THE CITY COLLEGE OF NEW YORK

DOCTORAL DISSERTATION

Human Activity Analysis using Multi-modalities and Deep Learning

Author:
Chenyang ZHANG

Advisor:
Dr. Yingli TIAN

A dissertation submitted to the Graduate Faculty in Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

The City College of New York

2016

Copyright Page

This manuscript has been read and accepted for the Graduate Faculty in Engineering in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Yingli Tian, Chair of Examining Committee

Date

Ardie D. Walser, Associate Dean for Academic Affairs

Date

EXAMINING COMMITTEE
Prof. Yingli Tian, Mentor,
Dept. of Electrical Engineering, The City College of New York

Prof. Jizhong Xiao,
Dept. of Electrical Engineering, The City College of New York

Prof. Zhigang Zhu,
Dept. of Computer Science, The City College of New York

Prof. Ioannis Stamos,
Dept. of Computer Science, Hunter College

Dr. Lisa M. Brown,
Thomas J. Watson Research Center, IBM

THE CITY COLLEGE OF NEW YORK

THE CITY COLLEGE OF NEW YORK

Abstract

Human Activity Analysis using Multi-modalities and Deep Learning

by
Chenyang ZHANG

Advisor: Professor Yingli Tian

With the successful development of video recording devices and sharing platforms, visual media has become a significant component of everyone's life in the world. To better organize and understand the tremendous amount of visual data, computer vision and machine learning have become the key technologies to resolve such a huge problem. Among the topics in computer vision research, human activity analysis is one of the most challenging and promising areas. Human activity analysis is dedicated to detecting, recognizing, and understanding the context and meaning of human activities in visual media. This dissertation focuses on two aspects in human activity analysis: 1) how to utilize multi-modality approach, including depth sensors and traditional RGB cameras, for human action modeling. 2) How to utilize more advanced machine learning technologies, such as deep learning and sparse coding, to address more sophisticated problems such as attribute learning and automatic video captioning.

To explore the utilization of the depth cameras, we first present a depth camera-based image descriptor called histogram of 3D facets (H3DF) and its utilization in human action and hand gesture recognition and a holistic depth video representation for human actions. To unify both the inputs from depth cameras and RGB cameras, this dissertation first discusses a joint framework to model human affections from both facial expressions and body gestures with a multi-modality fusion framework. Then we present deep learning-based frameworks for human attribute learning and automatic video captioning tasks. Compared to human action detection recognition, automatic video captioning is more challenging because it includes complex language models and visual context. Extensive experiments have also been conducted on several public datasets to demonstrate that our proposed frameworks in this dissertation outperform the state-of-the-art approaches in this research area.

Acknowledgements

Firstly, I would like to thank my advisor, Professor Yingli Tian, for her continuous support and patient guidance during my PhD study. Professor Tian has opened the gate of computer vision research to me and helped me to find my research motivation with her immense knowledge and insightful advises. Prof. Yingli Tian has been a mentor in both my academic work and life, she always sincerely help me when I am encountered with troubles and frustrations.

Besides my advisor, I would also like to express my gratitude to the rest of my thesis committee: Dr. Lisa M. Brown, Prof. Ioannis Stamos, Prof. Jizhong Xiao, and Prof. Zhigang Zhu, for attending to my dissertation defense, as well as their insightful comments and valuable encouragements.

My sincere thanks also go to Dr. Amir Tamrakar and Dr. Shengyang Dai for the opportunities to join their teams in SRI International and Google Photos for internships. During the internships in their teams, they have generously shared their valuable experience in both academic research as well as industrial development with me.

In particular, I would also like to thank Dr. Jingen Liu and Prof. Xiaojie Guo for the discussions about research ideas and their valuable comments while we were working together.

I would also like to thank all my colleagues during my PhD study: Dr. Shizhi Chen, Dr. Chucai Yi, Dr. Xiaodong Yang, Ms. Yang Xian, Mr. Xuejian Rong, Mr. Yuancheng Ye, Dr. Zhi Liu, and Dr. Zisis Petrou. I really learned a lot during working together with them.

In addition, thanks for the sponsors that have supported my research: NSF grants IIS-0957016, IIS-1400802, EFRI-1137172, IIP-1343402, Microsoft Research, NIH 1R21EY020990, CITY SEEDs, PSC-CUNY Award 64720-0042, FHWA DTFH61-12-H-00002, and ARO Grant W911NF-09-1-0565.

Finally, my special appreciation goes to my beloved parents and my wife. They have been encouraging and supporting during these years and their selfless love will always give me support now and in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human activity analysis is a significant component in image and video understanding. The large visual variance as well as semantic ambiguity underlying this topic makes it a difficult task. Applying advanced feature engineering and machine learning models, researchers in computer vision can build automatic software systems to recognize activity categories in controlled environments, such as smart-home surveillance and video gaming interactions. Over the past decade, many research efforts have been made towards recognizing human actions from RGB videos [85, 84, 111, 72].

Recently, with the increasing applications of depth cameras in surveillance and human-computer interaction, multi-modality-based human action recognition using both RGB and depth information becomes more attractive to the success of many intelligent systems. Compared with solely using RGB channel, including depth cameras provide more geometric information such as human body size, shape and position, which is significantly important for action recognition. This dissertation addresses how to effectively utilize and combine the depth maps with RGB frames to address multiple human activity related tasks, such as hand gesture recognition, holistic human action recognition and affection recognition.

Compared with conventional hand-designed features, deep-learned features are advantageous not only because they need much less effort and domain knowledge to become more generic to different modalities, but also because of their potential to automatically learn an organized hierarchy of semantic features [174]. Although the deep learning network has been very successful in visual recognition, the deep features are usually treated as mid-level features [129], and function like signal filters, which affect the recognition performance and limit their applications. Therefore, inspired by [83], instead of directly mapping deep features onto action labels, we define a set of action attributes. These attributes can boost recognition and enable new applications such as zero-shot learning. As human bodies/joints are easier to track than open source videos in [83], we argue that action attributes are more appropriate for actions in depth videos. To our knowledge, our work is the first attempt to leverage "attributes" to recognize actions from depth videos.

Including [83], most existing attribute learning approaches tend to learn the attribute detectors independently. As a result, some detectors may learn the properties that do not belong yet correlate to the attribute of interest. In other words, they do not "learn the right thing" [55]. For example, the attribute detector "arm motion" may learn patterns related to "torso up-down motion" in action "jogging" because of their co-occurrence. It is believed that the semantic/geometric relationships among the attributes can serve as constrains during the attribute learning, and eventually enable the detectors to learn the exact human motions and postures. Therefore, we propose a joint attribute learning framework which leverages the relationships among attributes represented by a graph, as shown in Figure 6.1. The proposed algorithm incorporates the relation graph during the optimization of attribute detector learning. It tends to decorrelate attributes that are semantically distant, while enhance correlation of neighboring attribute detectors. This dissertation also includes a deep learning-based action attribute learning framework which embeds group sparsity among attribute groups. The group sparsity introduced to the learning process is proven to be effective in decorrelating the features for different attributes.

However, solely recognizing human activity from images or videos is not enough in providing more descriptive information about the human activity. To generate more informative and detailed descriptions about the human activity content in videos, a deep learning-based sequence-to-sequence framework for video captioning is proposed in this dissertation. The proposed framework is based on recurrent neural networks and long short-term memory cells which are effective to model sequential signals. The application of this framework on automatic American Sign Language recognition is also discussed in this dissertation. To accomplish this task, we collected about $20,000$ video clips from YouTube website containing ASL signing and captions and trained a network to learn the correspondences between the video and text.

Overall, the contributions of this dissertation are summarized as following:

- Two types of novel and effective depth map-based image descriptors are proposed (a local one and a holistic one). Their applications include both human action and hand gesture recognition tasks. The proposed descriptors show superiorities over a lot of related approaches on a variety of benchmark datasets.

- A combinational multi-modality recognition framework is proposed to handle human affection recognition from two perspectives: facial expressions and body gestures. The proposed framework utilize both depth and RGB data modalities and jointly handle variances introduced by subjective differences and is demonstrated effective on human affection recognition. In addition, a novel multi-modality dataset is proposed.

- To further investigate how semantic information can affect human action recognition, the cutting-edge deep learning techniques and a semantic graph are combined to jointly learn mid-level human action attributes from depth data. The study on human action attributes bridges the gap between features such as motion traits and action categories.

- Beyond simply learning the mapping from the visual features to a discrete set of action or attribute labels, we further propose a combinational framework to jointly model the sequential information in input videos describing human activities and describe the visual contents with human-level English sentences.

The rest of the dissertation is organized as following. The previous related research work are reviewed in Chapter 2. Then a two depth camera-based descriptors for human action and hand gesture recognition are introduced in Chapters 3 and 4. A multi-modality human affection recognition framework is presented in Chapter 5. Deep learning based action attribute learning framework is introduced in Chapter 6 and automatic video description method is presented in Chapter 7. Finally, the dissertation is summarized in Chapter 8 and the future direction of our research is also elaborated.

Chapter 2

Related Work

In this chapter, the context of the dissertation is discussed by reviewing the related work. Firstly, both RGB-based and depth-based human action analysis frameworks are discussed. Secondly, multi-modality-related frameworks as well as their application areas in human activity analysis are reviewed. Thirdly, the recent developments and progress of deep learning techniques are reviewed.

2.1 Human Activity Recognition

2.1.1 Overview

Human activity analysis is a critical component in many important applications, such as video surveillance systems and intelligent household robots. However, recognizing human activities from visual signals, mainly video sequences, is a challenging task due to many issues such as scaling, rotation, occlusion, background clutter, *etc*. In this section, the related work is firstly reviewed and categorized based on their signal modalities, RGB or Depth-based. Secondly, a mid-level representation: action attribute, and its related methods are briefly reviewed.

2.1.2 RGB Camera-based Human Action Recognition

The RGB or gray scale-based human action recognition has been studied more than two decades, which forms the majority body of human action recognition research.

A popular paradigm for action recognition is to firstly extract visual feature descriptors and then apply a final classification such as a multi-class SVM classifier. Space-time approaches tackle the feature representation problem by considering a human action as a 3D space-time volume. Then the 3D volume is represented by a set of space-time features [31, 120, 56] or trajectories [77, 142, 92]. In [31], the authors firstly tracked figure-centric image patch sequences from input videos and represented the sequences using optical flow-based motion descriptors. Then a nearest-neighbor classifier was applied to retrieve action labels. In [120], the motion patterns in human actions were represented by space-time features [71] corresponding to moving 2D image structures at moments of non-constant motions. Then a SVM classifier is applied for classification. In addition, the concept of "histogram of interest points" was introduced as an action descriptor in [120]. The authors in [56] proposed a hierarchical approach to model the input video into a set of feature descriptors ranking by their complexities.

In [92], trajectory snippets of interest points were extracted by KLT feature tracking algorithm [89] and quantized to build a visual codebook for action representation. Similarities among actions were also modeled with angles between associated subspaces in a trajectory system [77]. Sliding window technique was also exploited in [142] to relax the tight constraints of bounding box-based tracking, which resulted to a scale and shape invariant spatial-temporal action descriptor.

Another perspective of human activities is to treat them as a stochastically predictable sequence of states [116, 156, 133]. Therefore, many stochastic techniques such as hidden Markov model (HMMs) and hidden conditional random fields (h-CRFs) [68] were applied to inference useful traits in human activities. In [116], the authors treated the whole video as a sequence of actions. Action recognition was achieved by probabilistic search of feature vectors composed of location, speed and visual descriptors. HMMs were exploited to perform human behavior analysis by smoothing the detected actions. Wang *et al.* [156] proposed a part-based approach for human action recognition from video sequences using motion features. The human actions were modeled by a flexible constellation of parts conditioned on image observations. Then h-CRFs were applied for action recognition. Instead of using a holistic representation of the whole video, the authors of [133] described each video sequence as a set of short clips corresponding to a latent variable in an HMM model. Besides single-person action or event recognition, stochastic approaches were also proven to be useful in activity recognition tasks of multiple subjects, such as group activities[17] and human interactions [64]. Stochastic approaches are useful when the action categories have complex temporal structures.

Other related human action recognition methods include rule-based methods [12, 83] and shape-based methods. The rules were modeled as graphs in [12], where the authors sought to pose activity detection as a maximum-weight connected subgraph problem over a learned space-time graph. The optimal subgraph that maximized the activity classification score was interpreted to find both the probable action label as well as the spatial-temporal position. In [83], mid-level semantic labels named "action attributes" were proposed to model the relationships among different actions. Action recognition was performed by pre-defined rules of action label combinations. However, the relationships among attributes were not modeled in [83]. In this dissertation, a framework to jointly learn action attributes guided by their relationship graphs will proposed in Chapter 6.

As for shape-based methods, human body parts are usually modeled as rectangular patches in 2D space or volumetric spaces in 3D space. For example, Thurau *et al.* [139] modeled a set of pre-defined human-shape templates as "pose primitives" and the action recognition was conducted based on the template matching. Instead of utilizing holistic human shapes, Ikizler *et al.* [53] proposed to model human bodies as sets of 2D rectangles. Holistic representations were then carried out by computing histograms of the body parts.

2.1.3 Depth Camera-based Human Action Recognition

Due to the explicit 3D structure representation of objects and human body parts from depth maps, there are more and more research efforts invested in depth map-based human activity recognition. This is especially true since the release of low-cost 3D sensors (*e.g.* Microsoft Kinect) and associated software development kits (*e.g.* Microsoft Kinect SDK) and the success in real-time body joints position estimation [125]. Action recognition from depth sequences can be roughly categorized into two groups, depth map-based methods [79, 103, 155], and joint based methods [163, 147, 173, 153], which are based on a skeleton joint estimator.

Depth map-based Methods. Early research has focused on applying existing 2D image representations on 3D depth data, such as bag of 3D points by Li *et al.* [79], which sampled representative 3D cloud points from depth maps for action recognition; histogram of 3D gradient orientations (derived from histogram of orientation gradients (HOG) [22]); and extending 2D interest point detectors to depth maps [44]. In our previous work [169], projections of 3D depth maps onto three 2D orthogonal planes were stacked as three depth motion maps, and then HOG descriptors were computed from the depth motion maps as the global representations of human actions. This method transfers a sequence of 3D depth maps to a 2D image that is further treated as a gray image without explicitly encoding 3D shape information. Recently, researchers have paid more attention to intrinsic features from depth images. Surface-normal, as a natural and explicit description of a local 3D volume, has been used in depth image descriptors [103] [137] and graphics [108], and has demonstrated its potentials in both activity recognition and object recognition.

Joint-based Methods. Another branch of depth video-based action recognition methods focuses on a pre-detected set of skeleton joints such as "head", "torso", "leg", *etc.* Compared to directly using depth maps, skeleton joints are more compact and abstract. In [102], the authors described actions by an affinity matrix between joint angle features. Xia *et al.* [163] modeled the action representations by histograms of 3D joint locations and then applied discrete HMMs for classification. In [153], the authors exploited a hierarchical structure of body parts composed by different sets of skeleton joints which can explicitly model body-part based movements and relationships among different body parts. 3D positions were also exploited in [173] to form a kinematic-based descriptor called "moving pose", which also encoded velocity and acceleration information. In one of our previous work [176], both motion and structure features were computed from 3D positions of joints by composition of different pair-wise location offsets among joints. In [167], the authors extended this idea by adding a layer of principle component analysis (PCA) for obtain a more compact and efficient descriptor.

2.1.4 Action Attribute Recognition

As mid-level semantic features, attributes serve as important components in image-based visual recognition tasks [69, 34]. The utilization of attributes can enable several new tasks such as zero-shot learning [54] and transfer learning [66]. This idea was extended to action recognition tasks [83, 170, 80], demonstrating that action attributes are useful for zero-shot action recognition and improving the performance of action recognition. More specifically, the authors in [170] integrated both action attributes (poselets) and object parts to model the interactive actions between human and objects. More recently, in [178], the authors showed promising results of deep learning in human attribute classification. The relationships among attributes are often ignored in attribute learning, which may result in learning the correlated yet wrong properties. In [55], the authors proposed to decorrelate the attributes by grouping object attributes into disjoint groups to eliminate the ambiguity. However, simply grouping attributes is inadequate to model the complexness of action attributes. Therefore, in this dissertation, a novel joint attribute learning algorithm which integrates an undirected graph to preserve the complex action semantics in Chapter 6.

2.2 Multi-modality Human Activity Analysis

2.2.1 Overview

An activity can be described by more than one types of features which convey more informative signals. Therefore, mutli-modality fusion is also a very active research direction in human activity analysis. Despite from audio-visual analysis, multi-modality frameworks have shown its effectiveness in many application areas. In this section, the multi-modality methods are reviewed in three categories based on its application area: 1) human affection recognition, 2) behavioral and social networking-based activity analysis, and 3) American sign language (ASL) recognition.

2.2.2 Human Affection Recognition

Human affection is the core component to understand the relationship between emotional states and human activities. It is capable to shed light on the intrinsic logic of human activities and to predict how one reacts to others [109]. In [87], the authors argued that single modality is not enough for affective computing, therefore they proposed to combine textual information with other visual features to identify the affection in a static image. A new classification diagram named "joint h-CRF" was proposed in [126], where four affective dimensions were analyzed (namely, "activation", "expectancy", "power", and "valence") and the proposed new diagram can take advantage of the multi-modality data.

Besides the methods discussed above, human affection also includes several other core modalities such as hand gestures, facial expressions, physiological changes, speech and other activities [104]. In this section, we mainly focus on reviewing the hand gesture and facial expression modalities of them and several main fusion mechanisms as following.

Hand gesture recognition. Hand gestures serve as a significant modality of human activity analysis as well affective computing tasks. Hand gestures convey important information that covers multiple function categories in communication including conversational gesture, controlling gestures, manipulative gesture, and communicative gestures [161]. As a first step of hand gesture recognition, hand detection and tracking is either done by skin color or shape-based segmentation, which can be inferred from the given RGB images [38]; or directly resolved by leveraging the depth information [61] [81]. Based on detection and tracking of hand regions, both dynamic and semantic features are extracted and utilized for gesture recognition [161]. Because of its intrinsic vulnerability to background clutter and illumination variation, RGB-channel-based hand gesture recognition usually requires a clean background, which limits its application. Van den Bergh and Van Gool *et al.* [8] successfully used a Time of Flight (ToF) camera combined with RGB camera to recognize four simple hand gestures in an HCI application by simply using small depth patches of hands. In [112], Ren *et al.* employed a template matching-based approach and recognized hand gestures using a histogram distance metric of Finger-Earth Mover's Distance (FEMD) with near-convex estimation [113]. However, this method only considered the contours of fingers while ignoring the palm region (which also provides important information for complex hand gestures.) Pugeault and Bowden [110] employed Gabor Filter features at different scales and orientations to recognize characters in American Sign Language (ASL).

Facial expression recognition. Another very informative channel for human activity analysis is the facial expression. Other than body motions or gestures, facial expression is one of the non-verbal communication methods which convey the mood or mental state of a person (*e.g.*, happy, sad, fear, anger *etc.*) The early facial expression analysis frameworks are based on facial action coding system (FACS) [39], which has been treated as the foundation of facial expression description. The FACS contains 44 action units (AUs) and each of which is reserved for a different facial action and has three or five levels of magnitude. The Facial expression methods can be roughly grouped into two categories: deformation-based [21, 50, 63] and motion-based [140, 3, 4, 165]. More specifically, Gabor filters and PCA were applied to extract facial features in [21] and in [50] the authors proposed a point distribution model to capture face deformations. Authors in [63] built up a 3D geometric face model for representation of human faces. In [140], the authors divided the face patch into upper and lower parts. Then several geometry traits were utilized on each part for feature extraction. Neural networks were trained to compute AU confidence scores. Dense optical flow fields was utilized to capture facial movements in [3]. Similar to [63], a 3D model was built from face but it was designed to represent 3D movements of human faces. In [165], the authors proposed to model nonrigit local facial motions by adapting local motion models. In addition, the local models can also be used for parametrize the shape of eyes, noses, *etc.* In addition, recent facial expression recognition frameworks are based on local features such as local binary patterns (LBPs) [124] or sparse PCA [99].

Modality fusion methods. The fusion methods of multiple modalities can be categorized into three groups: 1) shallow fusion and 2) mid-level fusion. Shallow fusion is the most common method by simply either concatenating features from different modalities or averaging over multiple classification scores by some pre-defined priors. Feature-level fusion is effective in combining different type of information together as a holistic feature vector [135]. One drawback of this fusion type is that the fusion mechanism is not controlled and follows a hidden mechanism determined by subsequent learning algorithms. Another drawback is that the feature dimensions of different feature types can result in imbalance and undesired results. Decision level fusion is superior that it can be computed in parallel, but one has to tune the weights among different modalities. Another type of fusion is to transferring between feature spaces, such as canonical correlation analysis (CCA) [45], which enable feature vectors matching between different modalities.

