]. In Klaser's work, the main differences between 2D and 3D HOG is the calculation of gradient. In 2D-HOG, only two dimensions x and y are considered, the cell is a square. While in 3D-HOG, additional time dimension t is taken into consideration, so, the cell is a cube. The blocks which used to contrast-normalization also became from 2D square to 3D cube. The overview of the descriptor computation is illustrated in Figure 2.3.

For the similar reason as that in 3D-SIFT, 3D-HOG calculates gradients in both spatial and

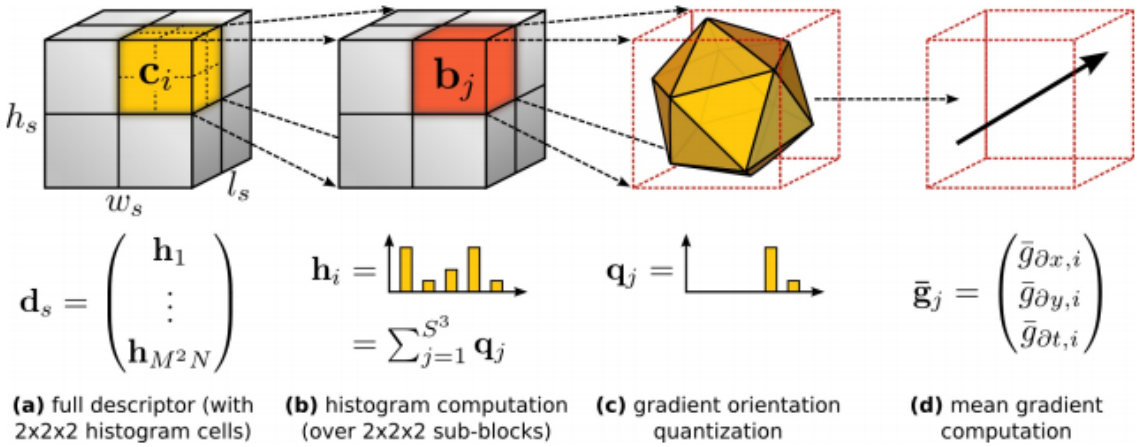


Figure 2.3: Overview of 3D-HOG descriptor computation; (a) the support region around a point of interest is divided into a grid of gradient orientation histograms; (b) each histogram is computed over a grid of mean gradients; (c) each gradient orientation is quantized using regular polyhedrons; (d) each mean gradient is computed using integral videos. Reprinted from [12].

temporal dimensions, so, the motion information contained in the time dimension can be nicely represented by 3D-gradients.

2.2.3 Improved Dense Trajectories feature descriptor

Improved Dense Trajectory (iDT) is a very successful algorithm for video action recognition among all hand-crafted feature descriptors which was proposed by Wang et al. [32] [33]. [32] introduces the Dense Trajectory (DT) algorithm and [33] improves DT algorithm by eliminating background optical flow trajectory caused by camera motion.

The overview of DT feature descriptor is illustrated in Figure 2.4, including dense sampling, key points tracking and features description. The video is firstly extended to several different spatial scales to keep this feature descriptor scale invariant.

Feature points are densely sampled on a grid spaced by W pixels. Sampling is carried out on each spatial scale separately. Since it is hard to track feature points in homogeneous areas, Wang et el. remove those points which have very small eigenvalues of the auto-correlation matrix.

Feature points are tracked on each spatial scale separately. Given a point $P_t = (x_t, y_t)$ in frame I_t, its tracked position in frame I_{t+1} is calculated by the position in frame I_t and the the components of optical flow computed w.r.t. I_t and I_{t+1}. Due to the fact that it is unstable to track a feature point in long time, all the feature points are re-sampled every L frames. Then for every feature point, the trajectory feature vector is $(P_t, P_{t+1}, P_{t+2}...., P_{t+L-1})$. The trajectory feature vector contains motion information of feature points thus it can represent motion information for videos.

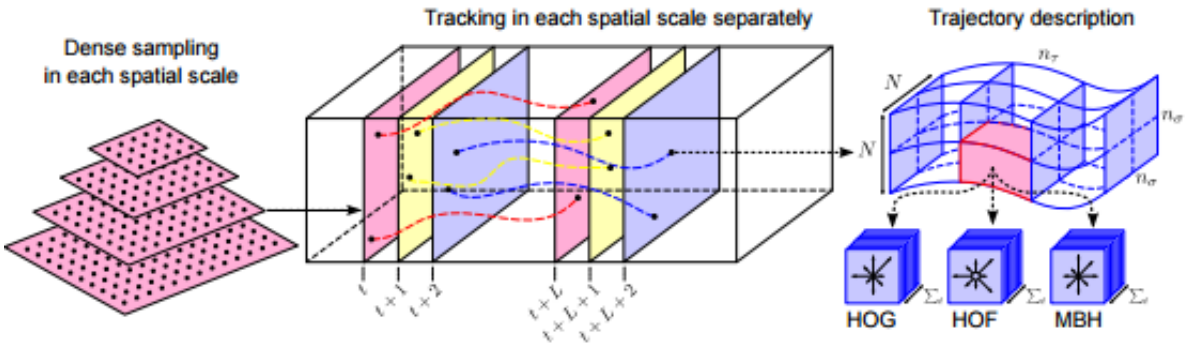


Figure 2.4: Illustration of DT algorithm to extract and characterize dense trajectories. Left: Feature points are densely sampled on a grid for each spatial scale. Middle: Tracking is carried out in the corresponding spatial scale for L frames by median filtering in a dense optical flow field. Right: The trajectory shape is represented by relative point coordinates, and the descriptors (HOG, HOF, MBH) are computed along the trajectory in a $N * N$ pixels neighbourhood, which is divided into $n\sigma * n\sigma * n\tau$ cells. Reprinted from [32].