2.2.3 Behavioral and Social networking-based Activity Analysis

Human behavior is a complex association of human personalities and psychological states. Its application could be human identification in video surveillance systems or human-computer interaction systems. Effort has been made in recognizing human emotions [95, 119, 94, 13]. Defining emotional attributes for multi-modality dyadic interactions in the data annotation level has been studied in [95]. Audio information was also integrated in hybrid frameworks as in [119] and [94], where the former method studied how to synchronize the multi-modality features and the latter one focused on combining audio information with facial expressions via a GMM-based probability model. In [13], the emotion recognition system reached real-time by modeling 3D facial features with random forests. More complex human activities have been studied in [160] for audio-video combination by late-fusion of multiple classifiers.

As for more complex relationship among a group of people, research effort has also been invested in social networking-based activity analysis [35, 70]. In [35], the social interaction types were recognized by inferencing different social roles of different people. Their social interactions were estimated by their locations and the orientations of their faces. There were three types of interactions been discovered: dialog, discussion and monolog. As a result, the algorithm would generate a social network among all detected persons. The authors in [70] studied social interactive activities in sports by jointly recognizing both low-level actions and high-level events. Other social activity recognition systems include recognizing abnormal behaviors in group activities [20], attribute-based social activities [40], functional positions of sport players [88] and social interactions in TV shows and movies [91, 48].

2.2.4 American Sign Language Recognition

As an important sign language in the world, ASL is used by deaf people across U.S. and Canada. Some researchers have estimated that the population using ASL as a primary language was about $500,000$ [58]. In automatic ASL recognition, early attempts have been made to explore the use of Hidden Markov Models (HMMs) in sequence modeling [152, 132]. ASL intrinsically covers multiple modalities such as hand gestures, body movements and facial expressions. In [100], the authors proposed a Hough transform-based hand gesture recognition system for ASL recognition. Neural networks were applied for classification. In [96] and [86], the authors proposed to combine facial expression recognition with hand gestures by tracking varies facial landmarks in ASL videos. In [29], the authors proposed a combinational and continuous speech recognition framework for sign language. The facial and hand gesture features were extracted by both head and hand tracking algorithms. In addition, n-gram language models were also applied for sign speech recognition. In recent years, since the progress in commercial multi-modality sensors, researchers have been focusing on exploring the utilizations of multiple sensors. For example, in [110] and [37], the authors proposed to employ Kinect and Leap Motion sensors, respectively, for real-time hand-gesture-based ASL recognition. Attempts have also been made towards educational software to automatically recognize ASL from video for students learning the language [52]. In this dissertation, we propose to study ASL recognition from the perspective of data-driven video captioning. To the best of our knowledge, this is the first time ASL recognition is combined with video captioning.

2.3 Deep Learning in Human Activity Analysis

2.3.1 Overview

In recent years, due to the rapid growth of computing and data communicating capacities of computers, artificial neural networks have won numerous contests in the field of pattern recognition and machine learning. Besides, these techniques has also shown their great potential in many application areas, such as hand-written digit recognition [46], object recognition [76], scene understanding [27, 130], and action recognition [57]. In particular, deep architectures of neural networks are the most favored not only because their superior performance but also their intrinsic hierarchical structures which could learn meaningful features. In this section, we mainly focus on reviewing deep learning techniques in activity-related topics.

2.3.2 Deep Learning in Action Recognition

Deep Convolutional Neural Network (CNN) has been applied in video classification [60] and action recognition [74]. In [74], learned spatio-temporal features from video sequences using independent subspace analysis achieved the state-of-the-art performances on several benchmark datasets. Combined with the exploration by Zeiler *et al.* [174], the deep-learned features demonstrate desirable properties such as increasing invariance and class discrimination with ascending layers. In [57], the authors proposed 3D CNN models to capture the temporal information inside multiple adjacent frames and the developed model was demonstrated to be able to recognize human actions in real-world environments. Similarly, the authors in [143] also proposed to learn action recognition-oriented features form a large set of labeled human action videos. In [129], the authors explored multi-modality fusion with deep learning by training two streams of input videos (one regular stream and one optical flow-based stream) for action recognition.

2.3.3 Deep learning in video captioning

Deep-learning based visual content captioning is originated from machine-translation and is first applied in image captioning [90, 59, 151, 164, 28, 14] due to the astonishing success of RNNs with LSTM cells [158, 49]. Many of these approaches took two steps: 1) firstly the input image was encoded using a deep network, often pre-trained from a large dataset (such as ImageNet [23]) and then 2) the encoded vector was fed into a RNN to output a sequence of encoded words, which formed an output sentence. In [59], the authors tackled the dense captioning problem by combining Region-CNN and the previous mentioned frameworks to generate rich descriptions for image regions. The success in image captioning provides many useful schemes for video captioning, such as attention model [164].

Similar to image captioning, RNNs were also employed to video description [118, 172, 149, 148, 118, 117]. In [172], the authors treated a video as a sequence of image frames and exploited the temporal structure of them in both feature extraction level and pooling level. In feature extraction level, they employed CNN with 3D convolutional filters to capture local temporal structures; in the pooling level, a soft-attention model was employed to capture the information of frame order, which is potentially capable to capture the global temporal structure. Similarly, in [148], the authors directly delegated the responsibility of learning temporal structure to the LSTMs. Particularly, they followed a more sophisticated sequence-to-sequence scheme which was used in machine translation area [136]. Additionally, attention models were also applied in [164] to learn a weighting function over sampled key-frames. The temporal-aware model was demonstrated to capture more temporal dynamics than pure average pooling. In this dissertation, a novel sequential-modeling framework for human activity-oriented video captioning framework will be described in Chapter 7. Our proposed framework utilizes two separated streams of networks to handle different modalities. Additionally, applying such a separate model can enable us to conveniently combine multiple channels of input instead of raw-feature concatenation [164] or late score fusion [148].

Chapter 3

H3DF: A Local Depth Descriptor for Action and Hand Gesture Recognition

In this chapter, a local depth-based image descriptor and its applications in both human action and hand gesture recognition are described.

The recent successful commercialization of depth sensors has made it possible to effectively capture depth images in real time, and thus creates a new modality for many computer vision tasks including hand gesture recognition and activity analysis. Most existing depth descriptors simply encode depth information as intensities while ignoring the richer 3D stereo information. In this chapter, we propose a novel and effective descriptor, the Histogram of 3D Facets (H3DF), to explicitly encode the 3D shape information from depth maps. A 3D facet associated with a 3D cloud point characterizes the 3D local support surface. By robust coding and circular pooling 3D facets from a depth map, the proposed H3DF descriptor can effectively represent both 3D shapes and structures of various depth maps. To address the recognition problems of dynamic actions and gestures, we further extend the proposed H3DF by combining it with an N-gram model and dynamic programming. We extensively evaluate the proposed descriptor on two public 3D static hand gesture datasets, one dynamic hand gesture dataset, and one popular 3D action recognition dataset. The recognition results outperform or are comparable with state-of-the-art performances.

3D shape representation is a significant component of object categorization and action recognition. Compared to 2D image-based appearance representation, 3D depth-map-based representation is not only invariant to lighting changes, but also very robust to viewpoint and pose changes. Therefore, the depth-map-based representation holds great promise for modeling physical-related attributes such as positions, poses, shapes, and scene contexts.

Over the last few years, the successful commercialization of a variety of depth sensors and corresponding development toolkits has made 3D shape information more accessible for objects as well as human activities. Research topics reformed by 3D depth maps have attracted more and more attention [154, 79] [121] [125]. RGBD cameras have demonstrated their capabilities to provide more information about object sizes, shapes, poses, and positions. Compared to research with traditional RGB cameras, research with depth cameras has significant advantages for capturing strong boundary clues and spatial layouts, especially in environments with cluttered backgrounds and large illumination changes. In particular, conventionally challenging tasks such as object segmentation and scene parsing [127] have become much easier with depth information involved. The depth sensors have also motivated recent research efforts to explore object and human gesture recognition by using 3D information [110] [112] [8]. However, these methods for 3D depth-map-based hand gesture recognition have only applied the existing 2D feature descriptors black to the depth images, such as Gabor Filter Bank [110] or contour matching [112].

In order to directly and effectively capture and encode 3D shape information using depth maps, in this chapter we propose a novel characteristic descriptor we call Histogram of 3D Facets (H3DF). In 3D depth maps, we define a 3D cloud point together with its surrounding points as a "3D Facet", which includes the informative local surface pattern surrounding the cloud point. We first model each facet using a small plane. Then we apply a spatial centric pooling strategy to organize the collection of facet planes using their normal orientations to describe the current region of interest (ROI), which forms the final H3DF descriptor. In our applications of hand gesture recognition and human activity recognition, a region of interest may be an image patch describing a hand gesture or a body part. To integrate the static depth map descriptor with temporal information in depth video sequences, we propose two approaches: 1) We approximate the depth video sequence as an ordered collection of a number of representative frames. The optimal collection of representative frames is selected by minimizing a sequential loss function defined by using only selected frames to represent the whole video using Dynamic Programming (DP). 2) We capture and represent the local temporal structure patterns via N-gram modeling. The N-gram model can be viewed as a collection of "visual word transitions," which is insensitive to different temporal structures caused by different execution rates.

Compared to existing depth-map descriptors, our proposed H3DF depth-map descriptor has three advantages: 1) it explicitly captures the 3D informative shape patterns conveyed by depth maps. 2) It applies a compact representation to describe a depth image compared to other 2D feature descriptors, *e.g.* Histogram of Orientated Gradients (HOG) [22]. 3) Compared to existing surface normal-based descriptors such as HONV [137] and HON4D [103], H3DF utilizes a circular grid for spatial pooling to encode more information such as shape and local depth patterns, which implicitly manifests the importance of the center part and makes the descriptor more robust to external contour deformations. By utilizing Dynamic Programming-based temporal segmentation and N-gram-based representation [11], we generate more robust representations for depth video sequences by combining H3DF with temporal structure information. We evaluate the proposed descriptor on two public datasets of hand gesture recognition: the NTU Hand Digits Dataset [112] and the ASL Finger Spelling Dataset [110]; one dynamic hand gesture data set: the MSR 3D Gesture Dataset [155], and one popular action recognition dataset: the MSRAction3D [79]. The recognition results on all the tasks demonstrate that our approach outperforms or is comparable to state-of-the-art methods.

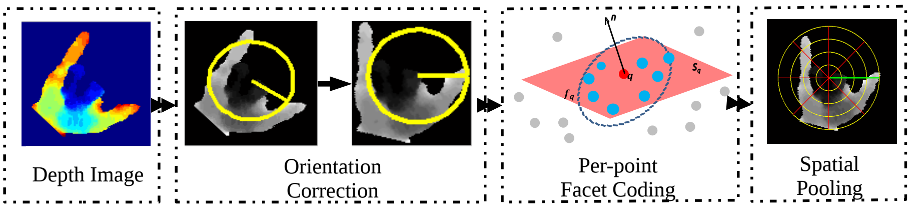


Figure 3.1: Pipeline of the proposed Histogram of 3D-Facets (H3DF) modeling for each depth frame. H3DF utilizes surface normals and centric spatial pooling together to encode a depth frame.

3.1 Histogram of 3D Facet (H3DF) Representation of Single Image

In this section, we describe the computation procedures of the new 3D feature descriptor, Histogram of 3D Facets (H3DF). The pipeline of H3DF representation from a single depth image is illustrated in the top row of Figure 3.1 (the bottom row will be explained in Section 3.2). Given a static depth image, we first delimit its in-plane rotation freedom by normalizing the dominant orientation of the depth image. Then for each 3D point in that image associated with its neighbor points (a 3D Facet), we compute the normal vector located in that point and then encode the normal vector to represent the current 3D Facet. The encoding is processed by projecting the normal vector onto three orthogonal planes (*i.e.* xy, yz, xz) and quantizing each projection. To generate a compact description of the whole image, we design a concentric spatial pooling to organize all encoded 3D Facets into a compact descriptor vector to capture the spatial layout and local structure of the depth image. In the following subsections, each step will be elaborated in detail.

3.1.1 Gradient-based Object Orientation Normalization

One challenge of hand gesture recognition is the large appearance variations when hand rotates. To make H3DF rotation invariant, we first conduct gradient-based orientation normalization for an input depth image or patch. For each depth patch as shown in Figure 3.2(a), the dominant orientation (denoted as θ) of the hand depth patch is first computed based on its shape and gradients. We then can rectify the 3D cloud points set (denoted as P) to obtain orientation-corrected 3D cloud point set P' of its salient orientation with the following equation:

$$P' = PR(\theta)^T, \tag{3.1}$$

where P and P' are $K \times 3$ matrices as the collection of K 3D points; $R(\theta)^T$ = $R(-\theta)$ represents an in-plane correction rotation matrix.

Let D be the depth image patch before orientation correction, we define a pixel-to-point mapping $I(\cdot)$, as it takes a 2D coordinate as input and output a 3D coordinate, where $P = I(D)$, and its inverse mapping $I^{-1}(\cdot)$, *vice versa*, where $D = I^{-1}(P)$. Together with Eq. (3.1), we have the corrected

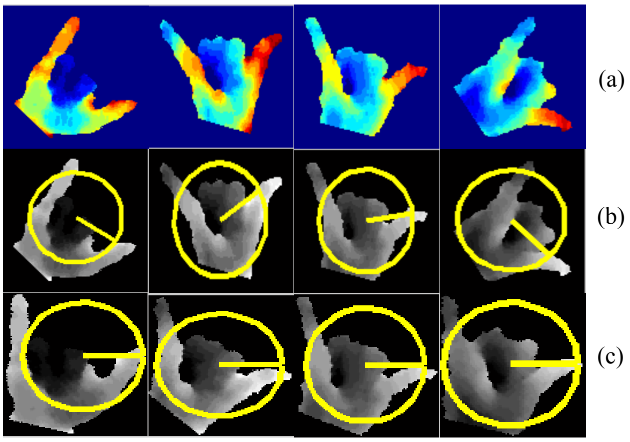


Figure 3.2: Examples of gradient-based orientation correction results of hand gestures. (a) Significant appearance variations of the same hand gesture when hand rotates. (b) Estimated dominant orientations are illustrated as yellow orientated circles. (c) Orientation normalized depth patches with removed appearance variations.

patch as:

$$D' = I^{-1}(I(D)R(\theta)^T), \tag{3.2}$$

which provides the orientation correction of a depth image patch of dominant orientation θ. As illustrated in Figure 3.2(a), depth images (D) of the same hand gesture may significantly vary due to rotation. Dominant orientations (see Figure 3.2(b)) can be detected based on *Gradient Consensus* to rectify the images to more similar corrected images (D') as shown in Figure 3.2 (c).

In order to estimate the dominant orientation θ and achieve in-plane rotation invariance, we compute the dominant depth gradient orientation as the normalization used by most local image descriptors [71]. A dominant orientation corresponds to the largest bin of the histogram of gradient angles, weighted by gradient magnitudes and smoothed by a Gaussian filter. As suggested in [71], each local maximum bin with a value above 80% of the largest bin is retained as well. Thus, each depth image might be associated with multiple orientations which are considered as multiple samples in our training set. As for a testing image with multiple dominant orientations, we choose only the key angle corresponding to the largest gradient angle bin. In this way, we ensure that the training set includes as much information as possible and that for each testing image there is only a single sample, to avoid decision ambiguity.

3.1.2 Defining a 3D Facet

To model a 3D object in a depth image, in addition to the outer contour, 3D surface properties and different shape patterns such as bumps and grooves provide rich and discriminative information. In some cases, the outer contour cannot be defined, and features inside the contour convey relative plentiful details.

Since these 3D surface details from depth information can be visualized as intensities in a gray image, it is natural to directly apply existing

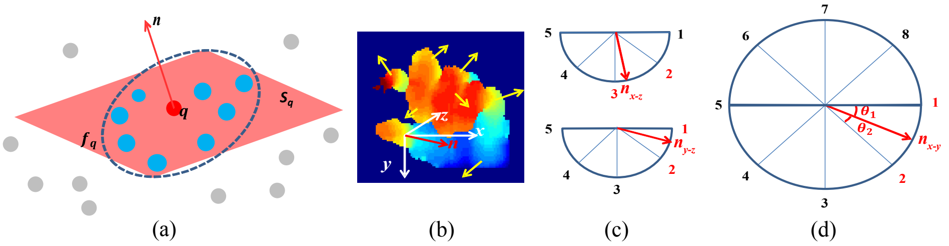


Figure 3.3: (a) Computing the 3D facet S_q of a cloud point q according to its neighbor cloud point set f_q. The pink plane is the fitted plane S_q and the blue region indicates the local constraint. The normal vector n is used as the representation of the 3D facet. (b) The normal vector n is encoded by projecting onto three orthogonal planes in (c) (xz and yz) and (d)(xy). As n_z is non-negative, the projected normal orientation ranges in xz and yz (c) are both $[0, \pi]$, but $[0, 2\pi]$ in the $x - y$ plane (d). A soft assignment strategy is employed to weight the two nearest orientation bins as shown in (d).

2D visual descriptors to obtain a compact representation. For example, the authors in [169] employed dense HOG to describe motion energy distributions from 2D motion maps which were generated from projections of 3D depth maps on three 2D orthogonal planes. However, this method used only the 2D information rather than explicitly modeling 3D surface details. In this chapter, we propose a novel 3D surface feature descriptor which can directly represent the rich information conveyed by 3D object surfaces.

3D Facets are used to model the shape details of a 3D surface, as shown in Figure 3.3. A 3D Facet associated with a cloud point q is determined by a local support surface defined by its surrounding cloud point set f_p:

$$f_p = \{q' | q, q' \in Q, \|q' - q\|_p \leq \sigma\}, \tag{3.3}$$

where σ is a threshold to control the size of the support region around the cloud point q, applying a locality constraint that only neighbor points can contribute to f_q. We then fit a plane S_q according to f_q such that the sum of distances between each point in f_q and the fitted plane is minimized. The normal vector $\mathbf{n}$ of a fitted plane S_q is then calculated as the representation of a 3D Facet. The normal fitting can be computed as a least-squares solution to the stack of N equations of the form $\mathbf{n}^T p_i = 1$ where N is the number of cloud points p_i in the 3D facet f_q. When we set N equal to 4, there is an analytical solution for the normal, which will be discussed in later sections.

Additionally, in Eq. (3.3), the parameters p together with threshold σ can jointly control the granularity of sampling surrounding points of q. In this work, we utilize two particular forms of them:

- (p, σ) = (1, 1): Bi-linear (analytical solution) or 4-neighbor (least-squares solution)
- (p, σ) = (inf, a): a × a patch. (least-squares solution)

In the first case, the difference between *"Bi-linear"* and *"4-neighbor"* is that the former one excludes the center point (*i.e.*, q) where the latter one does not. In the second case, the Chebyshev (l_∞) distance is used to define the supporting area as a patch in the corresponding 2D depth map. The difference of different selections of (p, σ) will be discussed in Section 3.1.4.

3.1.3 3D Facet Coding

Since a 3D Facet is defined as a plane, which is a subspace projected from a 3D space, it can be represented by using $[n_x, n_y, n_z, d]^T$, where the first three coefficients are the normal vector $\mathbf{n} = [n_x, n_y, n_z]^T$ of this plane and the forth attribute d is the Euclidean distance from origin point to the plane. Although all four coefficients are used to fix a plane, in this chapter we focus on the orientation rather than the distance of the plane, thus d is not coded and is highly dependent on the distance of an object to a camera. Therefore, a 3D Facet is only coded by its normal vector. The procedure of coding is angular-based using the orientation of each 3D Facet as illustrated in Figure 3.3 (b-d).

First, the normal vector (the vector $\mathbf{n}$ colored in red in Figure 3.3(b)) is projected into three orthogonal planes, *i.e.*, xy, yz, and xy planes as shown in Figure 3.3(c-d). Since the 3D point set is mapped from a 2D depth image, every cloud point corresponds to a pixel in the 2D depth image. Consequently, all the 3D points actually locate in front of the surface they formed (namely, the normal are pointing outward). So we can safely assert that all the normal vectors are pointing outward, in other words, their z-attributes are always non-negative.

Then, we evenly deploy m (for xz and yz planes) and n (for xy plane) bin centers on different planes. Each normal projection votes to two nearest bin centers (indices are colored red in Figure 3.3(c-d)). The benefit of this local soft assignment strategy over a hard assignment (in which each normal projection only votes to the nearest bin center) is that the loss of information can be significantly reduced and thus the coded feature vector is much more informative. The weights of each normal vector assigned to the two nearest bin centers are given as:

$$w_i = \frac{sin\theta_j}{\Sigma_k sin\theta_k}, i, j, k \in I_2, j \neq i, \tag{3.4}$$

where θ_i is the angular offset between the normal projection and the bin center indexed i. I_2 is the bin center indices that composed by two nearest bin centers (c_1, c_2). Therefore the encoded 3D Facet is represented as a vector of length $2m + n$, in which there are up to six non-zero elements.

3.1.4 3D Facet Pooling to Generate H3DF

Once all encoded 3D Facets are computed, we design a concentric spatial pooling scheme to group these 3D Facets from the image patch into a compact H3DF descriptor as shown in Figure 3.4. Another perspective of the proposed spatial pooling is to capture the information of facets arrangement coordinated in the center. In this phase, we address the boundary information as in [112].

For a spatial grid centered at (p_x, p_y), the bin index (a, b) of a pixel in the depth image $D(i, j)$ can be determined by the spatial distance $\|i - p_x, j - p_y\|_2$ and the angle $\arctan((j - p_y)/(i - p_x))$, where $a \in [1, A]$ and $b \in [1, B]$ and A, B are the spatial bin dimensions. Therefore, the dimension of the final H3DF descriptor of the image patch is $A \times B \times (2m + n)$.

The proposed pooling strategy is inspired by the invariant property of shape context in modeling rotations and scales of exterior contours [6].

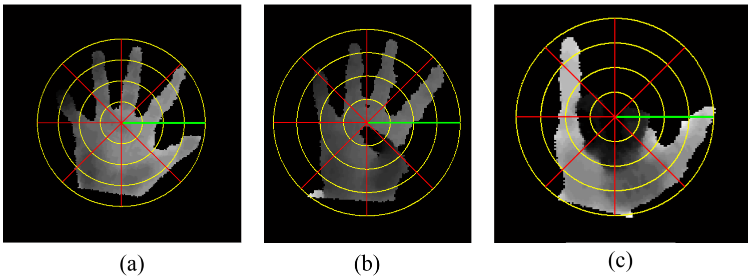


Figure 3.4: Illustration of the first phase spatial pooling for creating H3DF descriptors. The region of interest of the the depth image or patch is divided into 4×8 bins which are determined by both radial and angular offsets. (a), (b), and (c) are from three different hand gestures. Red line segments illustrate angular bin boundaries, yellow circles illustrate off-center radial distance bin boundaries, and green line segment shows the normalized patch orientation.

However, our usage of circular bins is beyond modeling exterior contours. Circular bins intrinsically put more weight in modeling interior parts of a depth object and thus it enables H3DF to capture more local depth patterns such as holes and bumps. Besides, bigger outer bins are capable to capture the shape information and robust to subtle shape variants. The usage of circular bins is a key difference between other surface normal-based descriptors [103] [137] by discriminating information intrinsically from interior parts and exterior parts of a depth object.

3.2 Video Sequence Representation using H3DF

In order to represent the temporal structure of depth video sequences by H3DF, we propose two approaches for coping with dynamic temporal structure information of the videos: Dynamic Programming-based (DP) representation and N-gram bag-of-phrases-based representation [11].

Traditionally, Temporal Pyramid (TP) is used to extend an image representation model (*e.g.*, Bag of Words) to represent a video sequence. However, TP is sensitive to time, speed, and state-composition variances within each video sequence. The phenomenon can be intuitively illustrated in Figure 3.6. In particular, if two time sequences share very similar contents but are not well aligned, they are far from each other in the metric space generated by temporal pyramid matching.

To overcome this issue and adapt H3DF to accommodate varied temporal structures, we propose two methods: 1) Dynamic Programming-based (DP) temporal segmentation to dynamically partition a video into cohesive sub-sequences and 2) N-gram bag-of-phrase-based representation.

3.2.1 Dynamic Programming-based Representation

The pipeline of DP-based representation is illustrated in Figure 3.5. Let $\mathbf{V} = \{vec(I_1), vec(I_2), ..., vec(I_t)\}$ be a sequential set of t frames with each frame I_i of dimension $M \times N$, *i.e.*, $vec(I_i) \in \mathbb{R}^d, I_i \in \mathbb{R}^{M \times N}$. A K-segmentation S of the video is a partition S of the frames into K non-overlapping contiguous segments, *i.e.*, $S = (s_1, ..., s_k), s.t. \bigcap_{i=1}^k s_i = \emptyset, \bigcup_{i=1}^k s_i = V$. The optimal

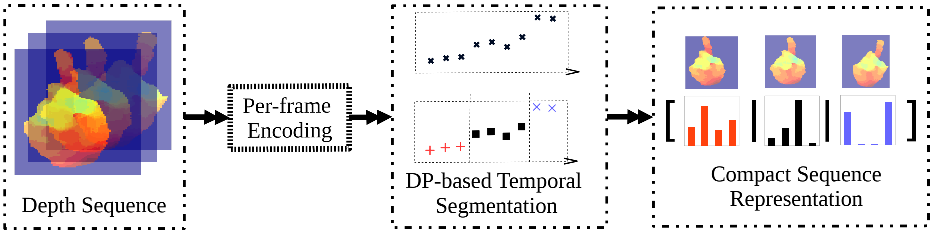

Depth Sequence
Per-frame Encoding
DP-based Temporal Segmentation
Compact Sequence Representation

Figure 3.5: Pipeline of the proposed DP-based video sequence representation example. DP-based temporal segmentation is used to partition each depth video sequence into a fixed number of segments, while the sum of within-segment intra-variances is minimized. A compact video representation is the concatenation of pooled H3DF codes of all segments.

segment $\hat{S}$ is defined as:

$$\hat{S} = argmin_S(\sum_{s \in S} \sum_{t \in s} \|t - \mu_s\|_2^2), \tag{3.5}$$

where μ_s is the mean of samples in each segment s.

This optimization problem is well-known to be efficiently solved by Dynamic Programming [5]. We implement the DP-based temporal segmentation in a recursive manner, as is detailed in Algorithm 1.

Algorithm 1: DP Temporal Segmentation, $(c, \hat{S})$ = DP_TS($\mathbf{V}$,K)

```
Input: Video sequence $\mathbf{V}$, number of partitions $K$
Output: Optimal Partitions $\hat{S}$, cost $c$
if $K$==$1$ then
    $\hat{S} = \emptyset$;
    $c = \sum_{v \in \mathbf{V}} \|v - \mu(\mathbf{V})\|_2^2$;
    return $\hat{S}$, $c$
end
$c = \infty$;
for $i \in \{1, ..., |\mathbf{V}| - 1\}$ do
    $(c_1, S_1)$ = DP_TS($\mathbf{V}(1 : i), K - 1$);
    $(c_2, S_2)$ = DP_TS($\mathbf{V}(i + 1 : end), 1$);
    if $c_1 + c_2 < c$ then
        $c = c_1 + c_2$;
        $\hat{S} = [S_1, i, S_j]$;
    end
end
return $c$, $\hat{S}$
```

This description is robust to dynamic warping of a video sequence. For example, as shown in Figure 3.6, since the initial hand gesture occupies 50% of total frames, the evenly TP-based method generally assigns a large weight to the initial pose. However, because the overall representative error is minimized (Eq. 3.5) in our proposed DP-based temporal segmentation, only the most representative frames are selected, while dynamically tuning the partition boundaries and thus the selected representatives are more

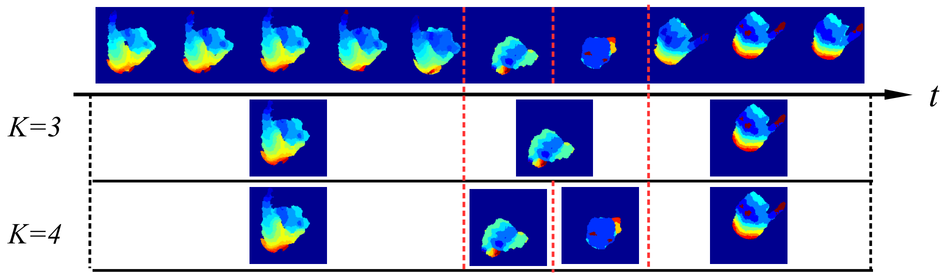


Figure 3.6: For a depth video sequence, Dynamic Programming-based temporal segmentation computes an optimal segmentation in terms of minimum representative error. We illustrate the idea with a dynamic American Sign Language (ASL) gesture for character *"j"* and two segmentations with number of segments, K, set to 3 (middle row) and 4 (bottom row) respectively. In particular, the DP-based segmentation is a better representation than the temporal pyramid since it can overcome the uneven gesture distribution, *e.g.* in the example case, initial pose occupies almost 50% of total frames.

informative and generic.

3.2.2 N-gram Bag-of-phrase based Representation

In the N-gram Bag-of-phrase model, instead of building a global representation of the whole temporal structure of a time sequence T, we attempt to discover local patterns of the time sequence. Using the same notation as in the previous section for DP, the time sequence $T = (t_1, ..., t_n)$ is characterized by its local N-grams, *i.e.*, the tuples constructed by every consecutive N signals. For example, if $N = 2$, the bi-grams of the time sequence are $\{(t_1, t_2), (t_2, t_3), ..., (t_{n-1}, t_n)\}$.

The N-gram model has been successfully used in speech recognition and natural language processing [11]. In computer vision, the N-gram model is used to generate Bag-of-phrases model [106] and is effective in image retrieval because it conveys more temporal information than the traditional Bag-of-words model. In our work, we propose to use the Bag-of-phrases model to represent video sequences, with each N-gram (a visual and phrase) describing a local pattern of the action. In particular, with $N = 2$, let $B = \{b_1, b_2, ..., b_t\}$ be the sequence of image (frame) representations, b_i is the Bag-of-visual-words representation of frame i, the sequence B is then modeled as a non-sequential set of tuples $\{(b_1, b_2), (b_2, b_3), ..., (b_{t-1}, b_t)\}$. Each tuple (b_i, b_{i+1}) is simply represented by their concatenation $[b_i^T, b_{i+1}^T]^T$. Then to fix the dimensions of representations of video sequences, we compute the codewords of the set of concatenations using Sparse Coding (the bag-of-phrase model works better with Sparse Coding than K-means in our experiments) and then use a max-pooling to generate a histogram of codewords for each video. A flowchart is shown in Figure 3.7.

3.3 Applications and Implementation Details

In this section, we introduce the applications of H3DF on hand gesture recognition and human action recognition and the implementation details of H3DF.

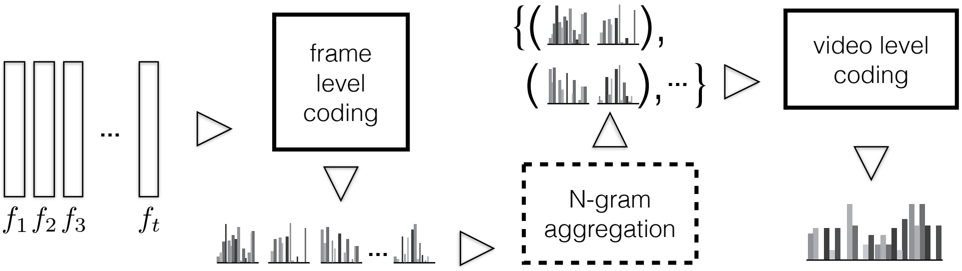


Figure 3.7: Illustration of a two-layer bag-of-phrases model for video description. Firstly, a bag-of-words model using K-means is used to generate a representation vector for each frame in the video (frame level coding). Secondly, a bag-of-phrases model using Sparse Coding is used to generate a representation vector for the video (video level coding). The final output is a histogram of N-gram codewords.

3.3.1 Hand Segmentation

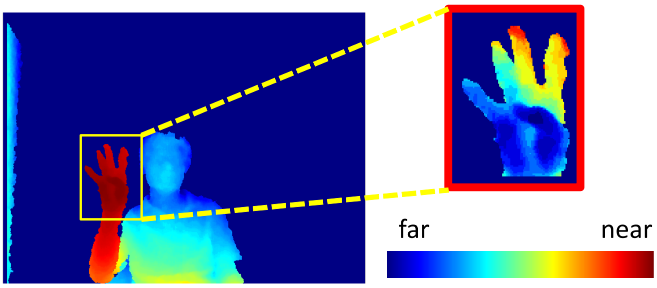


Figure 3.8: Illustration of segmenting a hand from a 480×640 depth image. Relative intensity in each pixel indicates its depth value.

In hand gesture recognition, if the hand region is not segmented, extracting hand region from background is necessary. One method is to retrieve hand joint using a pose estimator [121, 125] or a hand tracker [144] in the corresponding 2D color image. In a human-computer interaction setting, as in [110, 112], it is a reasonable to assume that the hand is always the most front body part facing to the camera. In our work, we inherit this assumption and use it to pre-process the 3D depth image to segment hand regions based on the depth information. As a special case, in [112], all the subjects wore a black hand-wrist band to obtain the accurate hand regions.

As displayed in Figure 3.8, we first select a cloud point with the shortest camera-object distance from the depth image and record its value as d_{near}, then threshold the depth image within the range of $[d_{near}, d_{near} + t]$, where t is the threshold of the distance of the hand region. In our experiments, we set $t = 100$ millimeters. Thus, the segmented hand region can be represented as a set of 3D points calculated by:

$$p = \{ \bigcup_{(i,j) \in Y \times X} (i, j, D_{i,j}) | D_{i,j} \leq d_{near} + t \}, \tag{3.6}$$

where X and Y indicate the image coordinates and D represents the depth image.

3.3.2 Pooling Center Selection

In both hand gesture representation and human action recognition, how to select the center point (p_x, p_y) for the hand or a body part is an essential step which can greatly affect the recognition. An ideal center point should be relatively stable for similar objects and robust to minor shape changes. One option is to use the centroid of the convex hull of a shape. We prefer to find a center on the object rather than on the background, while centroid cannot be ensured when the shape is neither convex nor near-convex (as shown in Figure 3.9). Therefore, we propose to use an interior center instead. The procedure of computing the interior center is as: first the depth map is transferred to a binary map(setting foreground pixels as 1 and background pixels as 0), second Euclidean Distance Transform [93] is applied on the binary map and the "brightest" point is selected as the interior center. The benefit of selecting the interior center rather than the centroid is that the center locates inside the boundary and the major part thus is more robust to minor shape changes such as extensions and branches.

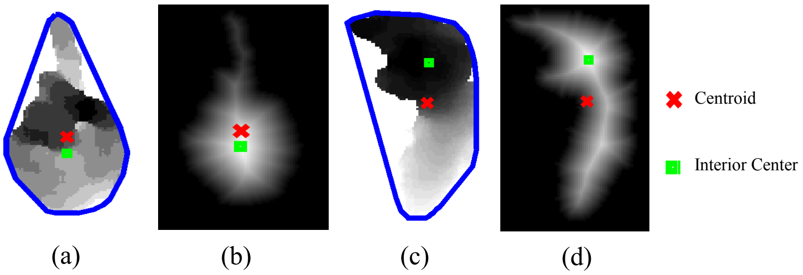


Figure 3.9: (a) and (c) are two depth images of hand gestures and associated convex hulls. (b) and (d) are the Euclidean Distance Transform maps of (a) and (c), respectively. For near-convex shapes such as in (a), the centroid and interior center are similar, but for non-convex shapes such as in (c), the interior center can be ensured to locate within the object and robust to extensions and branches, such as fingers. Brightness of pixels in distance maps (b,d) indicates the Euclidean distance from the nearest boundary pixel to the corresponding pixel locations.

3.3.3 Normal Estimation Methods

Here we discuss two estimation methods of the normal vector of a 3D Facet: bilinear estimation (analytical) and least-squares (plane-fitting) estimation. Bilinear normal estimation is suitable for a grid-organized 3D point set (or 2D depth image). Similar to bilinear interpolation, it takes the four neighbors and calculates the two orthogonal line segments that each connects two of them. Given the 3D Facet whose center is at $(i, j, d_{i,j})$, it computes a vector as the normal of this 3D Facet such that this vector is orthogonal to two line segments, one which connects points $(i-1, j, d_{i-1,j})$ and $(i+1, j, d_{i+1,j})$, while the other connects points $(i, j-1, d_{i,j-1})$ and $(i, j+1, d_{i,j+1})$. This approach is simple to implement and suitable for depth image calculation where 3D points are organized as gridded depth pixels. However, when considering 3D point clouds with non-uniform density, this approach will not work.

Plane fitting-based normal (least squares) estimation is more general and can be used in the situations where point density is non-uniform. It takes the center of a 3D Facet along with its neighbor points in a certain range, which we define as its local support surface. Then a plane is fitted using them. Despite its ability to generalize, there is a risk of losing detail when the size of the local support surface is enlarged.

3.3.4 Sparse Representation based Classification

To further explore the discriminative power of the proposed H3DF descriptor and its compatibility with different classification schemes, we apply two classification methods with H3DF to recognize hand gestures or human actions: Linear-SVM and Sparse Representation-based Classification (SRC), which is proposed by Write *et al.* [159] with good performance in face recognition. A brief review of SRC is provided as follows: given C as the set of class labels, we have $A = [A_{C_1}, A_{C_2}, ..., A_{C_c}]$ as the dictionary of training samples. In our approach, A is the matrix of vectored H3DF descriptors, *i.e.*, $A_{C_i \in C} = [vec(x_1^{C_i}), vec(x_2^{C_i}), ..., vec(x_n^{C_i})]$, where $x_j^{C_i}$ is the j^{th} H3DF vector of gesture or action class i. For a query descriptor y, the SRC via l_1-minimization is:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \|\alpha\|_1 \quad s.t. \quad \|y - A\alpha\|_2 \leq \lambda. \tag{3.7}$$

Therefore, the classification rule is:

$$identity(y) = \underset{C_i}{\operatorname{argmin}}\, r_{C_i}(y), \tag{3.8}$$

where the class-wise reconstruction residual $r_{C_i}(y)$ is computed as:

$$r_{C_i}(y) = \|y - A\delta_{C_i}(\hat{\alpha})\|_2, \tag{3.9}$$

where δ_{C_i} is the characteristic function that selects the coefficients associated with that class.

Runtime of H3DF. Computing an H3DF descriptor is fast. Without preprocessing, calculation of the H3DF for a 100 by 100 depth patch is about 2 ms with a Matlab implementation on one Intel Xeon Core (2.13 GHz). H3DF is thus feasible for use in real-time applications.

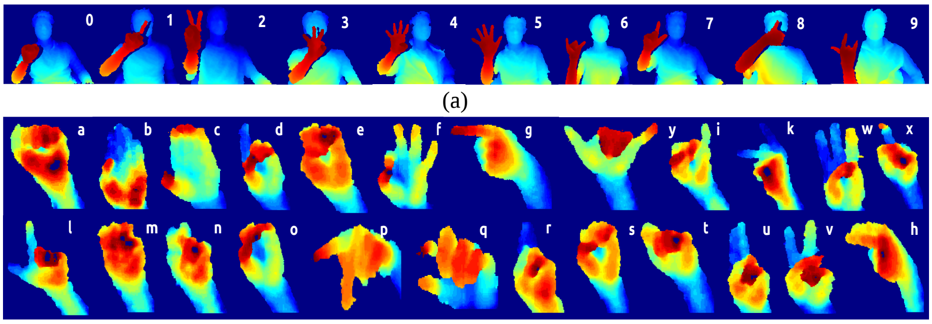
3.4 Experimental Results

In this section, we evaluate the proposed H3DF descriptor in two aspects: 1) static hand gesture recognition and 2) dynamic hand gesture and human action recognition.

3.4.1 Static Hand Gesture Recognition

Datasets and Experiment Set-up

For hand gesture recognition from static depth images, we employ two 3D datasets: the NTU Hand Digits Dataset [112] and the ASL Finger Spelling Dataset [110]. Both datasets were captured by a Kinect camera. The NTU Hand Digits Dataset [112] contains a total of 1000 depth images of 10 hand gestures of decimal digits $0 - 9$ from 10 subjects with 10 samples for each



(a)

(b)

Figure 3.10: (a) Sample depth images from the NTU Hand Digits Dataset for digits 0-9 [112]. (b) Sample depth images from the ASL Finger Spelling Dataset for English character from *"a"* to *"z"* (without *"j"* and *"z"*) [110].

gesture. The ASL Finger Spelling Dataset [110] captures hand gestures in 24 different categories, each of which represents one English character from *"a"* to *"z"* while *"j"* and *"z"* are excluded since these two characters are performed in ASL using motion. Compared with the NTU Hand Digits Dataset, this dataset is much larger, containing about $60,000$ depth images from 5 subjects. Unlike the NTU Hand Digits Dataset, the ASL Finger Spelling Dataset only provides segmented hand regions. Therefore the pre-processing step of hand segmentation as described in Section 3.3.1 is skipped. Some images of the datasets are shown in Figure 3.10.