Since only trajectory features are not enough to describe all the video features, Wang et al. introduce another three feature descriptors: Histograms of Orientated Gradient (HOG [3]), Histograms of Optical Flow (HOF [5] and Motion Boundary Histograms (MBH [5]). The HOG feature descriptor is similar with 3D-HOG but it is calculated along trajectory. HOF features represent the optical flow (movements) of objects in videos and MBH features are based on derivatives of optical flow which is a simple and effective way to suppress camera motion. 3D-HOG can well represent the both appearance/shape and motion information for objects in videos. The combination of HOF and MBH can further improve the video analysis performance as they represent zero-order (HOF) and first-order (MBH) motion information. All of these three feature descriptors are calculated in a $N * N * L$ spatial-temporal volume around the feature point along the trajectory, see the right image in Figure 2.4. The spatial-temporal volume is sliced into smaller $n\sigma * n\sigma * n\tau$ spatial-temporal cells. $N = 32, n\sigma = 2, n\tau = 3$ in the work of Wang et al. The histogram of HOG, HOF and MBH are calculated over all pixels in each cell.

The work of iDT [33] improves DT algorithm mainly by eliminating the camera motion. In DT algorithm, due to the camera motion, there are many trajectories in the background, and the trajectories of interest can also be affected by the camera motion. But those trajectories of background are unuseful for action recognition and usually confuse the results. By assuming the difference between 2 consecutive frames is small, the iDT algorithm assumes that the frame I_{t+1} can be calculated from frame I_t and a transform matrix H, that $I_{t+1} = I_t * H$. Then we can calculate $Iwarp_{t+1} = H^{-1} * I_{t+1}$, where $Iwarp_{t+1}$ is the frame I_{t+1} after eliminating camera motion. Since the transformation matrix H is calculated over the whole image I_t and I_{t+1} which include both background and interest human. So, the large movement of human body in consecutive frames will largely affect the accuracy of matrix H. Wang et al. use human detection

technique to detect the human body in all images and mask these areas to get more accurate transform matrix H and use this matrix to eliminate useless trajectory of background and human body caused by camera motion.

2.3 Deep Learning Based Feature Descriptor

Though hand-crafted feature descriptors achieve very nice performance in image and video content analysis, they are all based on pre-defined rules to extract features. Thus, they usually ignore those potential cues for video analysis which not be realized by the feature designer. A deep learning based feature descriptor has complex parameters and flex structure. A well trained deep learning feature descriptor can represent similar features represented by hand-crafted feature descriptor. For example, Convolutional Neural Network (CNN or ConvNet [15]) can learn edges in lower layer which is similar with HOG. Further, well trained deep learning feature descriptor has potential abilities to represent some features which are hard for hand-crafted features descriptors. That is learning feature descriptor is more general than hand-crafted feature descriptor. Due to the significant development of data science, deep learning based feature descriptor, especially CNN, shows powerful ability of feature representation and has achieved state of art performance in image classification/recognition [8].

Convolutional Neural Network is one type of deep artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of animal visual cortex. The architecture of one of the very first ConvNet, LeNet5 by Yann LeCun et al. [15], is illustrated in Figure 2.5. A typical ConvNet includes three main parts: convolutional layers, pooling/subsampling layers and fully connected layers. Convolutional layers extract features from input image by applying convolution. The convolution can be understood as a feature filter applying over the input image. We can perform operations such as edge detection just by setting different filter parameters. So, one filter (convolutional kernel) can extract one type of features. The more filters, the more features can be represented by the ConvNet. An intuitive example of convolution is illustrated in Figure 2.6. The pooling/subsampling layer reduces the dimensionality of each feature map but retain the most import information. There are different types of pooling: Max which takes the largest element in a $n \times n$ window; and Average which takes the average value of all elements in the window. Pooling operation can simplify the features thus has following advantages: 1). Making the network smaller and more manageable; 2). Reducing the number of parameters and computation, therefore controlling over-fitting; 3). Making the network invariant to small transformations, distortions, etc. Fully connected layer is a traditional multi layer perceptron in which every neuron in the previous layer is connected to every neuron on the next layer. The output of convolutional and pooling layers represents high-level features of input images while the fully connected layers use these features to classify these input images into various classes.

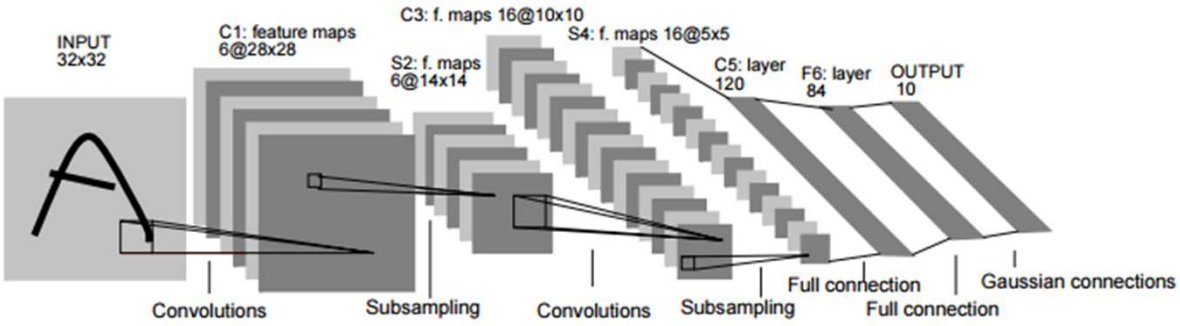


Figure 2.5: The architecture of a CNN example: LeNet5. Reprinted from [15].