For static hand gesture recognition, to explore the effect of subjective variance, we conduct two types of experiments. One is a subject-independent test, in which we use a "leave-one-out" strategy, *i.e.*, for a dataset with N subjects, we use $N-1$ subjects for training and the rest one subject for testing. This process is repeated for each subject and the averaged accuracy is reported as the overall accuracy. The other is a subject-dependent test in which all subjects appear in both the training and testing data, but no video appears in both training and testing.

Before comparison with the state-of-the-art approaches, we start by discussing the influences of 1) different approaches to estimate the normal of a 3D Facet, 2) different resolutions of extracted depth map, and 3) different numbers of grids while pooling encoded 3D Facet to generate the final descriptor. We discuss the issues using the NTU Hand Digits dataset [112].

Normal Estimation and Hand Patch Resolution

Here, we first analyze the influence of different resolutions of extracted depth maps as well as the robustness of proposed descriptors against resolution. We set different resolutions ranging from 150×150 to 25×25 for the normalized hand regions.

This experiment is conducted on the NTU Hand Digits Dataset [112]. As shown in Figure 3.11 (a), results in terms of overall classification accuracy of both leave-one-out subject-independent and subject-dependent tests are above 90%, which demonstrates the robustness of the proposed H3DF descriptor for different resolution of the normalized hand regions. Besides, as the resolution decreases, the performances are relatively stable, except in the case of 25×25 resolution. In all the following experiments of static hand

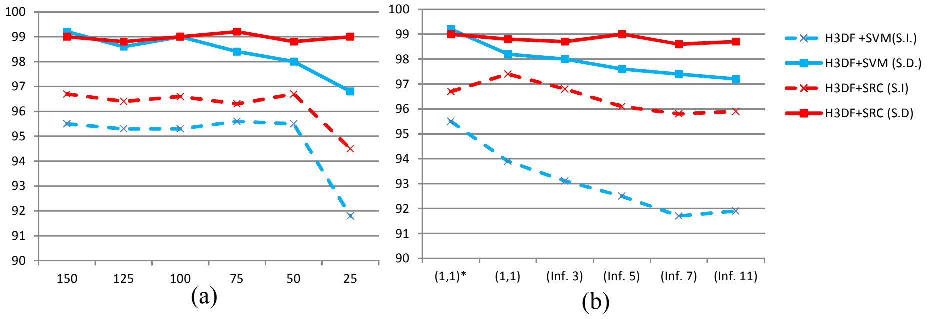


Figure 3.11: Accuracies of hand gesture recognition on the NTU Hand Digits Dataset [112] of (a) resolutions of hand-patches, and (b) different methods and parameters of normal estimation. Subject Independent (S.I.) and Subject Dependent (S.D.) accuracies of H3DF with both SVM and SRC [159] classifiers are shown.

gesture recognition, we use 150×150 as default patch size unless otherwise noted.

We also conduct an experiment on the NTU Hand Digits Dataset [112] to study the influence of different choices of normal estimation methods, as shown in Figure 3.11 (b). We compare the bilinear normal estimation method with plane fitting-based method of different patch sizes (In Figure 3.11 (b), $(1,1)^*$ indicates the analytical solution for normal computation.) As shown in Figure 3.11 (b), the analytical ($(1,1)^*$) approach performs best. For the plane-fitting approach with different sizes of local support surface, performances of both subject-dependent and subject-independent tests decrease and become stable when the size of the local support surface is greater than 7×7. This observation has demonstrated that for this particular problem, bilinear operator is more suitable and the proposed 3D descriptor favors more detail rather than less noise. Based on this observation, we use the bilinear estimation approach in all following experiments. We can also conclude from Figure 3.11 that subject-dependent tests perform better than subject-independent tests and are less affected by increases in the local support surface size. Additionally, proposed H3DF combined with sparse representation-based classification (SRC) [159] performs better than linear SVM. Thus, the default classifier is SRC in the rest of this chapter, unless otherwise noted.

Discussion of Pooling Granularity

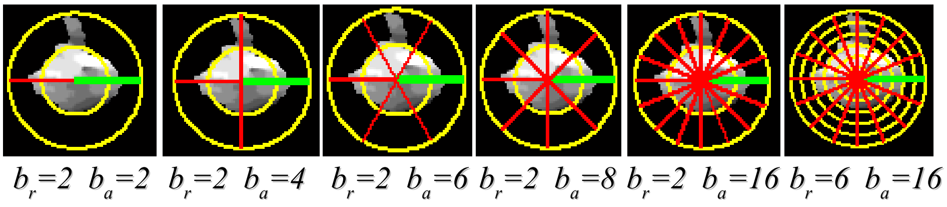


Figure 3.12: Illustration of pooling bins layouts with different radial bin (b_r) and angular bin (b_a) settings.

To explore how the pooling granularity affects the discriminative power of our proposed descriptor, we first conduct different settings of radial bin layouts (number of bins = b_r) and angular bin layouts (number of bins = b_a). Some of the pooling grids are illustrated in Figure 3.12. Since inside each cell of the pooling grid, the encoded facet vectors (dimension = 18) are pooled together by taking the average, thus the total dimension of the final H3DF descriptor is proportional to the product of $b_r \times b_a$.

The recognition accuracies are illustrated in Figure 3.13. We observe that low pooling granularity (from upper-left corner to bottom-right corner, pooling granularity increases) associates with relative low recognition accuracy. As granularity increases, recognition accuracy tends to increase and gradually reaches a stable value. We set a default of ($b_r = 4, b_a = 10$) unless otherwise noted because this is an appropriate trade-off between feature length and discriminative power based on our experiments.

Comparison with the State-of-the-arts

To compare our proposed H3DF feature descriptor with the benchmark methods as well as traditional 2D HOG descriptor on both datasets, we compare the H3DF with the Histogram of Gradients (HOG) on static hand gesture recognition. In our implementation of HOG, we evenly separate the normalized region of interest into 8×8 non-overlapping patches and each patch has 8 orientation bins. Thus the dimension of each HOG descriptor is 2046. We first evaluate our method on the NTU Hand Digits Dataset [112]. The average accuracies are shown in Table 3.1. Our method outperforms the benchmark method and the traditional 2D HOG descriptor for both subject-independent and subject-dependent tests. Compared with [112], our H3DF feature descriptor contains more information, such as folded thumb in palm than only contour information. Our method performs 3.5% higher than [112] and 4.3% higher than 2D HOG descriptor in the subject-independent test. As can be predicted, performances in subject-dependent test are much higher than in subject-independent test, where our method achieves 99.2% (H3DF+SVM) and 99.0% (H3DF+SRC) classification accuracy. A confusion matrix of our method in subject-independent test is shown in Figure 3.14. We observe that, for H3DF+SRC, all classes achieve accuracies higher than 90%, which demonstrate the effectiveness of our proposed 3D H3DF feature descriptor. Recently, classification results on this dataset are saturated (99% and 100% reported in [26]) via combining over three kinds of features which are specifically designed *for hand-shape only*. Since the proposed H3DF descriptor is a generic descriptor and can be used for multiple purposes such as action recognition and object recognition, we will not directly compare it with the fusion mechanism as proposed in [26].

Compared with the NTU Hand Digits Dataset [112], the ASL Finger Spelling Dataset [110] contains more complicated (24 gesture categories *vs.* 10 gesture categories) and realistic (all gestures are as in American Sign Language (ASL)) hand gestures. The ASL Finger Spelling Dataset is also much larger (over 60,000 images) than the NTU Hand Digits Dataset (1,000 images).

We follow the same experiment setting as previous stated. The class-wise accuracies of subject-independent test are shown in Figure 3.15 and

Table 3.1: Performance comparison of different methods on the NTU Hand Digits Dataset [112]. Best results are shown in bold.

Approach	Subj. Ind. Test	Subj. Dep. Test
Ren *et al.* [112]	93.9%	N/A
HOG [22]	93.1%	94.6%
H3DF+SVM	94.5%	**99.2**%
H3DF+SRC	**97.4**%	99.0%

Table 3.2: Performance comparison of different methods on the ASL Finger Spelling Dataset [110]. Best results are shown in bold.

Approach	Subj. Ind. Test	Subj. Dep. Test
Pugeault *et al.* [110]	49.0%	N/A
HOG [22]	65.4%	96.0%
Keskin *et al.* [61]	**84.3**%	97.8%
H3DF+SVM	73.3%	99.0%
H3DF+SRC	77.2%	99.9%
denseH3DF+SVM	83.8%	**100.0**%

the average accuracies of both subject-dependent and subject-independent tests are shown in Table 3.2. Our descriptor achieves 77.2% average accuracy in the subject-independent test, which significantly outperforms [110] with 28.2% higher accuracy, partially because we perform orientation correction before coding. Compared with the traditional 2D HOG descriptor, which is also with orientation correction, our method still achieves 11.8% higher accuracy and demonstrates the effectiveness of the proposed H3DF descriptor in describing 3D depth images than just applying an existing 2D descriptor. As shown in the confusion matrices (Figure 3.15), some ASL gestures are difficult to distinguish, such as *"p"* and *"q"*, where hand poses are almost the same and the only difference is the layout of two fingers (see Figure 3.10 (b) for hand gestures); another ambiguous pair of ASL gestures are *"m"* and *"n"*, which shares quite similar shapes.

To further explore the capability of the proposed H3DF as a local pattern descriptor, we combine the H3DF with dense sampling as used in DenseSIFT [146] with an evenly dense sampling grid at multiple scales (denseH3DF). In our experiment, we sample keypoints every 4×4 pixels at scales $\{8, 12, 16\}$. In each sampling keypoint, we compute the H3DF with radial bin number as 2 and angular bin number as 8. The local descriptor is then encoded using a soft vector quantization with a codebook of 1024 codewords computed from training set. For spatial pooling, we use a 4×4 spatial grid which partition the sampled points into 16 sets. Within each set, the sampled points (codes) are pooled using max pooling. Thus the resulting dimension of the feature vector is $4 \times 4 \times 1024 = 262,144$. We test denseH3DF using a linear SVM and the performance achieves 83.8% in subject independent test (Table 3.2), which is very close to the current best result obtained by Keskin *et al.* [61] (84.3%). However, [61] is specially designed only for hand poses, not a generic descriptor as H3DF is.

3.4.2 Dynamic Hand Gesture Recognition and Human Action Recognition

Datasets

To validate our H3DF descriptor together with DP-based temporal segmentation for dynamic hand gesture recognition from video sequences, we employ the MSR 3D Gesture dataset. This dataset contains 12 dynamic American Sign Language (ASL) gestures performed by 10 subjects. There is a total of 336 video sequences captured by a Kinect camera. The gesture categories cover ASL gesture signs such as *"Where"*, *"Store"*, *"Pig"*, *etc.* The hand region has been segmented. This dataset was collected by Wang *et al.* [155] and state-of-the-art performance has been demonstrated by Oreifej *et al.* [103]. We normalized each image along its height to 50 pixels for efficiency, while keeping the width/height ratio unchanged. We follow the same setting as in [103], which leaves one subject out for testing and trains on the rest and 10 repeats are processed to generate an averaged accuracy as the reported accuracy.

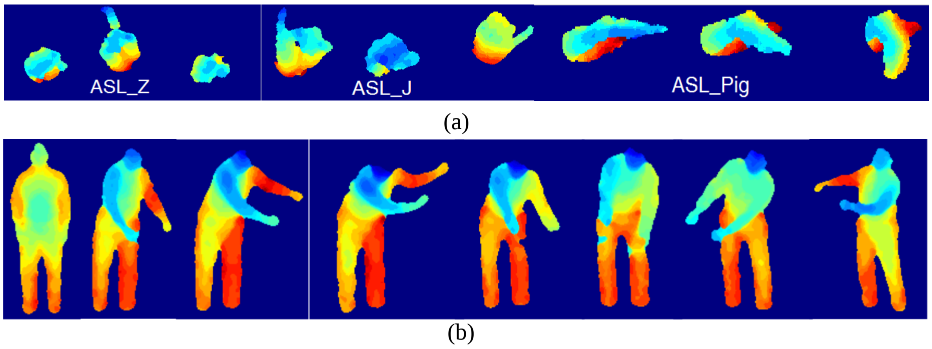


Figure 3.16: (a) Sample frames from the MSR 3D Gesture dataset for dynamic ASL hand gesture recognition. (b) Sample frames of action *"Golf Swing"*from the MSRAction3D dataset [79].

To further investigate how well our proposed descriptor can cope with more complex spatial-temporal feature descriptions, we also evaluated the H3DF for human action recognition using a very popular benchmark, the MSRAction3D dataset [79], and compare its performance with existing state-of-the-art methods. The MSRAction3D dataset includes 20 action categories such as *"high arm wave"*, *"hand catch"*, *etc*, which are performed by 10 subjects facing the camera. Each subject performed each action two or three times. The actions in this dataset capture a variety of motions related to arms, legs, torso, and their combinations. Several samples from mentioned datasets are shown in Figure 3.16.

Discussion of Pooling Granularity

Before comparing proposed H3DF descriptor with others on these two datasets, We first conduct experiments to investigate both spatial and temporal pooling granularity on MSR 3D Gesture Dataset. The experiment settings for spatial pooling granularity are the same as in Section 3.4.1 and the temporal segments number (K) is set to 5 for consistency. The results are shown in

Figure 3.17, we can observe similar patterns as in Figure 3.13, which again validate our default settings for r_a and r_b. A second issue is how temporal pooling granularity affects the recognition accuracy of dynamic gesture recognition. We compare the proposed dynamic programming-based temporal segmentation with traditional evenly partitioning of different numbers of temporal segments (K), the accuracies are shown in Figure 3.18. We observe that as K increases, more complementary information is modeled which results in higher accuracies. We further observe that 5 is a good selection for K; because normally *"neutral"*, *"on-set"*, *"peak"*, *"off-set"* and *"neutral"* is a general sequence of action. Dynamic partitioning is consistently a better strategy than even partitioning (except for $K = 4$) because dynamic partitioning is more robust to variance in time sequences due to its invariance to action execution rate.

Comparison with the State-of-the-arts

Dynamic Gesture Recognition. We further evaluate the proposed H3DF descriptor together with DP-based temporal partitioning in the application of dynamic hand gesture recognition on the MSR 3D Gesture Dataset [155]. We compare our proposed descriptor with several state-of-the-art algorithms for dynamic hand gesture representation such as the Histogram of 3D Gradient Orientations (klaser2008spatio) [62] and Histogram of 4D normals (HON4D) [103] which combines surface normals and Fourier transforms to represent spatial-temporal 4D volumes. As shown in Table 3.3, our framework (DP-H3DF+SRC) outperforms all previous methods (the best recognition rate of our method on the MSR 3D Gesture dataset is $\mathbf{95.6}\%$ with a different pooling grid setting, but to be consistent, we report the performance with default grid setting here).

Table 3.3: Performance comparison of different methods on the MSR 3D Gesture Dataset [155]. Best results are shown in bold.

Approach	Avg. Recognition Rate
H3GO [62]	85.23%
ROP [155]	88.50%
DMM [169]	89.20%
HON4D [103]	92.45%
DP-H3DF	$\mathbf{95.00}\%$

Human Action Recognition. We also evaluate the proposed H3DF descriptor in the application of human action recognition from depth sequences on the MSRAction3D [79] and compare it with existing state-of-the-art methods. Since the actions are not constraint to hand gestures in this dataset, instead of extracting hand patches, we compute H3DF around each skeleton joint and use a codebook with 3000 codewords to encode each H3DF. Then each frame is represented by the max-pooled histogram of all H3DF to generate a Bag-of-word representation of that frame. Next we use Bi-gram representations as discussed in Section 3.2.2 to obtain a set of Bi-grams for each video sequence. Finally, sparse coding is employed to generate a sparse histogram for each video using a dictionary with 1024 basis bi-grams trained.

Our proposed method has decent performance compared with state-of-the-art methods (Table 3.4) and achieves comparable performance with the best results [153]. As described in previous sections, our main goal is to propose a generic depth descriptor to handle both static and dynamic gesture and human action recognition. In both dynamic hand gesture and human action recognition, our method achieves better performance than other counterparts.

Table 3.4: Performance comparison of different methods on human action recognition of the MSRAction3D Dataset [79].

Approach	Avg. Recognition Rate
Bag of 3D Points [79]	74.70%
HOJ3D [163]	79.00%
STOP [150]	84.80%
ROP [155]	86.50%
Actionlet [154]	88.20%
DMM [169]	88.73%
HON4D [103]	88.89%
DSTIP [162]	89.30%
Pose Set [153]	90.00%
Proposed Method	89.45%

3.5 Discussion

In this chapter, a novel depth-based descriptor is proposed for both hand gesture and human action recognition for depth videos. A dynamic programming based modification is added to handle dynamic hand gestures and human actions. The framework is evaluated on several public benchmark datasets and is demonstrated to be effective in modeling shapes in depth maps. The drawback of current pipeline is that the approach to model temporal dynamics is not unified. The proposed two approaches of extending H3DF to cope with dynamic representation of a video sequence have different objectives. DP-based partitioning aims to solve the temporal alignment problem caused by different execution rates. The N-gram-based method, on the other hand, is designed to model local transition patterns. For example, to model a sequence of "raising hand", the DP method seeks to end up with 2 (or 3) gestures that can sufficiently summarize the action, *i.e.*, "lowered hand", ("raising hand") and "raised hand"; while N-gram method pursues to capture the motion during "raising hand". In other words, the DP-based method generates *"a sequential collection of gestures"* while the N-gram-based method generates *"a bag of motions"*. The two perspectives are both useful to capture temporal structures and transition patterns. But in practice, we found that the proposed DP method works better with dynamic hand gesture recognition while the proposed N-gram method works better with human action recognition. The reason for this may be the intrinsic difference between hand gestures and action recognition. Hand gestures information is conveyed mainly by the shape of hand while motion is complementary information and human actions are highly performed by drastic motions of body parts. In addition, the l_2 metric used in our DP algorithm is prone to sparse noise but large in magnitude, which is more common in human action recognition. In our future work, a unified temporal dynamics-handling component will be studied.

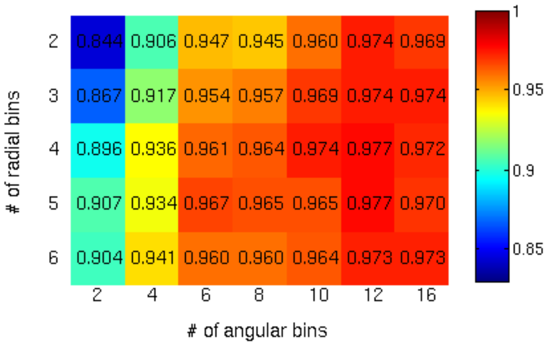
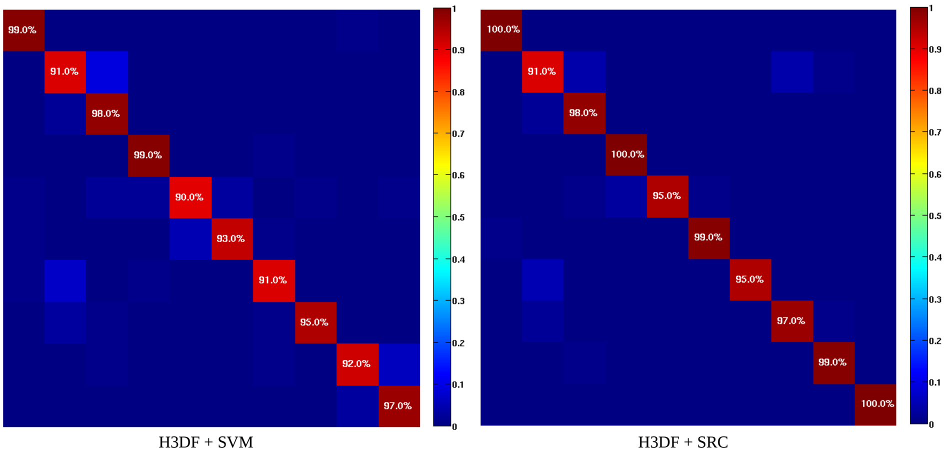


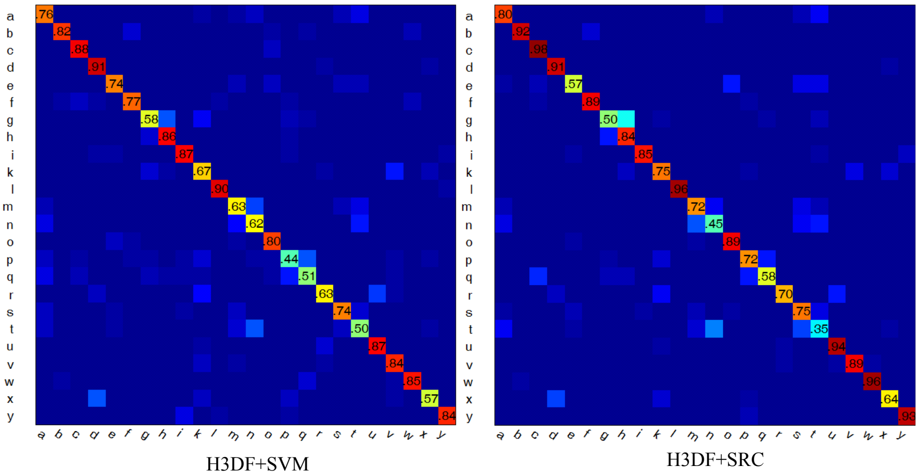
Figure 3.13: Recognition accuracies of different pooling granularity settings (y-axis for b_r, x-axis for b_a) on the NTU Hand Digits dataset [112].