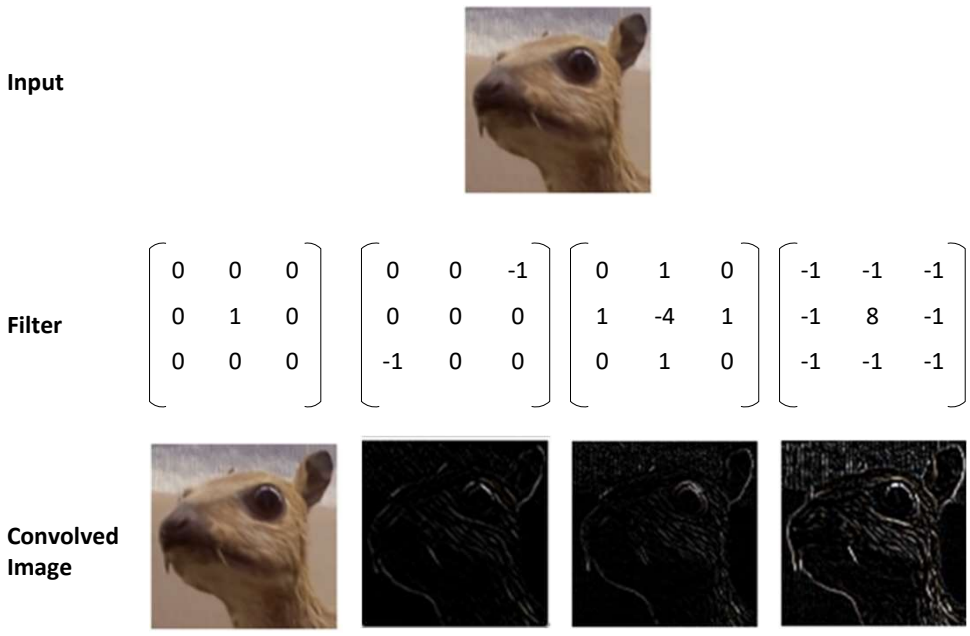


Figure 2.6: The intuitive illustration of convolution over image.

There is related work [21] in the earlier stage which just simply applies CNN and treats videos as set of single frames and averages the results over all frames as the overall result. The performance of such method is not very satisfying since it doesn't take the important temporal information into consideration and there may be many irrelevant frames in the video will confuse the result. In recent years, many new related works are proposed to solve these problems, like Spatial-Temporal CNN[11], Two-Stream ConvNet[27], and 3D Convolutional Networks(3D-ConvNet) [1] [10] [29].

2.3.1 Spatial-Temporal CNNs feature descriptor

Since spatial information represents the appearance and temporal information represents motion, they are both essential for video analysis. Karpathy et al. suggested a spatial-temporal CNN in their work[11]. They studied several approaches to extend the connectivity of CNNs to time domain to make use of spatial-temporal information of videos, including late fusion, early fusion and slow fusion. The architectures of fusion CNNs are illustrated in Figure 2.7.

Single frame architecture is used as a base-line in the work [11]. It is a standard CNNs framework and similar to the ImageNet challenge winning model [14] just with the different input image resolution. The architecture of the single frame network is illustrated in Figure 2.7 (a).

Late fusion model, illustrated in Figure 2.7 (b), places two separate single frame networks which share network parameters. The inputs of two networks are two streams which have distance of 15 frames. The two networks are merged in the first fully connected layer. Neither single network alone can detect any motion, but the first full connected layer can compute global motion characteristics by comparing outputs of the two networks.

Early fusion model, illustrated in Figure 2.7 (c), is similar to single frame model but the input is consecutive T frames instead of one single frame. T was set to 10 in the work [11]. The early and directly connectivity to pixel data allows the network to precisely detect local motion direction and speed.

The slow fusion model, illustrated in Figure 2.7 (d), combines the early and late fusion model. This model slowly fuses temporal information throughout the network such that higher layers get access to progressively more global information in both spatial and temporal dimensions.

2.3.2 Two-Stream ConvNet feature descriptor

Two-Stream ConvNet proposed by Simonyan et al. [27] is another extension of ConvNet to action recognition in video data. The Two-Stream ConvNet introduces a different architecture with Spatial-Temporal CNNs [11] based on two separate recognition streams (spatial and temporal), which are combined by late fusion. The spatial stream learns appearance features from still video frames while the temporal stream learns motion features in the form of dense optical flow of input videos. So, the combination of spatial and temporal streams can represent both appearance/shape

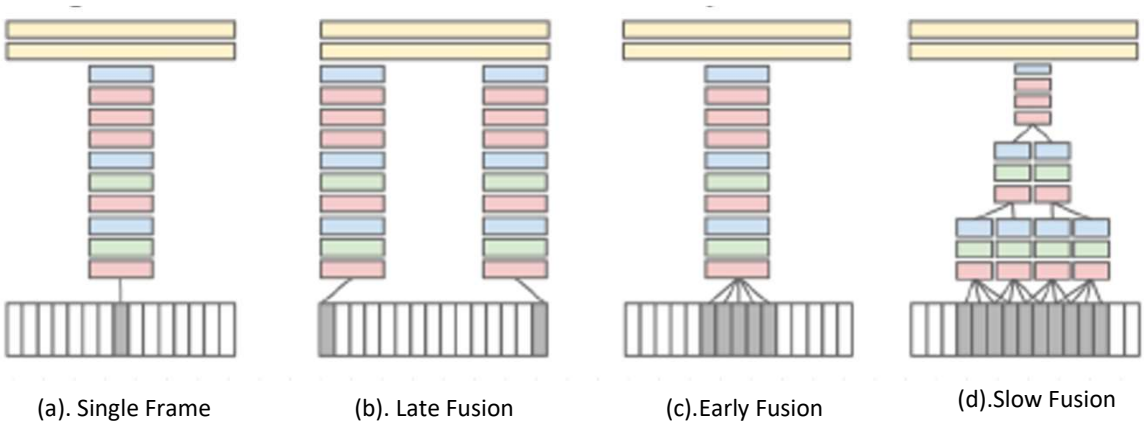


Figure 2.7: Explored approaches for fusing information over temporal dimension through the network. Red, green and blue boxes indicate convolutional, normalization and pooling layers respectively. In the Slow Fusion model, the depicted columns share parameters. Reprinted from [11].

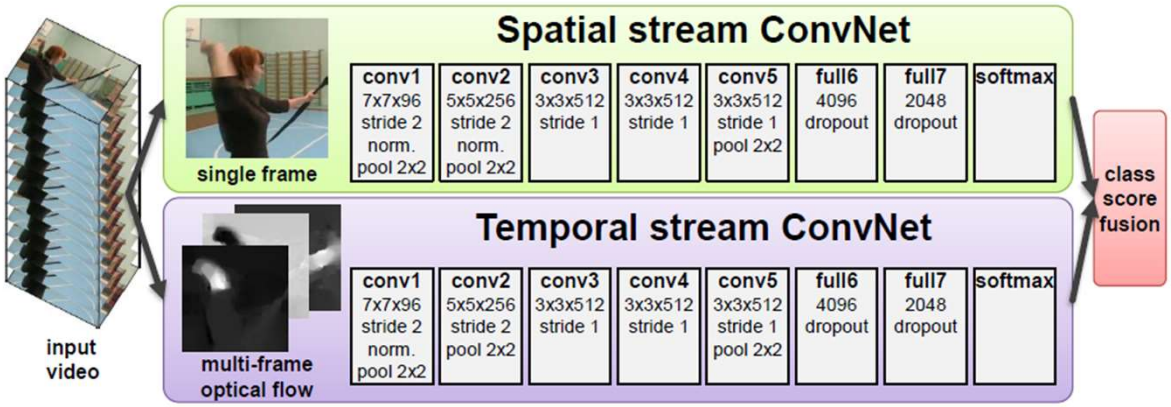


Figure 2.8: The architecture of Two Stream ConvNet. Reprinted from [27].

and motion for the input videos. The architecture of Two Stream ConvNet is illustrated in Figure 2.8.

2.3.3 3D ConvNet feature descriptor

3D ConvNet is first proposed by Ji et al. [10] with human body segmented video volumes as input and Tran et al. [29] improved it to accept full raw video volumes as input without any pre-processing. The main difference between 2D ConvNets and 3D ConvNets is the convolution and pooling are applied in three dimensions(x,y,time) instead of two(x,y). As illustrated in Figure 2.9, 2D convolution applied on a single image or a video volume (multiple frames as multiple channels) all result in an image. While 3D Convolution on a video volume results in another

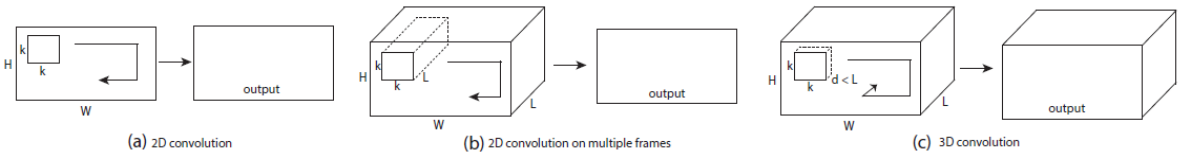


Figure 2.9: 2D and 3D convolution operations. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal. Reprinted from [29].

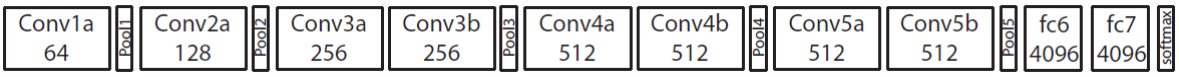


Figure 2.10: The architecture of C3D. C3D has 8 convolution, 5 max-pooling, and 2 full connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from pool1 to pool5. All pooling kernels are $2 \times 2 \times 2$ except for pool1 is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units. Reprinted from [29].

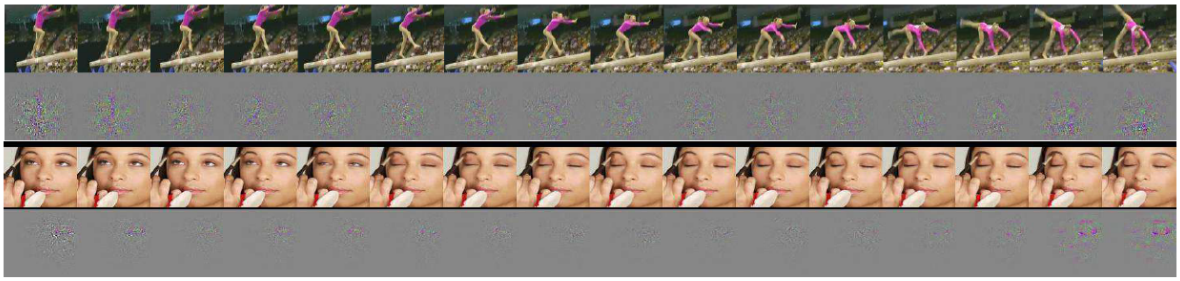


Figure 2.11: Visualization of C3D model. Interestingly, C3D captures appearance for the first few frames but thereafter only attends to salient motion. Reprinted from [29].

video volume. So, the video temporal information is lost after applying 2D convolution, while 3D convolution preserves both spatial and temporal information.

The network architecture used in [29] annotated as C3D is illustrated in Figure 2.10. The input of the network is a 16 frames video volume. Videos with frame number larger than 16 will be split into several 16 frames video clips with an overlap of 8 frames for each clip. All video clips for a video will calculate features separately and simply average them to get a 4096 dimensions features as the whole video's features.

The Figure 2.11 illustrates what the 3D ConvNet learns by using the deconvolution method explained in [35]. In the two examples, the features first focus on appearance and gradually track the motion over the rest of frames.

2.4 Datasets

Training dataset is one of the key factors to train a high performance feature descriptor. At the same time, published and widely used datasets can be helpful to compare between different approaches. After the publication of the KTH [25] dataset which contains 6 action classes and 600 videos (2391 sequences) in 2004, more and more human activity datasets were published. In recent years, the development of dataset has the following characteristics:

1. More action classes, the KTH has 6 classes, UCF101 [28] has 101 action classes and ASLAN [22] has 432 classes.

2. More training and testing samples, KTH has 192 videos for training and 216 videos for testing while ActivityNet [9] 200 has 10024 videos for training and 5044 videos for testing.

3. The video scene becomes more and more complex, KTH is acted by limited number of actors, while recent datasets are cut from realistic scenes, like youtube, cctv, BBC etc.

4. More challenges in video content, from fixed backgrounded without camera motion to non-static camera, multi-viewpoints, more complex background; from single person action to human-human interaction, human-object interaction, etc.

2.4.1 List of human activity video datasets

Part of widely used action and interaction video datasets are illustrated in Table 2.1 and Table 2.2 respectively.