H3DF + SVM

H3DF + SRC

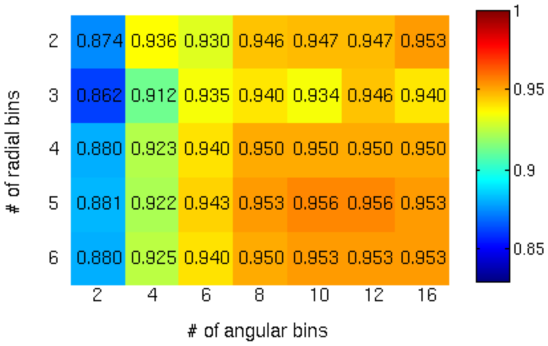
Figure 3.14: Confusion Matrices of our method on the NTU Hand Digits Dataset [112] in “leave-one-out” subject-independent test.



H3DF+SVM

H3DF+SRC

Figure 3.15: Confusion Matrices of our method on the ASL Finger Spelling Dataset [110] in “leave-one-out” subject-independent test.



# of radial bins \ # of angular bins	2	4	6	8	10	12	16
2	0.874	0.936	0.930	0.946	0.947	0.947	0.953
3	0.862	0.912	0.935	0.940	0.934	0.946	0.940
4	0.880	0.923	0.940	0.950	0.950	0.950	0.950
5	0.881	0.922	0.943	0.953	0.956	0.956	0.953
6	0.880	0.925	0.940	0.950	0.953	0.953	0.953

Figure 3.17: Recognition accuracies of different pooling granularity settings (y-axis for b_r, x-axis for b_a) on the MSR 3D Gesture dataset.

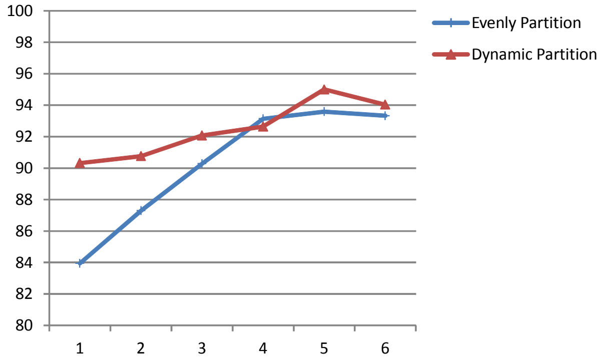


Figure 3.18: Recognition accuracies of different temporal strategies and numbers of segments (*i.e.*, x-axis shows K).

Chapter 4

Edge Enhanced Depth Motion Map for Dynamic Hand Gesture Recognition

After introducing a local depth descriptor (H3DF), we introduce a holistic depth descriptor (E^{2}DMM) in this chapter. Compared to single depth map-based descriptor (H3DF), E^{2}DMM is computed based on temporal accumulations of global motion traits, therefore it can naturally capture temporal information of human activities. In this chapter, we also introduce the Dynamic Temporal Pyramid (DTP) technique which can be applied in pooling strategy and counter the effects of temporal offsets.

In recent years, the ease of using depth cameras together with the promising application potential of depth cameras has attracted a lot of research efforts into it. As the representative successful debut of Xbox-Kinect [19] by Microsoft in Human Computer Interaction (HCI) and Entertainment, it has caused substantial revolutionary affects both in marketing as well as academia areas such as computer vision and image processing. Human action and gesture recognition, as a significant component of computer vision, naturally has benefited and evolved obviously.

Action and gesture recognition in depth videos has its endowed advantages over that in traditional color or grayscale videos. First, the background is relative clean since the depth sensor implicitly ignores the complex clutter pattern on the background, which is often the major headstream where noise comes from. Second, human body or other parts become easier to be segmented since the 3D spatial information is captured and visualized. Last but not least, the new type of data enables a different area of information which has rarely been touched by traditional action and gesture recognition research on color or grayscale videos. Although action and gesture recognition based on depth cameras is still a relatively new topic, many researchers have paved pebbles to its promising future.

One successful direction is to discover the correlation between action categories and body part joints, which uses estimated body joints [125] [134] to obtain a reduced representation of human body structure. Shotton *et al.* in 2011 and 2012 [125] [134], proposed to model the body joints estimation problem from a single depth frame. The authors found modes from census of per-pixel classification and solved it utilizing Random Forest and Conditional Regression Forests. On one hand, their work has enabled efficient human body joint extraction from a depth video; on the other hand, it provided many of other researchers a powerful tool to manipulate this kind of raw representation to help to solve their specific human action recognition problems. Simple features computed from body joints solely are proved to be effective in human action recognition problem from Human Computer Interaction (HCI) perspective [154] and Activity of daily living (ADL) perspective [176]. In [154], Wang *et al.* made a very interesting observation that, comparing with using the universe of body joints, using an action-category specific subset of them makes more sense.

However, since methods which are relying on pose estimation are vulnerable to the failure of such pre-processing by self-occlusion, twisted gesture, or unknown human body layout. All the difficulties, which are common in real life, are summed up to some researchers getting rid of using estimated joints [79] [169] [155]. Li *et al.* selected representative points on the contours of three orthogonal projections of the 3D point cloud of human body. In [155], the authors proposed a novel random sampling mechanism using class separability measure together with a novel feature called Random Occupancy Pattern (ROP). This method performed effectively with sparse coding. Motivated by the idea of Motion History Image [9], our previous work, Depth Motion Map with Histogram of Gradients (DMM-HoG) [169], was proposed to model an action as an energy distribution map over time and also reached good performance in several public datasets [114]. These methods have bypassed the obstacle of joint-based methods because body part estimation sometimes is not accurate and ambiguous for some subtle actions. These methods have also enabled gesture recognition from some part of the whole body, for example, hand gesture recognition.

Hand gesture recognition is a distinct and significant component of human action and gesture recognition since the information hand gestures convey is more sophisticated and linguistic than traditional activities. For example, American Sign Language (ASL) expresses more complicated information than jumping, hand-waving, running, *etc.* Since the substantial difficulty and complexity beneath the problem, related research work, especially those using depth cameras, is still on its infancy but a lot of work shows a promising potential. One approach is to recognize hand gestures using static depth frame as in [110] [112] [177]. In [110], the authors treated each static depth frame as a regular grayscale image and used a bank of Gabor filters to capture gradient information and solved the classification problem using random forests. Different from [110], the authors of [112] focused on a different type of information: contour; and a different application area: hand digits recognition. Other than using gradients and contours, our previous work [177] as described in Chapter 3 uses a histogram of 3D normals. For dynamic hand gesture recognition, ROP in [155] also achieved promising results.

4.1 E^{2}DMM and Dynamic Temporal Pyramid (DTP)

In this chapter, we propose a new representation: Edge Enhanced Depth Motion Map (E^{2}DMM), to recognize dynamic hand gestures based on depth video. Moreover, to capture more temporal structure information of hand gestures, we further propose a saliency prior Dynamic Temporal Pyramid (DTP) representation. The framework of proposed approach is as shown in Figure 4.1. By extracting (E^{2}DMM) and organizing using DTP, the input depth frame sequence transforms to two-layer motion maps. Then

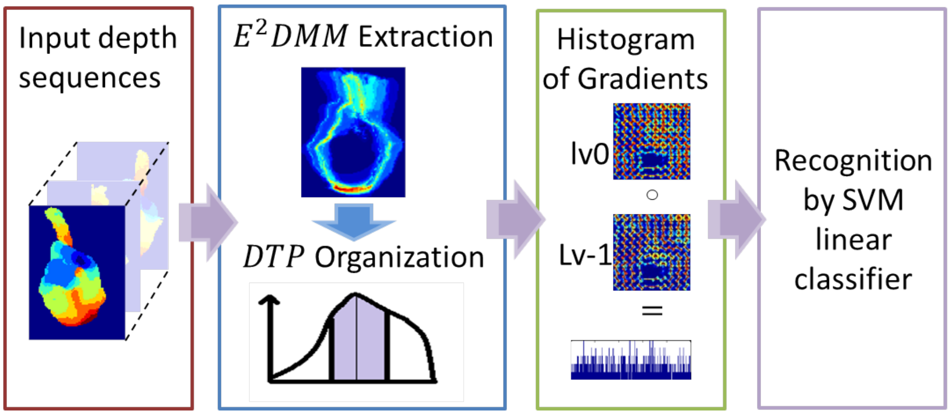


Figure 4.1: Flowchart of the proposed approach. After edge enhanced depth motion map extraction and dynamic temporal pyramid (DTP) organization, we apply Histogram of Gradients (HoG) to generate vectored representation of the two levels of E^{2}DMM (lv0 and lv-1). Finally, a SVM classifier is trained and utilized for classification.

Histogram of Gradients (HoG) feature is extracted from the two-layer motion maps and concatenated to generate a vectored representation. A SVM classifier is used to tackle the classification task. There are two benefits of such an approach. First, enhancing the edges provides more information for visually characterizing hand shapes. Second, salience prior temporal pyramid captures more accurate temporal layout of the depth frame sequence. Compared to the state-of-the-art methods [169] [154] [155] [57] [67], the proposed method can achieve higher accuracy in a public hand gesture recognition dataset [115] without more complicated decision models or any sparse favored dictionary learning, which further manifests the efficiency of proposed method. The rest of the chapter is organized as follows. Proposed approach is described in details in Section 4.2. Experimental results and the comparisons with the state-of-the-art are summarized in Section 4.3.

4.2 Extraction of E^{2}DMM

In this section, we firstly rephrase DMM in a more general form. We then propose the formulation and computation method of Edge Enhanced DMM (E^{2}DMM). A novel saliency prior dynamic temporal pyramid structure is also proposed and compared to traditional temporal pyramid.

4.2.1 Depth Motion Map (DMM)

Depth Motion Map (DMM) [169] is a visual representation of human activities by accumulating the motion of each frame in a depth video. DMM is a global descriptor mainly focusing on modeling the spatial energy distribution of human actions. A DMM of a depth video can be given as:

$$f(X_{i,j}) = \sum_{t=1}^{T-1} (\delta(|x_{i,j}^{t} - x_{i,j}^{t+1}| - \epsilon) + \sigma_{i,j}) \tag{4.1}$$

where X is a depth video given as a collection of depth frames $X = x^1, ..., x^T$ and ϵ is a parameter to determine the strength of motion, which is named *"penetration threshold"*; $\sigma_{i,j} = -e_{i,j}$ is a penalty term to suppress the energy accumulation in the edge pixels.

DMM accumulates the motion between each pair of consecutive depth frames to generate an energy distribution map to discriminatively represent an action. Usually Histogram of Gradients (HoG) operator is applied on a DMM to generate a concrete feature vector, as shown in Figure 4.2 (b) and (d). In this chapter, based on such modeling, we first discuss the difference and similarity between human action recognition and hand gesture recognition in depth video and propose edge enhancement process to mutate DMM to E^{2}DMM to adapt to hand gesture recognition. Then a new temporal pyramid based on temporal saliency is proposed to capture more temporal structure information.

4.2.2 Edge Enhanced DMM (E^{2}DMM)

According to [169], edge suppression is suitable for dramatic human action recognition since the contours do not provide useful information to help distinguishing between different actions, but may introduce variance between different subjects.

However, static pose or gesture plays a different role in the domain of hand gesture recognition. In fact, static pose or gesture conveys a significant portion of information and a lot of work has been done in this direction [177] [112] [110]. In the perspective of dynamic hand gesture recognition, static gesture together with motion provides an integral description of gesture category. Thus, we formulate Edge Enhanced Depth Motion Map (E^{2}DMM) computation by changing the edge suppression term to an edge enhancement term:

$$g(X_{i,j}) = \sum_{t=1}^{T-1} (\delta(|x_{i,j}^t - x_{i,j}^{t+1}| - \epsilon) + \rho * e_{i,j}) \tag{4.2}$$

where parameter ρ is a weight to tune the degree of edge enhancement, as shown in Figure 4.3. We will demonstrate the effectiveness of the edge enhancement term and the effect of different selections of ρ in Section 4.3.1.

4.2.3 Dynamic Temporal Pyramid

In this section, we show how to model the temporal structure using saliency prior dynamic temporal pyramid.

Traditionally, to capture the temporal structure or layout of an action, one may use temporal pyramid, which is very similar to the idea of Spatial Pyramid [73]. As shown in Figure 4.4 (a), temporal pyramid evenly divides the features into 2 or 4 or more buckets in the temporal dimension. Intuitively, since an action may usually have several phases: onset, apex, and offset while each phase contributes differently to the distinguish power of the final feature representation. Therefore, dividing the features in the time domain may help to address such temporal structure information. Moreover, the dimension of final feature representation is proportional to the total buckets the pyramid uses, *e.g.*, in Figure 4.4 (a), it uses a final feature vector with dimension 7 times to that only uses level 0 feature.

Therefore, we propose to use a new temporal organization method taking advantage of the energy distribution, which can be easily computed by modifying Eq. 4.2:

$$E(X) = \times_{t=1}^{T-1} \sum_{(i,j)\in M_t \times N_t} (\delta(|x_{i,j}^{t} - x_{i,j}^{t+1}| - \epsilon) + \rho * e_{i,j}) \tag{4.3}$$

where E(.) is a vector computed as the Cartesian product of sums of nonzero entries in frame t. E(.) is plotted as blue curve (the magnitude of the curve at a certain time represents the spatial integral of E(.) in a certain frame) in Figure 4.5 and its integral over time is plotted as red curve.

As a result, other than trying to use more space to store different representations of different temporal segments, we tend to use a separate feature space to summarize those frames with more significant information in terms of energy. Experimental results show that this organization is more suitable for temporal information capture and saves space which means less computation cost.

Detailed parameter discussion as well as settings will be conducted in Section 4.3. HoG is applied as in [169] to generate feature vectors. To evaluate the distinguish power we simply use linear Support Vector Machine (SVM) classifier without any sophisticate manipulating such as dictionary learning.

4.3 Experimental Results

In this section, we first introduce the public dataset we use for evaluation as well as its statistics. Then evaluation of E^{2}DMM and comparison with the state-of-the-art methods are described.

4.3.1 Experimental Setup

In the experiments, we use MSRGesture3D dataset [115], which was captured using a Kinect camera. The dataset is for dynamic American Sign Language (ASL). There are 12 gesture categories from two letters z and j to words: “Z", “J", “Where", “Store", “Pig", “Past", “Hungary", “Green", “Finish", “Blue", “Bathroom", and “Milk". There are 10 different subjects involved and each person performs each category 2 to 3 times. There are 336 samples in total, each of which is a depth sequence. The dataset is presegmented with only hand appeared in the depth videos

To extract E^{2}DMM, the input is a depth sequence and the output is a feature vector with a fixed length of dimension. In all our experiments, we normalize the patch sizes of E^{2}DMMs to 100×100. We use the off-the-shelf HoG generator in [25] with patch size 8, orientation bins 8, and all four normalization methods [1]. The dimension of final feature vector is 6400 after applying our proposed dynamic saliency prior temporal pyramid.

Following the benchmark setting as in [155], the performance evaluation is processed by using leave-one-subject-out strategy, which means each

[1] For more details related to parameters, please refer to [25].

time one subject is chosen for test while the SVM classifier is trained on the data composed by the remaining 9 subjects. The performance is calculated as the average classification accuracy after all subjects are tested.

4.3.2 Evaluation of the Proposed Method

In this section, we explore the discriminative power of proposed descriptor for dynamic hand gesture recognition and the effects of several parameters: penetration threshold and degree of edge enhancement (ϵ, ρ). Our default setting of (ϵ, ρ) is (10, 0.1), where the ϵ has unit of mm. The effects of different parameter settings can be visualized in term of classification accuracy as shown in Figure 4.6. From Figure 4.6, we can observe that with proper edge enhancement $(\rho = 0.1)$, the performance is better (1.5%) than no-enhancement $(\rho = 0.1)$ and edge suppression [169] $(\rho = -1)$. Thus, we select a stable penetration threshold ϵ = 10 and edge enhancement degree ρ = 0.1, as default parameters.

While adding level -1 for dynamic temporal pyramid representation, we firstly compute the frame index which has the highest peak in the energy distribution, then we search to the left till: 40% of total energy is included OR reaching the starting index, as left bound of level -1 window; we search to the right till: 40% of total energy is included OR reaching the ending index, as right bound of level -1. The level -1 uses another 3200 dimension feature vector which is then concatenated with level 0. In total, the dimension of final feature vector is 6400. We empirically set different weights for level 0 and -1 with 2 and 1. The overall accuracy after applying this is 90.5% for our proposed method, which is used for comparison with the state-of-the-art approaches. Comparison of proposed dynamic temporal pyramid framework and traditional temporal pyramid framework (level 0 and level 1 as shown in Figure 4.4(a)) is shown in Table 4.1 with the same weights setting. Proposed DTP outperforms traditional temporal pyramid (TP) with 1.6% and 1.8%, respectively, in overall accuracy both in un-weighed and weighed cases.

Table 4.1: Comparison between proposed Dynamic Temporal Pyramid (DTP) and traditional Temporal Pyramid (TP).

	Proposed DTP	Traditional TP
un-weighted	88.07%	86.45%
weighted	**90.53%**	88.70%

4.3.3 Comparison with the State-of-the-arts

In this section, we compare our method with several other state-of-the-art methods in term of accuracy. The confusion matrix is as shown in Figure 4.7. Our method performs well in most classes and the worst one reaches 74%.

We compare the proposed method with several state-of-the-art methods such as [57] [67] [155] [169] as listed in Table 4.2. Our proposed method performs best (90.5%) in term of averaged classification accuracy. Compared with DMM in [169] without edge enhancement, proposed E^{2}DMM together with dynamic temporal pyramid outperforms the previous one with 2% accuracy.

Compared with Random Occupancy Pattern feature together with Sparse Coding in [155], our approach outperforms the performance in [155] with about 2% and more details can be seen from Figure 4.8. Besides, our method performs rather stable while the worst accuracy in “j" is 74%, which is much higher than the worst cases in [155], 57% (in class “green" and “where"). The results demonstrate that our method is more general and suitable to represent dynamic hand gestures of different categories.

Table 4.2: Comparison of proposed method and other methods.

Method	Accuracy
CNN [57]	0.69
Occupancy Features [67]	0.805
Silhouette Features [67]	0.877
ROP [155]	0.868
ROP-SC [155]	0.885
DMM [169]	0.882
Proposed no TP	0.899
Proposed + TP	0.887
Proposed + DTP	**0.905**

4.4 Discussion

In this chapter, a depth map-based holistic descriptor is proposed. It is demonstrated that focusing and magnifying hand contours can improve dynamic hand gesture representation accuracy. In addition, a energy-based temporal pyramid-based pooling method is proposed to capture more temporal information. We also notice that the structure of temporal pyramid is fixed and needs to be pre-defined by hand. In the future work, our direction will be relaxing this constraint and delegating the decision power of pyramid structure to the algorithm. Therefore, the model can be further applied in action segmentation and recognition for longer videos.