Table 2.1: List of human action video datasets

Dataset	Classes	Videos	Annotations	Properties
KTH[25]	6 action classes	• Training : 192 • Validation: 192 • Testing: 216 • Resolution: 160×120 @ 25fps	• Action labels • Temporal segments	• Static camera • Simple background • Acted by 25 subjects, 6 actions and 4 scenarios
Hollywood2[18]	12 action classes and 10 scene classes	For actions: • Training : 823 • Testing: 884 For scenarios: • Training : 570 • Testing: 582 Resolution: $400 - 300 \times 300 - 200$	• Action labels • Scene labels	• Non-static camera • Realistic scenarios from movies
UCF101[28]	5 types, 101 action classes	• Training : 9537 • Testing: 3783 • Resolution: $400 - 300 \times 300 - 200$	• Action labels	• Realistic scenarios from YouTube • Large variations in camera motion, object appearance and pose, object scale, viewpoints, cluttered background, illumination conditions, etc.
ActivityNet 200[9]	200 action classes	• Training : 10024 • Validation: 4926 • Testing: 5044	• Action labels • Temporal segments • Hierarchy activities relationship	• Realistic scenarios • Large variations in camera motion, object appearance and pose, object scale, viewpoints, cluttered background, illumination conditions, etc.

Table 2.2: List of human interaction video datasets

Dataset	Classes	Videos	Annotations	Properties
UT-Interaction[24]	6 interaction classes	• 20 videos • 2 sets, 10 for each set. Evaluated by leave one out cross validation. • Resolution = 720 × 480	• Interaction labels • Temporal segments • Spatial segments	• Two-person interaction • Static camera • Acted in different background
ShakeFive2[31]	8 interaction classes	• 153 videos • Resolution = 1280 × 720	• Interaction labels • Joint position • Temporal segments	• Homogeneous background • Static camera
Multi-modal & Multi-view & Interactive [34]	9 interaction classes and 13 person-object interaction classes	• 1760 RGB videos • 1760 depth videos • Resolution = 320 × 240	• Interaction labels • Joint position • Foreground mask	• Homogeneous background • Static camera

CHAPTER 3

ARCHITECTURE

This project focuses on two-person human-human interaction recognition and detection in videos. Almost all of the few previous works [23] [19] [2] in two-person human-human interaction recognition adopt hand-crafted feature descriptors. For example, Narayan et al. [19] combine improved trajectory features and foreground motion map features to present interaction features while Patron-Perez et al. [23] and Choi et al. [2] use a hierarchical model to present interaction features. In the hierarchical model, each individual is tracked throughout the videos, and the atomic activity for each individual and their relative position and orientations are computed firstly then the collective interaction is represented. Our work adopts the hierarchical model similar to [23] but we replace the hand-crafted feature descriptors with deep learning feature descriptors since deep learning based methods [10] [29] [27] [20] have already achieve better performance in closely related domain, single person action recognitions in videos, than hand-crafted feature descriptors[7] [12] [26] [32] [33].

3.1 Overall Framework

Since the deep network has a very strong ability for feature learning and description, the most direct way is just feeding the whole video into a deep network and train it. Such a network can achieve decent performance if the training datasets are proper and rich enough. But for our project, two-person interaction recognition and detection, we only have three small scale two-person interaction datasets: **UT-Interaction** [24], **ShakeFive2** [31] and **MMI** [34] with very limited training samples. It is insufficient to train a deep learning network to achieve a nice performance on these datasets. But fortunately, we do have many large scale single person action datasets, like **UCF-101** [28],**Sports-1M**, **Activity-Net200** [9], etc. We can use those large

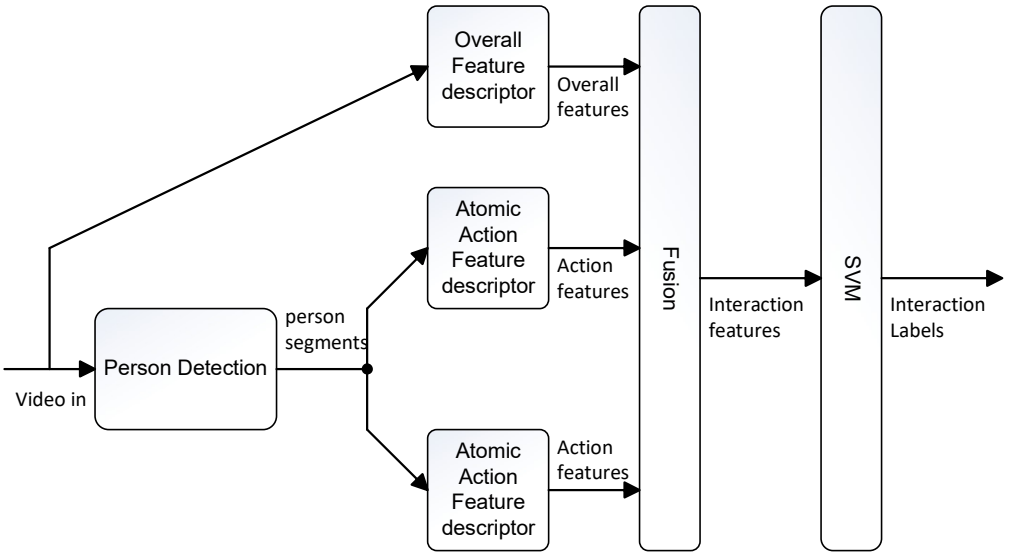


Figure 3.1: Overall framework.

scale dataset to pre-train our deep learning network and finetune it with the target interaction dataset.

Due to lack of sufficient proper two-person interaction datasets to just train a deep network, we adopt the hierarchical model similar with [2] and [23] which learn atomic action features for each individual in the video and combine these features to represent the collective interaction features. Different with [2] and [23] that use hand-crafted features, such as HOG and BoV, we use deep learning networks as the feature descriptors.

In our architecture, we first apply person detection to localize the positions of each person in the videos and then crop the video into two-video volumes, each containing one person's activity. Then we design an atomic action spatial-temporal feature descriptor for each person. Since there are lots of large scale single person action datasets available, we can pre-train the atomic action feature descriptors with those datasets and finetune the parameters on interaction datasets.

One drawback of this hierarchical model is ignoring the important relative information between the two-person, such as relative position and orientation, etc. So, we introduce another overall feature descriptor to learn those relative information which can be trained directly from our target dataset **UT-Interaction**.

The fusion of all learned features are fed to a SVM classifier. The overall framework is illustrated in Figure 3.1.

3.2 Person Detection

In this project, we have three interaction video dataset available, **UT-Interaction**, **ShakeFive2** and **MMI** respectively. And joint information is provided for **ShakeFive2** and **MMI**. In the

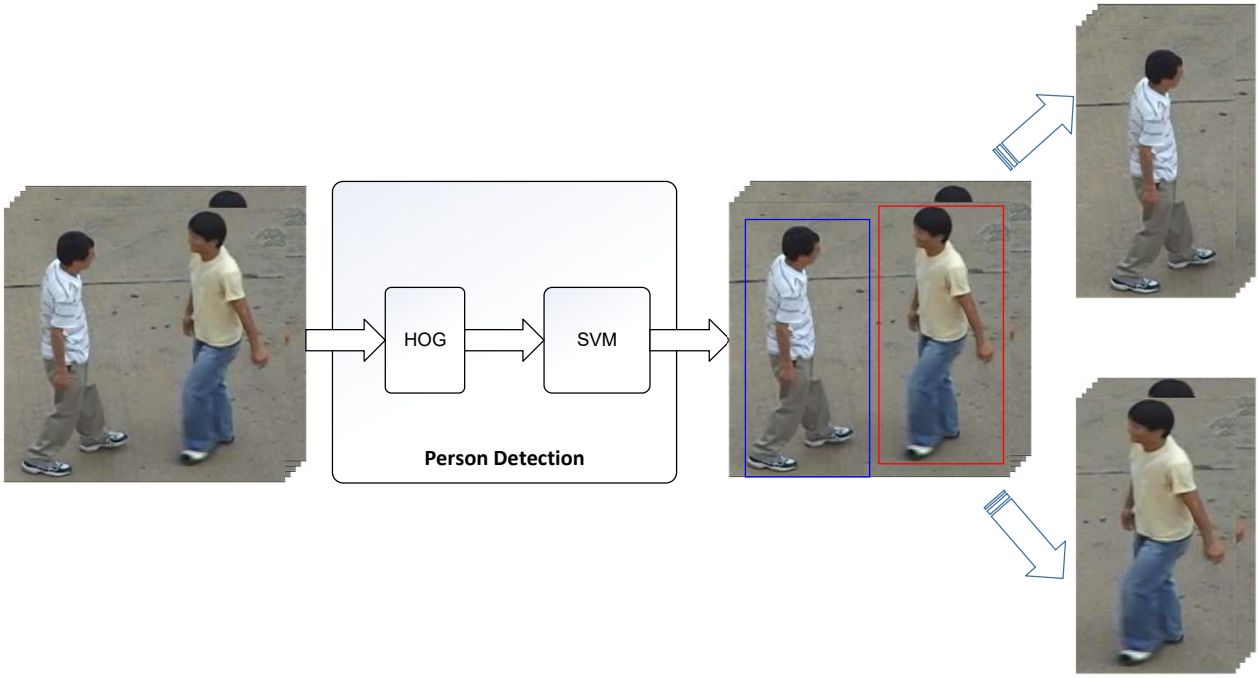


Figure 3.2: Illustration of Person detection

dataset **UT-Interaction**, all videos are filmed in static cameras with simple background and without much people occlusion. So, We use classical HOG plus SVM framework to construct person detector [4]. The structure of person detection is illustrated in Figure 3.2.

The Person Detection module will detect the location of people in each frame and track two people throughout a video. The crops for each person will be resized to the same size and construct two video volumes.

3.3 Feature Descriptors

We have two-level hierarchical feature descriptors in the project: an overall feature descriptor which learns the global features such as the relative position and orientation between two-person and two atomic feature descriptors which learn the local atomic action features for each person. The overall feature descriptor and atomic action feature descriptors share similar network frameworks. We use different parameters between overall feature descriptor and atomic feature descriptor while sharing the parameters between two atomic action feature descriptors. That is, we only train one atomic action feature descriptor for one person and totally duplicate it for another person.

Since temporal information is a very important factor for video analysis, we adopt two different types of spatial-temporal feature descriptor to present the overall and atomic action features: 3D

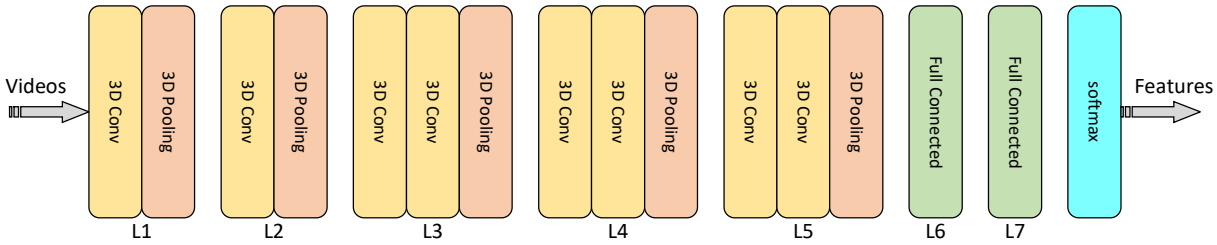


Figure 3.3: Overall Architecture of 3D-ConvNet.

Convolutional Network(3D-ConvNet)[29] and Two-Stream Convolutional Network(Two-Stream ConvNet) [27] respectively. Because above two type of spatial-temporal feature descriptor have already achieved very nice performance in video action recognition tasks. And they present the spatial-temporal features from different views: three dimensional(x,y,time) convolution for 3D-ConvNet and two dimensional(x,y) convolution on still frames plus multiple frame motion optical-flow convolution for Two-Stream ConvNet. We will compare their performance in interaction recognition and detection tasks among these two methods and even combine output features of these two methods together to see whether we can get better performance.