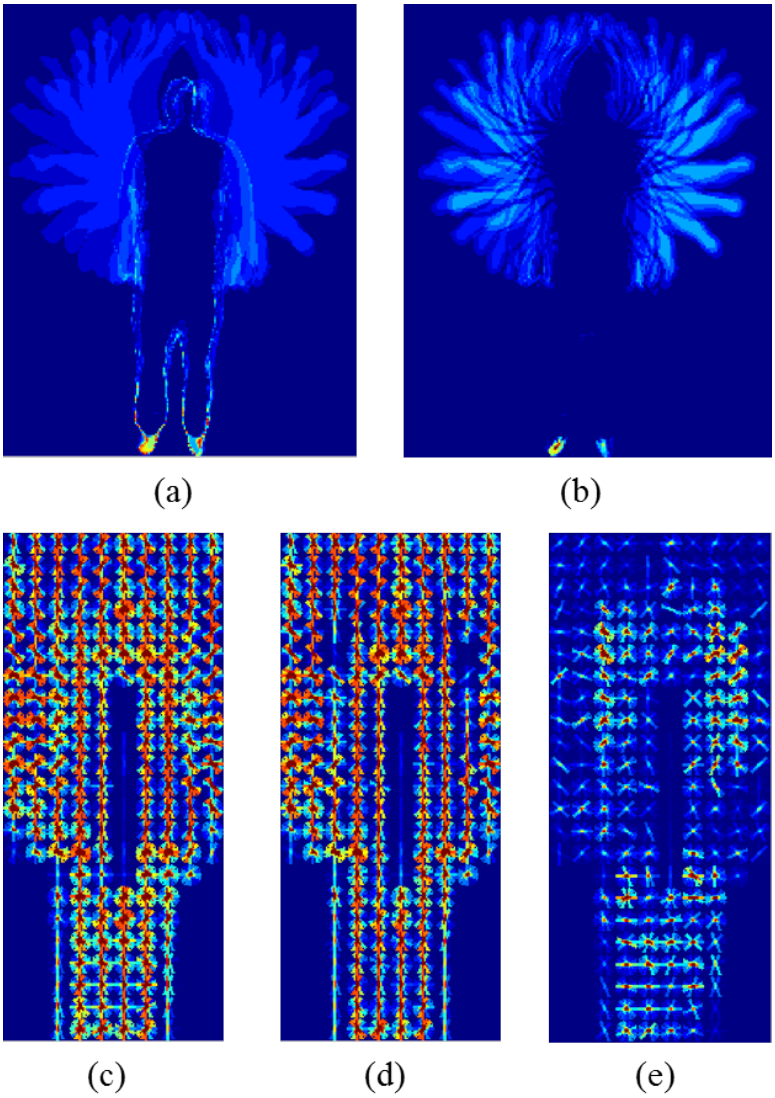


Figure 4.2: Illustration of edge suppression in DMM [169] computation of action “Two Hands Wave" in [114]. (a) DMM without edge suppression. (b) DMM after edge suppression in Eq. 4.1. (c) and (d) are Visualizations of HoG Representations of (a) and (b), respectively. (e) is the difference in feature space of (c) and (d) . The contour of human body can be obviously observed without edge suppression, which causes ambiguity as shown in (e).

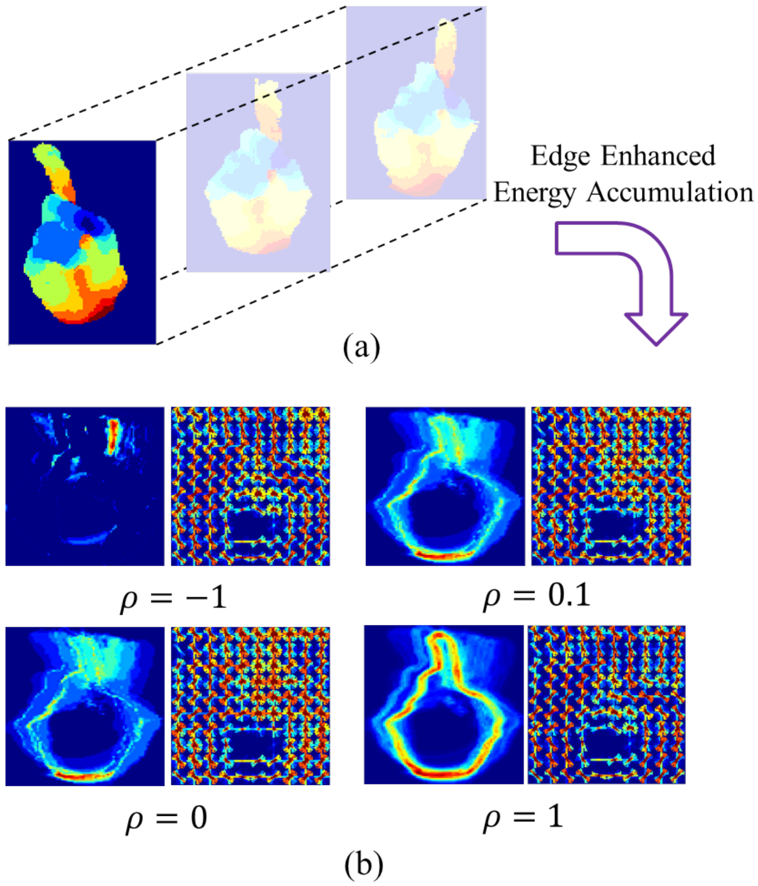


Figure 4.3: (a) A depth video showing the gesture “Where" in American Sign Language (ASL). (b) Accumulative Motion Maps and corresponding visualizations of HoG of different selection of ρ. When ρ = -1, it degenerates to DMM and when $\rho > 0$, it is in the form of E^{2}DMM.

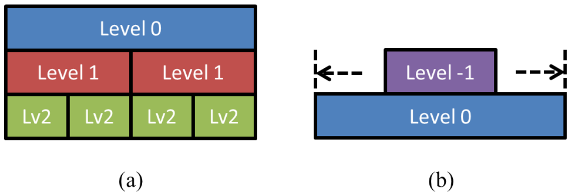


Figure 4.4: Illustrative comparison between (a) traditional temporal pyramid and (b) our saliency preferred dynamic temporal pyramid structure. Compared to the fixed branching in (a), the branching in (b) is dynamically determined based on the energy distribution.

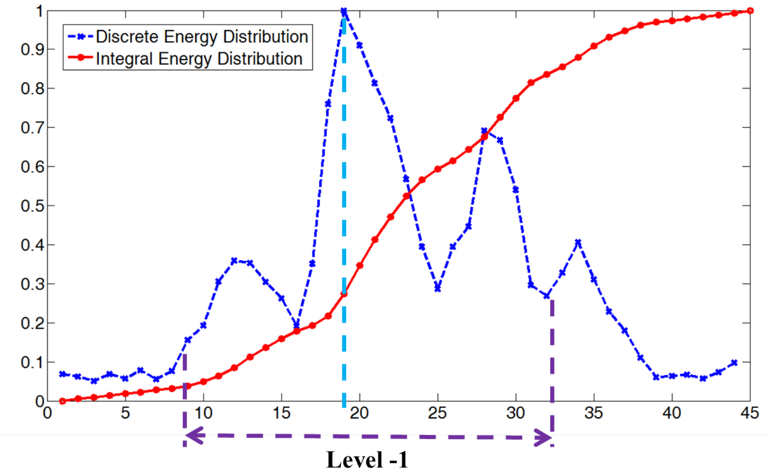


Figure 4.5: Illustration of how to dynamically select “level -1" based on the energy distribution over time. Blue curve shows the energy of each frame. Red energy shows the integral energy over time. Cyan dashed line shows the frame with highest energy and purple dashed line shows the chosen window for “level -1". X-axis is frame index and Y-axis is the relative magnitude of each curve.

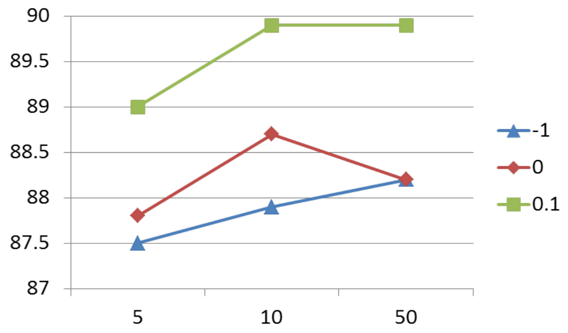


Figure 4.6: Performance evaluation of E^{2}DMM on dynamic hand gesture dataset [115] of different parameter settings. X axis: penetration threshold ϵ; Y axis: performance in term of classification accuracy. Blue, red and green colored curves indicate degree of enhancement -1, 0, and 0.1 respectively. Notice that when degree of enhancement is -1, it actually represents traditional DMM [169].

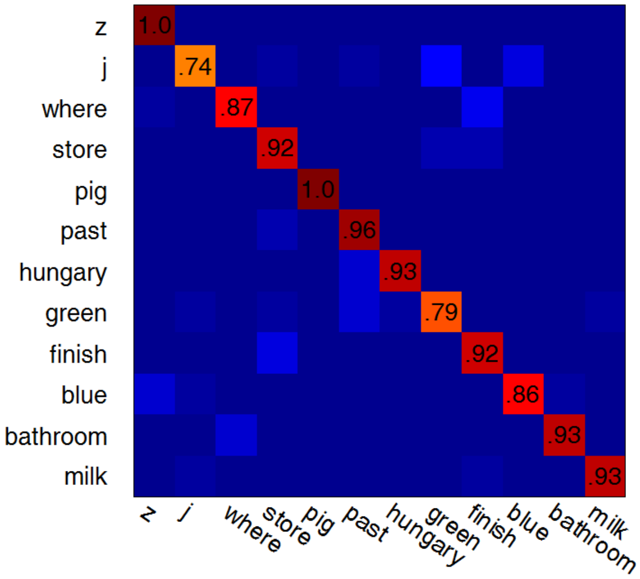


Figure 4.7: The confusion matrix of the proposed method on dataset Gesture3D.

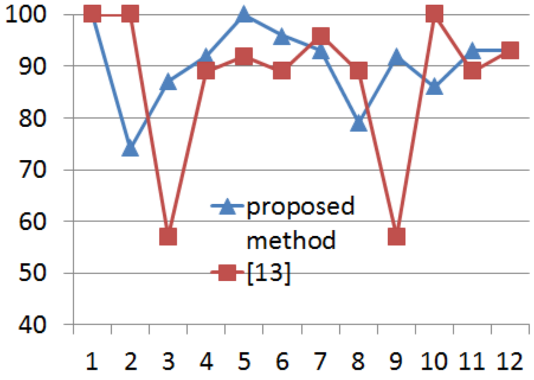


Figure 4.8: Class-wise accuracy comparison of proposed method and Random Occupancy Pattern with Sparse Coding (wang2012robust-SC) in [155]. Obviously, besides that our method outperforms method in [155] with 2%, proposed method is more stable, which means our representation is more general for different action classes.

Chapter 5

Subject Adaptive Human Affection Recognition

Overview. After introducing two depth-based descriptors for human action and hand gestures, we introduce and propose a multi-modality method in this chapter. The proposed method fuses both facial expression and body motions, depth and RGB channels for human affection recognition. Affection is a disposition of mind or body, which is often expressed by facial expressions and body gestures. Some affection categories can be conveyed solely from facial expressions or solely from body gestures. But it is more natural and common that facial expressions and body gestures jointly express an affection.

The success in facial expression recognition provides a plentiful of approaches to solve the problem in one perspective. Action Units (AUs) for Facial Action Coding System (FACS) [39] is a good modeling for facial expressions by decomposing the facial expressions into smaller organ-based movements, such as drawing brows and opening mouth. Facial expression recognition based on AUs is successful and has attracted a lot of attentions [140] [145] [179]. In addition, recognizing human emotions from body gestures is also a growing research area in recent years [123]. Especially after the debut of Kinect depth camera [19], the new type of sensor together with its technologies provides powerful tools for human activity analysis. The depth channel makes it easier to segment human from clutter background and therefore research based on this novel information channel has been conducted on an unprecedented scale [154] [175] [125].

However, the difficulty in facial expression recognition is always proportional to the degree of subjective variance. As illustrated in Figure 5.1, subjective variance in image space is much larger than expressional variances. In the domain of Face Recognition (FR), the subjective variance is inter-class variance and expressional variance is intra-class class variance. However in the domain of Expression Recognition (ER), the roles of the two kinds of variances are reversed and therefore this phenomenon brings benefits to FR but harms to ER. Approaches are proposed to reduce the with-in class variance and increase the between-class variance, such as Linear Discriminant Analysis (LDA), or Fisher's Linear Discriminant [36]. Sparse Representation based Classification (SRC) [159] provides an informative way to image classification. In SRC, a query image is coded using a sparse dictionary whose bases (columns) are training samples with or without sparsity constraint; then the query image is reconstructed by the bases with sparse coefficients as well as sparse residuals. In [166], the

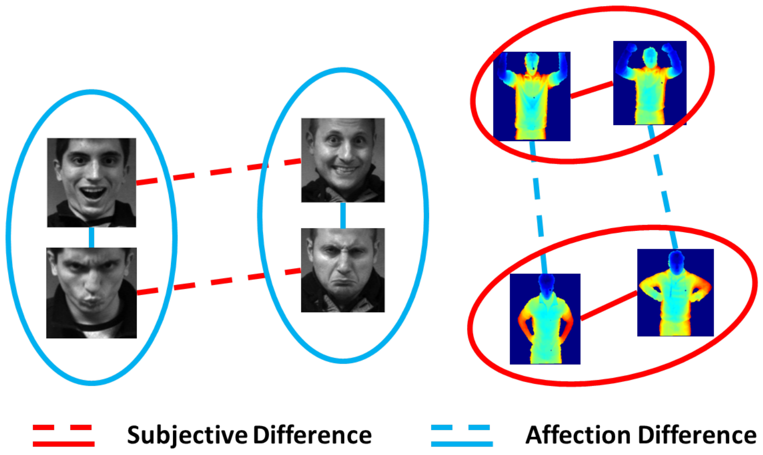


Figure 5.1: For facial expression, subjective (intra-class) variance is much larger than inter-class variances (expression). It brings benefits for expressional invariant face recognition but difficulties to subjective invariant expression recognition. This phenomenon is less severe for body gestures in Depth channel.

authors combined SRC and Fisher Discriminant criteria to propose an algorithm to learn a structured dictionary and providing informative reconstruction residual for class recognition. A low-rank regularization constraint is added to FDDL is also demonstrated to be useful in FR [78].

In this chapter, we utilized the classification scheme proposed in [166], which uses the residuals from class-wise reconstructions as classification criteria. We argue that instead of using all training samples for sparse reconstruction with the huge subjective variance, it makes more senses to select a subset of subjects using FR first and recognize affection then. Then we propose a joint affection recognition combining facial expressions (from RGB channels) and body gestures (from the Depth channel) with subjective adaption and joint decision making based on reconstruction confidence in sparse representation. The contributions of our work have two aspects:

1. First, we propose a subject adaptive sparse representation approach by combining the idea from [159] and [166] and reconstruct the query image from subject related subgroups.

2. Second, we address the joint recognition problem using the confidence computed from the residuals of sparse representation and experiment results demonstrate that the combination can be effective without additional computational cost.

Additionally, we also provide a combinatory dataset for joint affection recognition with both facial expressions and body gestures. Both color images and depth images are collected for multi-modal recognition.

An overview of the proposed framework is illustrated in Figure 5.2. Face patches and body gesture patches are extracted from RGB channels and depth channel of the input video respectively. We firstly apply Robust Alignment (RASL) [107] to align the frames in each video sequence. Then representative frames (queries) are selected based on "apex" position, where expressional intensity is highest as discussed in [16]. Subjective adaption is to select a group of most similar subjects based on SRC based

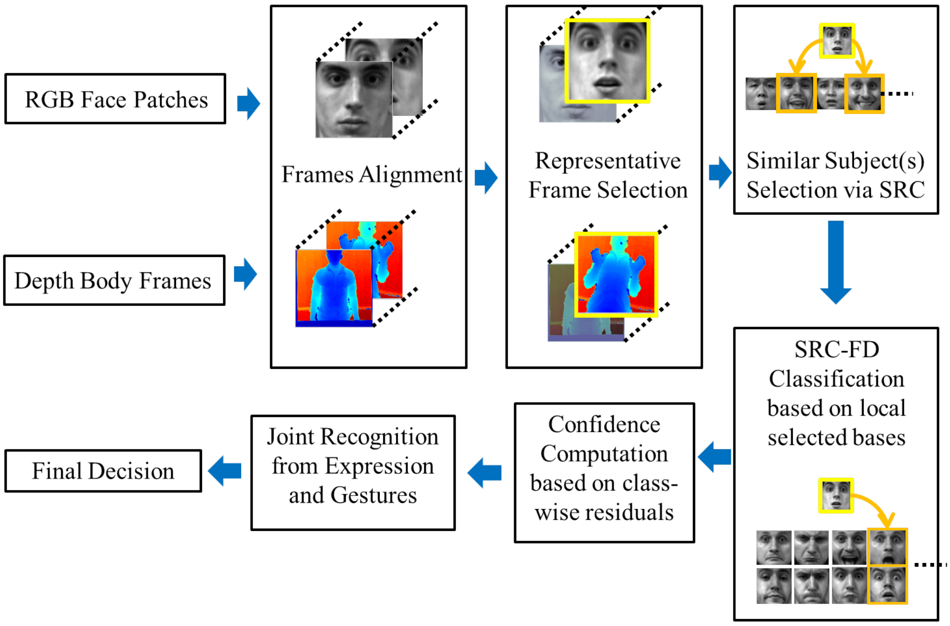

RGB Face Patches
Depth Body Frames
Frames Alignment
Representative Frame Selection
Similar Subject(s) Selection via SRC
SRC-FD Classification based on local selected bases
Confidence Computation based on class-wise residuals
Joint Recognition from Expression and Gestures
Final Decision

Figure 5.2: The workflow of our framework. After selecting representative frames from aligned frame sequence, we apply subject selection for a given testing subject. Affection class is recognized for testing queries with selected training data using sparse representation based classification. Then a joint decision from expression model and body gesture model is made based on confidences calculated from class-wise reconstruction residuals.

face recognition. The query image is then reconstructed from the selected subjective dictionary. Fisher Discriminative Sparse Representation Classification (SRC-FD) is used for class inference. The final decision of affection class is made according to the confidence score based on class wise reconstruction residuals.

The organization of the rest of this chapter is as following. Section 5.1 introduce how we align frames and select the most representative ones from them. Proposed subject adaptive affection recognition and joint decision making framework is introduced in Section 5.2. We also describe our new collected affection recognition data set in Section 5.3 and proposed framework on this data set is evaluated and discussed in Section 5.4.

5.1 Pre-processing

Given a sequence of face patches, we firstly need to align them and select the most representative frame out of them. The misalignment in the sequence is introduced by both human movement and noise in face detection. To align them, we apply the RASL [107] algorithm which uses sparse and low-rank matrix decomposition. Sparse learning based frame alignment takes advantages of the inner structure of the given sequence of similar frames (*e.g.* face patches of the same subject) and reduces the noises with rare occurrence. For representative frame selection, we select the middle frame of the apex area [16] according to expression intensity.

As illustrated in Figure 5.3, RASL [107] smooths the expressional intensity curve by representing the "intermediate" frame with "apex" or "neutral" frame. The red curve indicates the intensities of un-aligned sequence

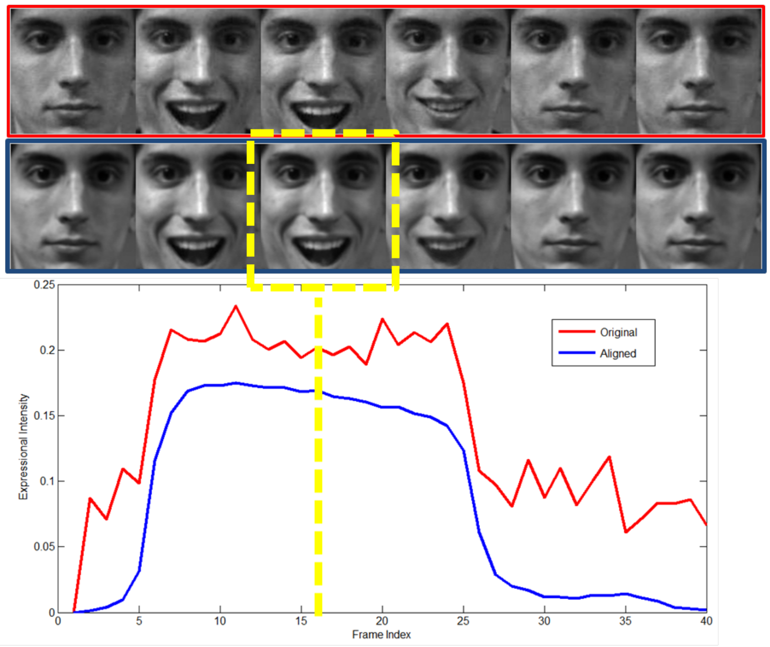


Figure 5.3: Alignment and representative frame selection. The first row with red boundary shows the original un-aligned face patches, one can observe variances due to poses and minor changes. The second row with blue boundary shows the aligned face patches via RASL [107], one can observe that differences due to factors other than expressions are eliminated. Expressional intensity curves are shown in the bottom. Selected representative frame is indicated by yellow dashed box.

and blue for aligned sequence. Yellow dashed box shows the final selected representative frame. For body gestures, the pre-processing step is the same as facial expression.

5.2 Subject Adaptive Joint Affection Recognition via Fisher Discriminant Sparse Representation

In this section, we firstly review sparse representation based classification (SRC) [159] and Fisher Discriminant. Then our proposed two layered subject adaption framework for affection recognition is described. Finally, a joint recognition framework is proposed based on the class-wise reconstruction residuals.

5.2.1 Sparse Representation Classification with Fisher Discriminant

Sparse representation based classification (SRC) was proposed in [159] by Wright *et al.*. Given C as the set of class labels, we have $A = [A_{C_1}, A_{C_2}, ..., A_{C_c}]$ as the dictionary of training samples. In our approach, A is the matrix of vectored frames, *i.e.*, $A_{C_i \in C} = [vec(x_1^{C_i}), vec(x_2^{C_i}), ..., vec(x_n^{C_i})]$, where $x_j^{C_i}$ is the j^{th} image (face path or gesture patch) of class i. Given a query image q and its vectored instance $y = vec(q)$, the SRC via l_1-minimization is given as:

$$\hat{\alpha} = \underset{\alpha}{\operatorname{argmin}} \|\alpha\|_1 \quad s.t. \quad \|y - A\alpha\|_2 \leq \lambda \tag{5.1}$$

Therefore, Classification rule is given as:

$$identity(y) = \operatorname*{argmin}_{C_i} r_{C_i}(y) \tag{5.2}$$

where class-wise reconstruction residual $r_{C_i}(y)$ is given as:

$$r_{C_i}(y) = \|y - A\delta_{C_i}(\hat{\alpha})\|_2 \tag{5.3}$$

where δ_{C_i} is the characteristic function that selects the coefficients associated with that class.

According to FDDL [166], the SRC classification rule can be re-written as an "SRC-FD" form:

$$\hat{\alpha_{C_i}} = \operatorname*{argmin}_{\alpha} \|\alpha\|_1 \quad s.t. \quad \|y - A_{C_i}\alpha\|_2 \leq \lambda \tag{5.4}$$

where A_{C_i} is the sub-dictionary associated with class C_i. Thus the class-wise residuals in Eq. 5.3 is re-written as:

$$r_{C_i}(y) = \|y - A_{C_i}(\hat{\alpha_{C_i}})\|_2 \tag{5.5}$$

Noted that Eq. 5.5 corresponds to FDDL with global cost weight as 0. We called this classification method as Fisher Discriminant SRC (SRC-FD). Comparing Eq. 5.3 and Eq. 5.5, the latter one is more intuitive. The classification is actually to find the optimal space spanned by bases of a certain class which minimize the reconstruction error. In addition, the residual formula in SRC-FD makes the inference much efficient, which is critical in applying SRC to real applications. To illustrate this argument, we test the l_1-minimization using Least Angle Regression [30] on a simulated dataset which has 10,000 instances and each instance is of dimension 1000; a query instance is given to reconstruct. The simulated data is randomly generated and each instance is normalize to unit l_2 norm. We partition the training set into n non-overlapping subsets and the query is reconstructed on each subset. The overall computation time and averaged reconstruction error for

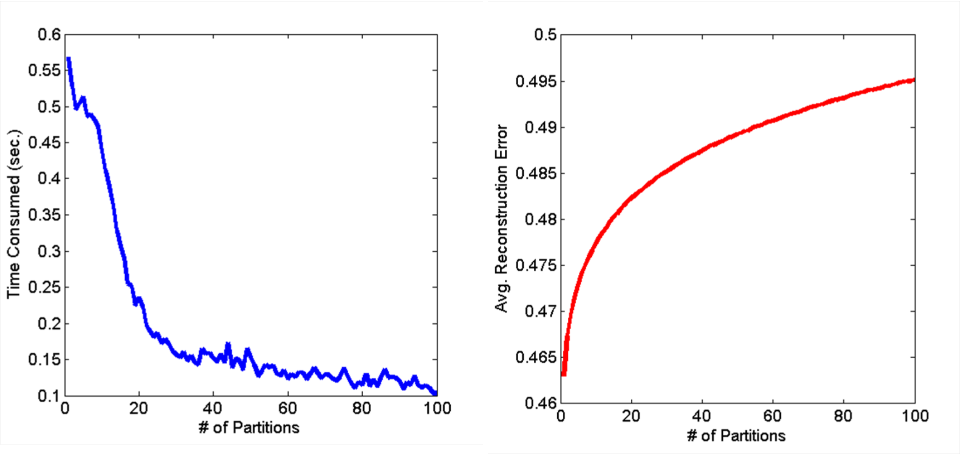


Figure 5.4: Timing and avg. reconstruction errors on Simulated data with 10,000 training samples and 1 query instance. We can observe that time consumed decreased as the number of partitions increases.

different n are as illustrated in Figure 5.4. With the number of partition increases, the computation time decreases. This is because l_1 minimization is not linearly proportional to the number of training instances. However, using a larger dictionary can achieve better reconstruction error. In our work, since the absolute reconstruction error is not a concern *per se*, so we use the classification form as in Eq. 5.5.

5.2.2 Subject Adaptive Affection Recognition

Both SRC [159] and FDDL [166] achieve impressive recognition rates in face recognition (FR) and have been demonstrated to be robust to varying illuminations, occlusion and expressions. However, subjective robust expression recognition (ER) is harder than expressional robust face recognition. The reasons are two-folded: 1) in term of pixel-wise variance, the distance between subjects of the same expression is much larger than the distance between expressions of the same subjects, as shown in Figure 5.1; 2) behavior habits of different subjects make the subjective variance much larger and the expressions harder to model. The two reasons jointly make ER a harder problem than FR.

The similar phenomenon also exists in gesture recognition. Instead of facial appearance, subjective appearance variances in body gestures are due to subjective body sizes, types and clothes. However, because 1) the body gestures are always much more drastic than facial expressions and 2) we extract body gesture information from Depth channel which ignore much of the appearance variances (*e.g.*, different clothes), this phenomenon is actually not crucial at all. Therefore, the effect of applying the subject adaptive framework on gesture recognition is limited to overcome subjective behavioral variance.

In this chapter, we propose a two layer recognition structure to pursue subjective robust affection recognition. The first layer is actually a face recognition problem. The motivation of the first layer recognition is straightforward: given a query image of an affection, it is more naturally to identify who is the subject and check if previous records exist in our training data, if so, using the data of the same subject is more efficient and accurate. However, it is unrealistic to assume all the subjects have already been included by the training dataset. Therefore, we define the first layer recognition step as to find a fixed number of most similar subjects such that the identity information (appearance, behavioral habits, *etc.*) can be approximated using the instances from the selected subjects.

The first layer recognition is as illustrated in Algorithm 2. To reach better consensus subjects selection, we use a batch of queries (from the same subject) each time, *i.e.*, $|Y| > 1$. An upper bound limit on maximal available subjects (N) is also given. The propose algorithm selects a subgroup (A^*) of training instances (A) for expression recognition using SRC-FD with l_1-minimization. One may doubt that it is not valid to know that the given batch of frames are from the same subject. We concede that sometimes it is the case but in reality, an affection recognition system can always capture more than one frames from the same subject and select similar subjects in training set as an off-line initialization. In the other hand, algorithm 2 is still functional when $|Y| = 1$.

Algorithm 2: Subject Adaptive sub-dictionary selection.

```
Input: training instance matrix $A$, testing instance matrix $Y$, subject number limit $N$
Output: subject adaptive sub-dictionary $A^*$
$A^*$ = empty;
for $y \in Y$ do
    for $S_i \in S$ do
        $\hat{\alpha} = \operatorname{argmin}_\alpha \|\alpha\|_1 \quad s.t. \|y - A_{S_i}\alpha\|_2 \leq \lambda$;
        $r_{S_i} = \|y - A_{S_i}\alpha\|_2$;
    end
    $\hat{S} = \operatorname{argmin}_{S_i} r_{S_i}$;
    vote($\hat{S}$) += 1;
end
$n = 1$;
while $n \leq N$ and $!allzeros(vote)$ do
    $s = \operatorname{argmax}_{S_i}$ vote($S_i$);
    $A^* = [A^*, A_s]$;
    vote($s$) = 0;
end
return $A^*$
```

Based on the selection of A^* (which can be represented as $A^* = [A_1^*, A_2^*, ..., A_c^*]$, where A_i^* is the class-wise subset of A^*), we can determine the affection class of each query y according to Eq. 5.4 and 5.5.

Figure 5.5 illustrates the proposed 2-layer recognition framework. When a query image is given, the first layer recognition process seeks at most N ($N = 2$ in this example) subjects whose samples can best approximate the query image. After we have selected the subset of subjects (colored as green rows in the training data matrix), the selected subset is further partitioned into C classes, $i.e.$, subsets with affection labels. The second layer recognition seeks the "row" which can best approximate the query image (final decision and selected subset are colored as red boxes).

5.2.3 Joint Decision Making via Confident Reconstruction Prior

When there are multiple models have the same set of class labels, as in this chapter, facial expression model and body gesture model, how to effectively combine them to make a joint decision is an issue. One can combine the models in an early phase by feature concatenation or make a joint decision only based on the decision scores given by different models. In this chapter, we apply the latter one since the early fusion in feature representation level can be overwhelmed by the dominant feature channel, if there exists one.

In our work, we have two models: facial expression model and body gesture model. Each model uses the same classification rule based on Sparse Representation (SRC). Since the decision in each model is made according to the smallest residual in term of l_2 norm. It is straightforward to derive the confidence score of an decision, denoted as $F(.)$ by the margin between the smallest residual and the second smallest residual. The assumption under the confidence score formula is that a "confident" decision should be

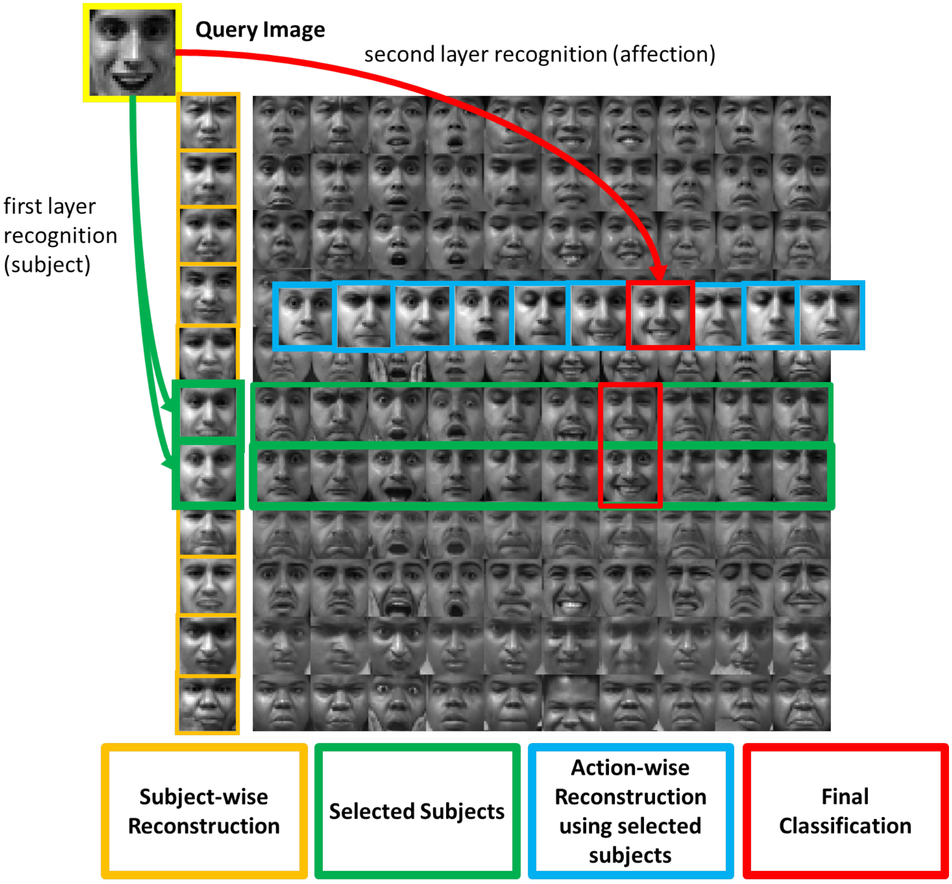

Query Image
second layer recognition (affection)
first layer recognition (subject)
Subject-wise Reconstruction
Selected Subjects
Action-wise Reconstruction using selected subjects
Final Classification

Figure 5.5: Illustration of our two-layer affection recognition framework. The first layer is to select a subset of subjects which can best approximate the query image(s). The selected rows (colored as green) are used for second-layer affection recognition using SRC. The action-wise approximations are colored as blue. The final decision is to find which class can best approximate the query image. The final decisions are shown in red.

made easier with a more comparative significant smallest reconstruction error. Then the confidence scores of both models are used for weighing the reconstruction error and we make the joint decision by selecting the smallest weighted sum of the class-wise reconstruction residuals. The procedure is as shown in Algorithm 3.

5.3 Face and Gesture RGBD Dataset

In this section, we introduce a new Face and Gesture RGBD dataset (FG-RGBD) we collected for affection recognition with 1920 affection samples.

In [43], the authors presented a widely used bi-modal public dataset for combinatory recognize affective behavior categories from both facial and gesture model. However, as recent success in Kinect and related research in the depth channel, there is a trend that researchers are mining more complementary information from this novel information source instead. There are a lot of datasets have been presented for using depth channel as a counterpart for research topics and have proved their effectiveness, such as MSR

Algorithm 3: Joint Decision Making from Face and Gesture.

```
Input: Facial expression dictionary $A^{*,f}$ and body gesture dictionary
       $A^{*,g}$, a query $y = [y_f, y_g]$
Output: Affection label $\hat{C}$ of y
for $C_i \in C$ do
    $\hat{\alpha^f} = \text{argmin}_\alpha \|\alpha\|_1 \quad s.t. \|y - A^{*,f}_{C_i}\alpha\|_2 \le \lambda$;
    $\hat{\alpha^g} = \text{argmin}_\alpha \|\alpha\|_1 \quad s.t. \|y - A^{*,g}_{C_i}\alpha\|_2 \le \lambda$;
    $r^f_{C_i} = \|y - A^{*,f}_{C_i}\hat{\alpha^f}\|_2$;
    $r^g_{C_i} = \|y - A^{*,g}_{C_i}\hat{\alpha^g}\|_2$;
end
$conf^f = F(r^f)/(F(r^f) + F(r^g))$;
$conf^g = F(r^g)/(F(r^f) + F(r^g))$;
$\hat{C} = \text{argmin}_{C_i} r^f_{C_i} * conf^f + r^g_{C_i} * conf^g$;
return $\hat{C}$
```

Gesture 3D, MSR Action 3D, and MSR Daily Activity 3D [114]. However, to the best of our knowledge, there is not such a dataset for affection recognition jointly from face and gesture combining both RGB channel and Depth channel. To fulfill this vacant slot, we thus present a Face and Gesture RGBD dataset for affection recognition (FG-RGBD) dataset which contains videos from both RGB channels and depth channel from a Kinect camera. Basic statistics are introduced briefly in this section.

There are ten affection categories in our FG-RGBD dataset, they are: *"uncertain"*, *"angry"*, *"surprise"*, *"fear"*, *"anxiety"*, *"happy (cheering)"*, *"happy (clapping)"*, *"disgust"*, *"boredom"* and *"sad"*. There are twelve subjects are recruited to perform the ten categories of affections according to a simple instruction. The subjects were asked to perform each affection in 4 different records (video clips), each record (video clip) the subjects were asked to repeat 4 times. The dataset contains a significant subject-variance because of two reasons: 1) the instruction used to direct the subjects has no more than two sentences for each action, so the subjects have a big freedom to perform the actions spontaneously, which is more close to reality. 2) The subjects are from different races and genders: there are 1 American-African, 2 Latinos, 4 Caucasians and 5 Asians; there are 2 women and 10 men.

In our FG-RGBD dataset, both RGB frames and depth frames are provided, skeleton joint estimations computed from off-the-shelf software are also provided yet not used in this work. There are in total of 480 videos as well as 1920 affection samples collected.

In this work, the 1920 samples are divided into training set with 960 samples and testing set with 960 samples while none of the subjects appears in both training and testing sets. Resolutions for RGB frames and depth frames are 1280×1024 and 640×480, respectively. Some sample frames from the FG-RGBD dataset are shown in Figure 5.6.

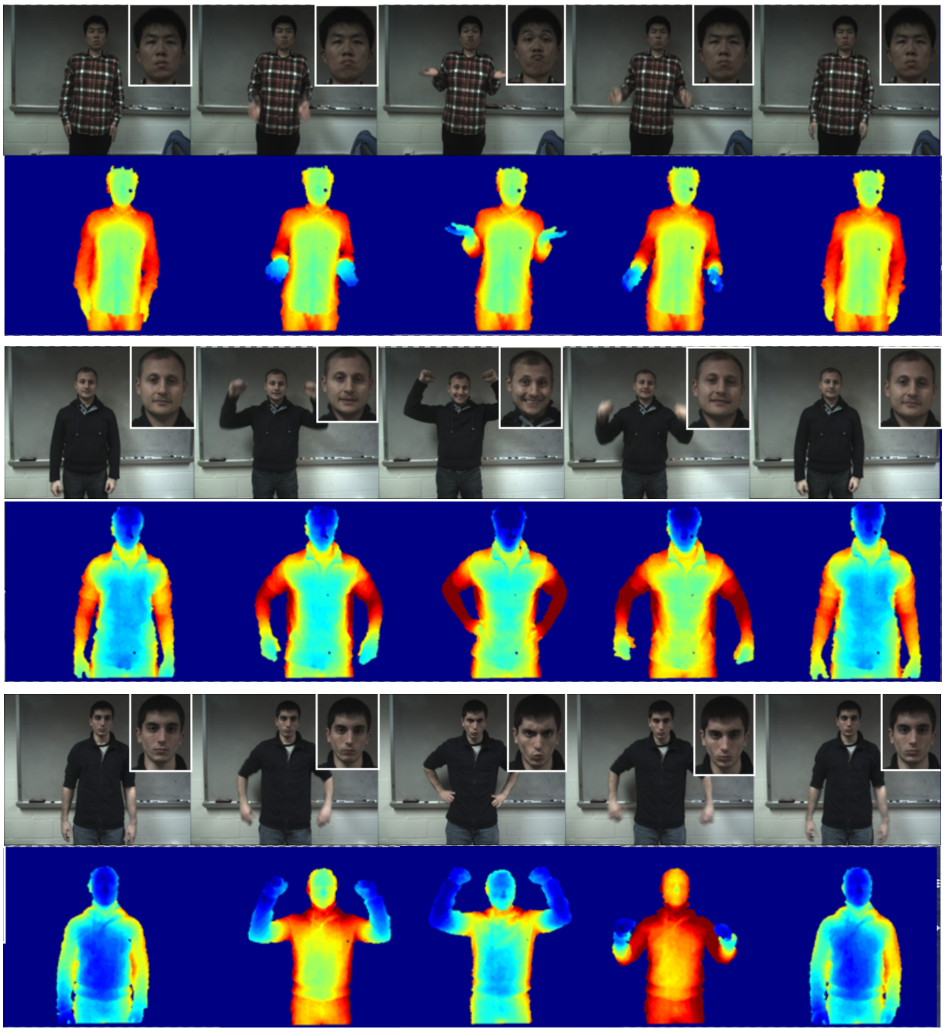


Figure 5.6: Example frames from proposed FG-RGBD dataset. Top, middle, and bottom panels are for *"uncertain"*, *"angry"*, and *"happy (cheering)"* respectively. RGB frames and enlarged face patches are shown on the upper and bottom rows of each panel.

5.4 Experimental Results

In this section, we use FG-RGBD dataset to evaluate tasks for facial expression recognition and body gesture recognition and joint affection recognition. Quantitative results in term of recognition rates are reported and compared with several baselines and state-of-the-art methods. Qualitative results in terms of cross-subject facial expression and gesture reconstruction are also illustrated for future discussion.