3.3.1 3D-ConvNet

3D ConvNet learns both spatial and temporal features at the same time by applying three-dimensional (x,y,time) convolution and pooling. We use the similar architecture as Tran et al.'s 3D-ConvNet[29] as our 3D-ConvNet based feature descriptor. The architecture of 3D-ConvNet is illustrated in Figure 3.3.

3.3.2 Two-Stream ConvNet

Different with a 3D-ConvNet learns spatiotemporal features in a single network, the two-stream ConvNet learns spatial features and temporal features in two separate 2D convolutional networks. The spatial stream network learns spatial features from single input frame and the temporal stream network learn temporal features from multiple frames of optical flow which represents the motion information between the consecutive frames. We take a similar architecture as the work of Simonyan et al.[27] as our two-stream ConvNet. The spatial and temporal features fuse at the end of two ConvNet to represent the overall features of input videos. The overall architecture of two-stream ConvNet is illustrated in Figure 3.4.

3.4 Training

Since the performance of a learning algorithm is highly dependent on the training datasets, it is important to select proper training datasets for each network and balance the performance and computational complexity at the same time.

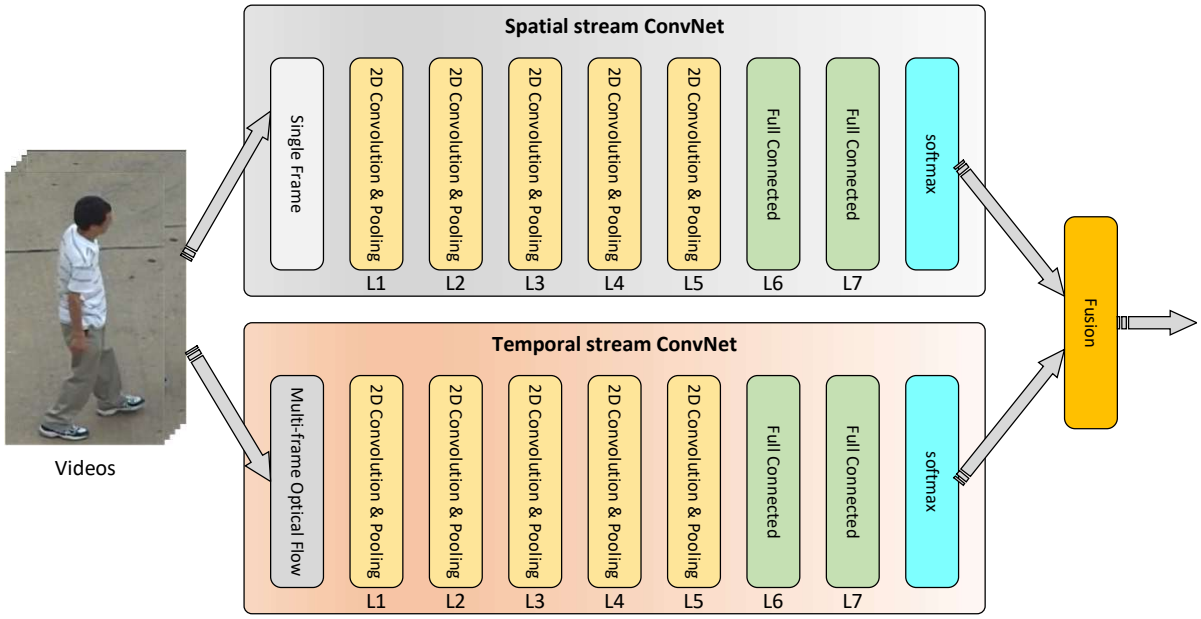


Figure 3.4: Overall Architecture of Two-Stream ConvNet.

3.4.1 Train Person Detection Network

We select **INRIA Person Dataset** [4] to train our person detection network. INRIA Person Dataset is one of the most used datasets for static people detection which provides original images and corresponding annotations. The training set contains 614 (2416 people) positive images and 1218 negative images. The testing set contains 288 (1126 people) positive images and 453 negative images. Only upright persons (with height larger than 100 pixels) are marked in each image. The sources of this dataset are mainly collected from GRAZ01 dataset, personal digital images, and Google images. Most of them have very high resolutions.

3.4.2 Train 3D-ConvNet

We adopt different training strategies between overall feature descriptor and atomic action feature descriptors.

Due to that the overall features descriptor learns to present interaction features and the target interaction video dataset **UT-Interaction** [24] is a small scale dataset, we will pre-train the overall feature descriptor on other interaction video datasets: **ShakeFive2** [31] or **MMI** [34], then fine-tune it on **UT-Interaction** dataset.

For the atomic action feature descriptors, because we have large scale single person action datasets available and the complexity of 3D-ConvNet, we adopt the pre-training and fine-tune strategy to train this network. Large scale action video dataset **UCF-101** [28] or **Activity-Net200** [9] will be used to pre-train the 3D-ConvNet.

In pre-training phase, we will feed the videos and labels of above two datasets to the 3D-

ConvNet plus a softmax classifier. Then the 3D-ConvNet will learn the features from those videos and labels.

In fine-tuning phase, we will feed videos and labels of **UT-Interaction** to the pre-trained network. All parameters of convolutional layers of pre-trained network remain and only parameters of fully connected layers will be fine-tuned.

3.4.3 Train The SVM Classifier

All output features of overall feature descriptor fg and atomic action feature descriptors $fa0$ and $fa1$ are fused and fed to a one-vs-the-rest SVM classifier. The SVM classifier is trained with the target dataset **UT-Interaction**.

3.5 Testing

We use **UT-Interaction**[24] to evaluate our interaction recognition and detection network for both interaction classification and detection. **UT-Interaction** dataset has 120 segmented videos (two sets, 60 videos for each set) for classification task and 20 videos with spatial and temporal ground truth annotations for detection task.

For test 'classification' task, the two sets of videos are evaluated separately. 10-fold leave-one-out cross validation is used to evaluate the classification accuracy. The performance is measured ten times and average accuracy is used as the overall accuracy.

For the 'detection' task, the interaction detection is measured to be correct if and only if the network correctly annotates an occurring interaction's time interval and spatial bounding box. If the annotation overlaps the ground truth more than 50% spatially and temporally, the detection is treated as a true positive. Otherwise, it is treated as a false positive.

Chapter 4

Design

CHAPTER 5

Experimental Results

CHAPTER 6

CONCLUSION

Appendix A

Begins an appendix

BIBLIOGRAPHY

[1] M. BACCOUCHE, F. MAMALET, C. WOLF, C. GARCIA, AND A. BASKURT, *Sequential deep learning for human action recognition*, Springer, (2011), pp. 29 – 39.

[2] W. CHOI AND S. SAVARESE, *A unified framework for multi-target tracking and collective activity recognition*, ECCV, (2012).

[3] N. DALAL AND B. TRIGGS, *Histograms of oriented gradients for human detection*, CVPR, (2005).

[4] N. DALAL AND B. TRIGGS, *Histograms of oriented gradients for human detection*, CVPR, (2005).

[5] N. DALAL, B. TRIGGS, AND C. SCHMID, *Human detection using oriented histograms of flow and appearance*, ECCV, (2006).

[6] P. DOLLAR, V. RABAUD, G. COTTRELL, AND S. BELONGIE, *Human detection using oriented histograms of flow and appearance*, VS-PETS, (2005).

[7] G. FLITTON, T. BRECKON, AND N. MEGHERBI BOUALLAGU, *Object recognition using 3d sift in complex ct volumes*, Proceedings of the British Machine Vision Conference, (2010), pp. 11.1–11.12.

[8] K. HE, X. ZHANG, S. REN, AND J. SUN, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, ICCV, (2015), pp. 1026– 1034.

[9] F. C. HEILBRON, V. ESCORCIA, AND B. GHANEM, *Activitynet: A large-scale video benchmark for human activity understanding*, CVPR, (2015).

[10] S. JI, W. XU, M. YANG, AND K. YU, *3d convolutional neural networks for human action recognition*, IEEE TPAMI, (2013).

[11] A. KARPATHY, G. TODERICI, S. SHETTY, T. LEUNG, R. SUKTHANKAR, AND L. FEI-FEI, *Large-scale video classification with convolutional neural networks*, CVPR, (2014).

[12] A. KLASER, M. MARSZALEK, AND C. SCHMID, *A spatio-temporal descriptor based on 3d-gradients*, British Machine Vision Association, (2008), pp. 275:1–10.

[13] Y. KONG, Y. JIA, AND Y. FU, *Learning human interaction by interactive phrases*, ECCV, (2012).

[14] A. KRIZHEVSKY, I. SUTSKEVER, AND G. E. HINTON, *Imagenet classification with deep convolutional neural networks*, NIPS, (2012).

[15] Y. LECUN, L. BOTTOU, Y. BENGIO, AND P. HAFFNER, *Gradient based learning applied to document recognition*, IEEE, (1998).

[16] D. G. LOWE, *Object recognition from local scale-invariant features*, ICCV, (1999).

[17] ——, *Distinctive image features from scale-invariant keypoints*, IJCV, (2004).

[18] M. MARSZAŁEK, I. LAPTEV, AND C. SCHMID, *Actions in context*, CVPR, (2009).

[19] S. NARAYAN, M. S. KANKANHALLI, AND K. R. RAMAKRISHNAN, *Action and interaction recognition in first-person videos*, CVPR, (2014).

[20] J. Y.-H. NG, M. HAUSKNECHT, S. VIJAYANARASIMHAN, O. VINYALS, R. MONGA, AND G. TODERICI, *Beyond short snippets: Deep networks for video classification*, CVPR, (2015).

[21] F. NING, D. DELHOMME, Y. LECUN, F. P. L. BOTTOU, AND P. E. BARBANO, *Toward automatic phenotyping of developing embryos from videos*, IEEE Trans. on Image Processing, (2005), pp. 1360–1371.

[22] L. W. ORIT KLIPER-GROSS, ORIT KLIPER-GROSS, *The action similarity labeling challenge*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2012), pp. 615–621.

[23] A. PATRON-PEREZ, M. MARSZALEK, A. ZISSERMAN, AND I. REID, *High five: Recognising human interactions in tv shows*, BMVC, (2010).

[24] M. S. RYOO AND J. K. AGGARWAL, *Ut-interaction dataset*, (2010).

[25] C. SCHULDT, I. LAPTEV, AND B. CAPUTO, *Recognizing human actions: A local svm approach*, ICPR, (2004).

[26] P. SCOVANNER, S. ALI, AND M. SHAH, *A 3-dimensional sift descriptor and its application to action recognition*, 15th ACM, (2007), pp. 357–360.

[27] K. SIMONYAN AND A. ZISSERMAN, *Two-stream convolutional networks for action recognition in videos*, CVPR, (2014).

[28] K. SOOMRO, A. R. ZAMIR, AND M. SHAH, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, CRCV-TR, (2012).

[29] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, AND M. PALURI, *Learning spatiotemporal features with 3d convolutional networks*, ICCV, (2015).

[30] C. VAN GEMEREN, R. POPPE, AND R. C. VELTKAMP, *Spatio-temporal detection of fine-grained dyadic human interactions*, Human Behavior Understanding. HBU 2016. Lecture Notes in Computer Science, vol 9997. Springer, Cham, (2016).

[31] ——, *Spatio-temporal detection of fine-grained dyadic human interactions*, 7th International Workshop on Human Behavior Understanding 2016, (2016), pp. 116–133.

[32] H. WANG, A. KLASER, C. SCHMID, AND C.-L. LIU, *Dense trajectories and motion boundary descriptors for action recognition*, IJCV, (2013), pp. 60–79.

[33] H. WANG AND C. SCHMID, *Action recognition with improved trajectories*, IEEE International Conference on Computer Vision, (2013).

[34] N. XU, A. LIU, W. NIE, Y. WONG, F. LI, AND Y. SU, *Multi-modal & multi-view & interactive benchmark dataset for human action recognition*, ICME 2015, (2015).

[35] M. D. ZEILER AND R. FERGUS, *Visualizing and understanding convolutional networks*, ECCV, (2014).

[36] Y. ZHANG, X. LIU, M. CHANG, W. GE, AND T. CHEN, *Spatio-temporal phrases for activity recognition*, ECCV, (2012).