5.4.1 Selection on subject limit N

In this part, we discuss the effect of subject limit N selection in subject adaptive phase. As can be inferred from Figure 5.4, if we select a small N, the reconstruction error should be large but the time consumed in testing phase is reduced. Although reconstruction error is not a concern *per se* in this

Table 5.1: Performance comparison of different methods.

Approach	Exp.	Gest.	Joint
Logistic Regression	38.49%	46.35%	54.89%
FDDL [166]	43.39%	61.25%	64.84%
SRC[159]-FD	46.72%	62.34%	69.3%
Proposed Method	**48.80%**	**62.66%**	**69.7%**

work, a over-relaxed reconstruction error can bring bad recognition accuracy. Therefore, we need to find a good tradeoff when selecting N.

As illustrated in Figure 5.7, with the increasing N, recognition rate increases. But after $N = 8$, it becomes stable. Two examples of "happy" are shown for illustrating the progress of reconstruction. With the increase of subjects available, the reconstruction error can be reduced from a "hybrid" reconstructed face.

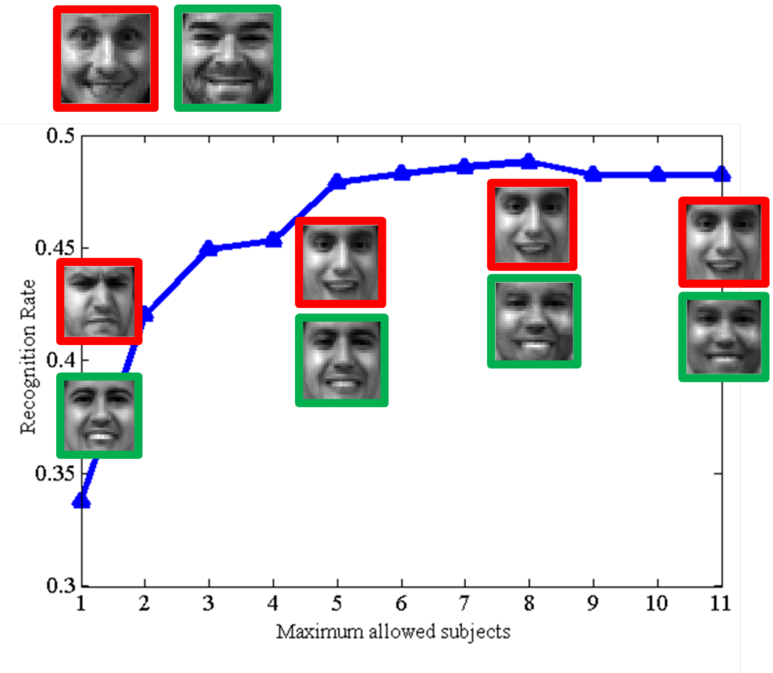


Figure 5.7: Recognition increases with N grows but after some certain value, it does not change. In this curve, the recognition rate reaches maximum as 48.8% when $N = 8$. An example of original query face y and reconstructed face using selected subjects ($A^*\hat{\alpha_y}$) are shown. We can observe that with the increase of N, the reconstructed face is de-personalized. Two samples of original query faces and reconstruction sequences with varying N are shown in red and green boxes ($N = 1, 5, 8, 11$, respectively). It can be observed that the reconstructed faces "evolve" to be more similar to the original queries.

5.4.2 Affection Recognition Performance Evaluation

In this part, we evaluation our affection recognition framework and compare our system with state-of-the-art methods [159] [166] in term of recognition rate. In our experiments, we conduct "leave-one-out" cross-subject tests for all methods and report the averaged recognition rates. We use the face detector in [97] to localize face patches in RGB channels. The extracted face patches are normalized to 150×150 resolution and aligned by RASL [107]. We extract representative frame in the low-rank part of the aligned face sequence (for details, please refer to [107]). The selected face patch is down-sampled to 30×30 and vectored as feature vector. The subject limit we select for facial expression channel is 8, but in average there are only 5 to 6 subjects are selected. As for body gesture model, we apply the same procedure as for face patches using [107], then the body gesture patch is normalized to 38×38 and vectored to be the feature vector for body gesture. For single model evaluation, the subject limit is also set to 8. In all our l_1 minimization process, we force the reconstruct coefficients $\alpha \leq 0$.

Table 5.1 shows the comparative results of several state-of-the-art methods and proposed framework. We also compare with a baseline method, logistic regression, since logistic regression can explicitly output classification probabilities of each class label. In Joint recognition, we apply the classification probability of facial expression model and body gesture model to the joint decision. As for FDDL, we directly use the published code for evaluation. Joint recognition with FDDL is accomplished by early fusion of facial expression frame and body gesture frame.

From Table 5.1, we observe that proposed method outperforms other methods, especially in facial expression recognition part. We also observe that in joint recognition, if we relax the subject limit constraint in body gesture channel, the joint recognition result is better. We report our best performance in Table 5.1.

Figure 5.8 illustrates the confusion matrices for affection recognition from facial expression, body gesture and joint decision making. We observe that the gesture recognition model solely perform superior than facial expression recognition model, especially between classes *"happy (cheering)"* and *"happy (clapping)"* since the facial expressional attributes are very similar while gestures vary drastically. Body gesture model performs much better in classes *"surprised"*, *"happy (cheering)"*, and *"sad"* since their gestures are much more distinct with others. However, this model is a little ambiguous in distinguishing between classes *"anxiety"* and *"happy (clapping)"* because both gestures have similar attributes; similarly, classes *"happy (disgust)"* and *"boredom"*, because both gestures contain action attribute like "raising hands in front of chest". Although facial expressional recognition rate is lower than body gesture recognition rate in almost every class, we can observe that the information contained in each model is quite complementary to each other: thus jointly recognizing affection classes reaches much higher recognition rates, such as in class "anxiety", "happy (clapping)" and "disgust".

5.5 Discusssion

In this chapter, a multi-modality affection recognition framework is proposed by applying a two-layer recognition framework. The proposed method can handle zero-shot facial expression recognition by searching the nearest person in the existing data pool. Although the framework can partially counter the effects of subjective differences, it is still problematic when the query sample is too distant from all training samples. In the future work, the framework will be combined with subspace learning technique to handle such cases more adaptively and accurately.

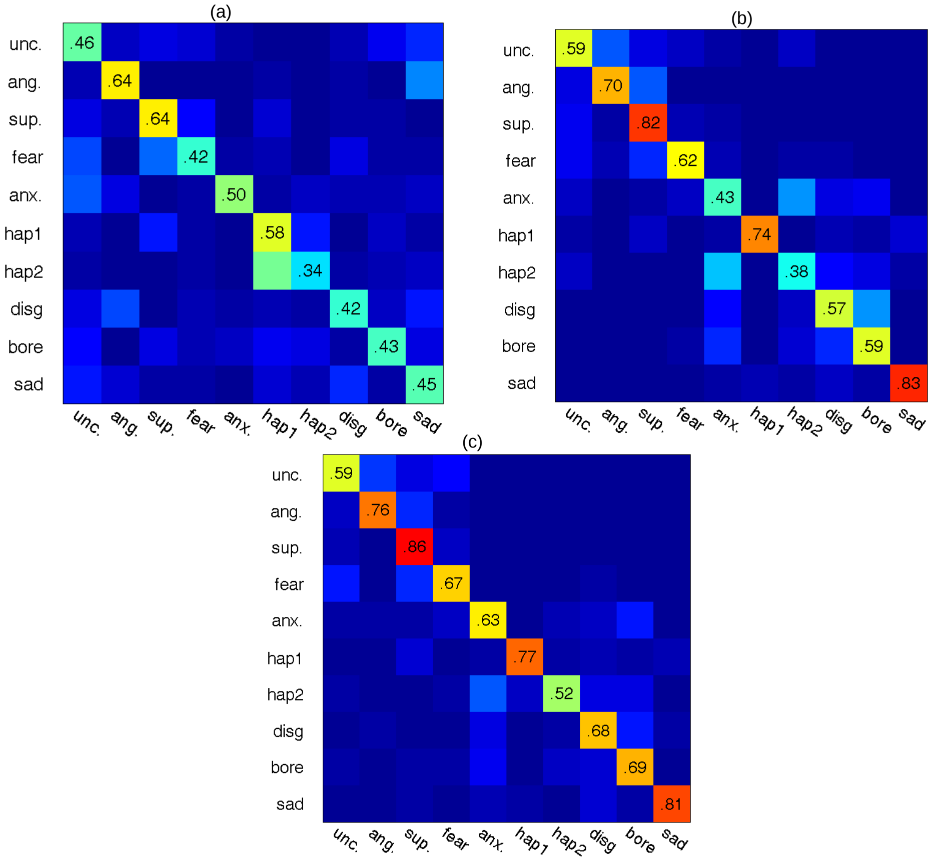


Figure 5.8: Confusion matrices for (a) facial expression model, (b) body gesture model, and (d) proposed joint recognition result. Obvious improvement is achieved in joint recognition infers that facial expressions and body gesture model contains very complementary discriminative information.

Chapter 6

Human Activity Attribute Learning based on Deep Neural Networks and Graph Lasso

In this chapter, we discuss two aspects in human activity analysis: 1) automatic feature learning and 2) mid-level action attribute learning. For the first task, we propose a multi-stream feature encoding pipeline using deep learning for automatic feature learning. For the second task, a graph-guided sparse coding framework is proposed to jointly learn different action attributes while following the two constraints, affinity and sparsity. In this chapter, we evaluate the proposed framework on both traditional action recognition and zero-shot action recognition.

In human recognition, information received from all channels can activate their corresponding neurons, which enable the high-level neurons in a deep network to accomplish the abstract recognition task (*e.g.*, face recognition) [101]. Inspired by this observation, we propose a uniform deep feature learning architecture, which can automatically learn homogeneous features from heterogeneous channels. These learned features are named as Deep Activations, since each feature element acts like a neuron that can be activated to capture some properties of the learning. Leveraging this deep learning network, our system is able to treat each information channel identically. In other words, only the Deep Activations are visible to the high-level learners such as the attributes proposed shortly. As a result, deep learning models have been successfully applied to large-scale visual recognition tasks using multiple layers of convolution filters [75, 122, 27].

Although the deep learning network has been very successful in visual recognition, the deep features are usually treated as mid-level features [129], and function like signal filters, which affect the recognition performance and limit their applications. Therefore, inspired by [83], instead of directly mapping deep features onto action labels, a set of pre-defined action attributes serves as mid-level representations. These attributes can boost recognition and enable new applications such as zero-shot learning. As human bodies/joints are easier to track than open source videos in [83], we argue that action attributes are more appropriate for actions in depth videos. To our knowledge, our work is the first attempt to leverage "attributes" to recognize actions from depth videos.

In object attribute learning, Jayaraman *et al.* [55] propose to use "groups" to define the relationships among object attributes. However, since object attributes are much more fine-grained than action attributes (*e.g.*, "furry" and "brown"), they can be organized into "groups" such as "color", "shape"

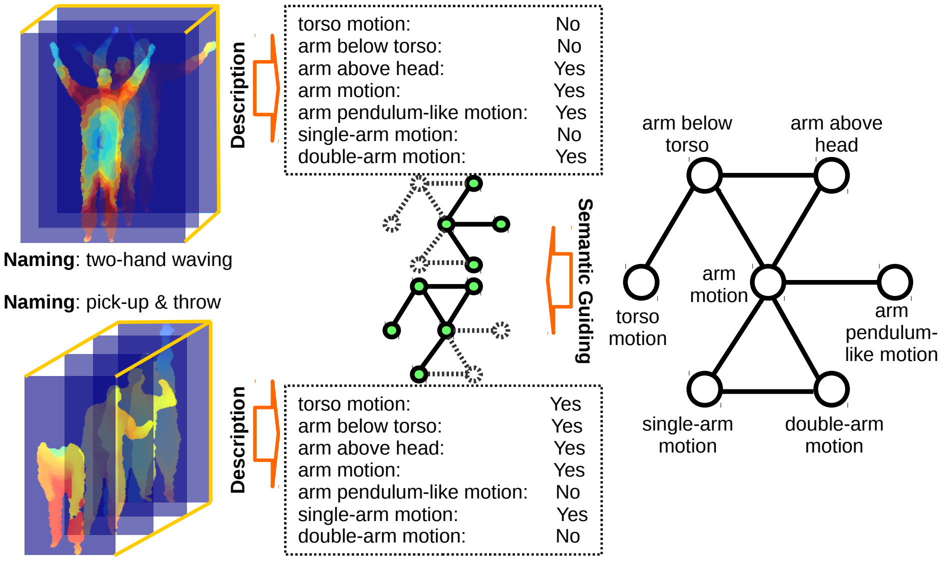


Figure 6.1: Illustration of the proposed joint action attribute learning algorithm. Instead of treating each action attribute independently, we apply a semantic graph to guide the joint action attribute learning algorithm to preserve the relationship among action attributes (*e.g.* "arm below torso" may share common information with "arm motion" and "torso motion".)

and "texture", but it is not helpful to coarsely group action attributes based on human bodies such as "arm", "head", "torso", *etc*. For instance (see Figure 6.1), the action attribute "arm below torso" is related to "arm above head" as both describe positions of upper limbs, but it is also related to "torso motion" because both are related to the body part "torso". Therefore, we argue that an undirected relation graph is better to capture the semantic/geometric relationships among action attributes compared to "groups". Actually, to some extent, the relation graph also groups attributes if they are close on the graph. But it captures more complex relationship beyond the "groups". Our experiments further verify that attribute detectors trained with the proposed graph perform much better than detectors trained with "groups".

In summary, as illustrated in Figure 6.2, our system takes the heterogeneous visual information received from the 1D, 2D and 3D channels as inputs, and then leverage the deep neural networks to automatically learn the homogeneous deep activations. Building on the deep activation, our system further jointly learns the attribute detectors by leveraging graph-based constraints. These attributes enable zero-shot learning and further boost the action recognition.

Our Contributions: 1) we propose a uniform framework to learn homogeneous deep activations from the heterogeneous information sources. It is superior to most previous work on recognizing human actions from depth videos, which heavily relies on hand-designed low-level features. 2) Our system jointly learns attribute detectors by incorporating the attribute relation graph as constraints, which de-correlates some attributes, and as a result enables the detectors to "learn the right thing". The relation graph captures the semantic/geometric relationships among action attributes, which is superior to "groups" based constraints for action recognition. To the best

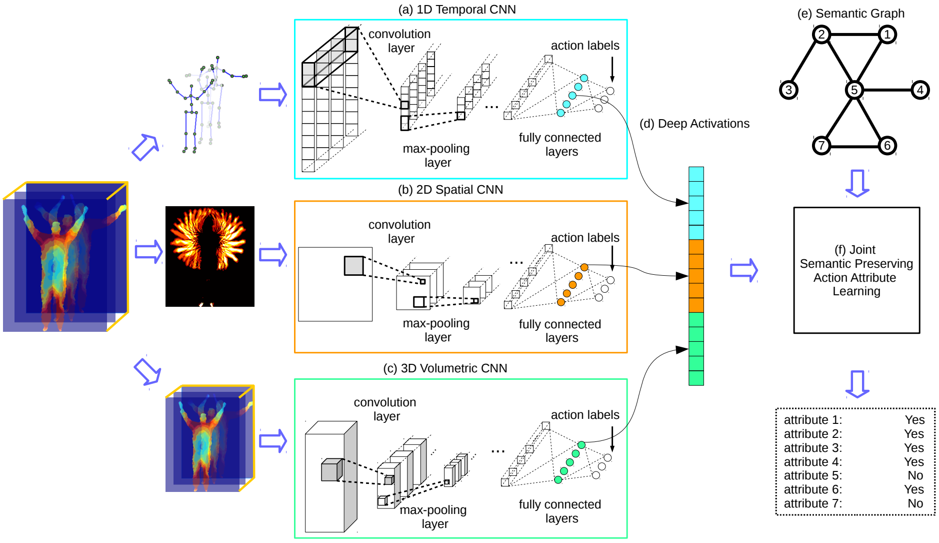


Figure 6.2: Overview of the proposed deep activation-based action attribute learning modeling. (a), (b) and (c) Multiple Convolutional Neural Networks are trained on different dimensional representations of the given depth videos such as 1D skeleton joint coordinates, 2D depth motion maps and 3D video volumes. The CNNs are trained in a supervised manner where action labels are used. (d) The second-last layer neuron activations from multiple CNNs are collected as Deep Activations. (e) and (f) Semantic preserving joint attribute learning algorithm is proposed by leveraging the prior knowledge of relations among attributes.

of our knowledge, this chapter is the first to leverage deep learning features to jointly learn action attribute detectors constrained on the relation graph to de-correlate attributes for action recognition from depth videos. The proposed algorithm are evaluated on three benchmarked datasets, and experimental results demonstrate the effectiveness of the proposed framework by achieving the state-of-the-art performances on both attribute detection and zero-shot action recognition.

6.1 Architecture of Learning Deep Activations

This section elaborates the architecture of each deep CNN in our multi-stream deep neural network framework. An overview of three types of CNN architectures is illustrated in Figure 6.3. Note that the numbers of dimensions in this figure are trained on the MSR Action3D dataset [79]. For different datasets, these numbers may vary. [1]

1D Representation: In the 1D-Temporal-CNN model, the input is a 1D sequence where the dimension is the frame number of the depth video. Each element in the sequence represents the skeleton joints in corresponding depth frame. Each coordinate of a skeleton joint is compared with 1) its two counterparts in the previous and initial frame and 2) the anchor joint in the current frame and the differences are used for representation. Thus the dimension of each joint is 6 and for each skeleton is 120 (20 joints). An

[1] The actual numbers of dimensions shown in the figure may vary in different datasets. Here the numbers of our models trained on MSR Action3D dataset [79].

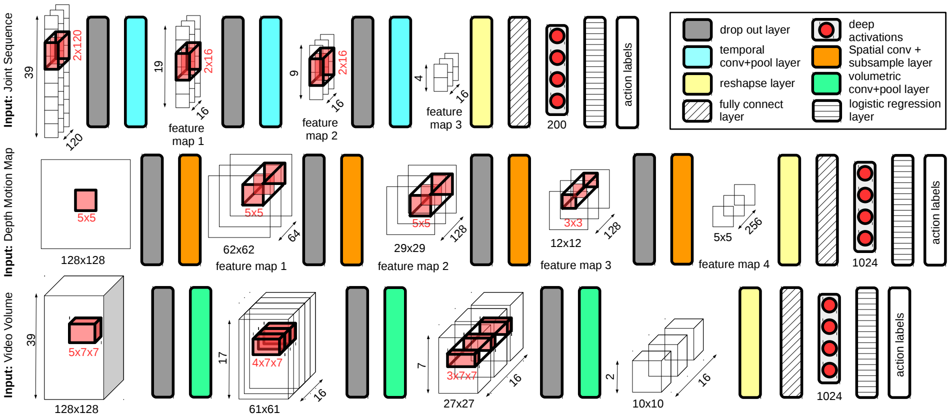


Figure 6.3: Overview of architectures for each of deep CNNs employed in proposed algorithm. Top row is for 1D-Temporal-CNN and the middle and bottom rows are for 2D-Spatial-CNN and 3D-Volumetric-CNN. Legend for layers is shown in the top-right corner. Convolution filters of each layer are shown as red cubes or rectangles. Dimensions of feature maps, deep activations and filters are shown accordingly.

abstract feature extraction layer is composed by one temporal convolution layer and one max-pooling layer. Three abstract layers and an additional 3-layer multilayer perceptron (MLP) are added. The deep activation layer here denotes the second layer in the MLP, which is composed by abstract features learned from input and supervised by its action label.

2D Representation: To capture the spatial energy distribution of an action, Depth Motion Map (DMM) [169] is employed for each depth sequence as the 2D representation. The input of 2D-spatial-CNN is a 128×128 depth motion map that characterizes the spatial movement during the whole action. Then 4 abstraction layers are employed before the MLP.

3D Representation: In many deep learning based action recognition algorithms [74, 60], the spatial-temporal video volumes can also be a representation *per se*. In our work, the depth spatial-temporal 3D volume itself is used as the 3D representation. The input of 3D-Volumetric-CNN is a $128 \times 128 \times T$ ($T = 39$ in Figure 6.3) tensor which is the normalized video volume itself. The filters are also 3D-tensors which are applied on the spatial-temporal subvolumes of the depth video to extract features. More implementation details for 1D, 2D and 3D representations can be found in the appendices. In this work, each CNN is trained individually in a supervised manner. By collecting deep activations learned from multiple representations, the deep activations are desired to be discriminative from different aspects. Another benefit of using multiple representations is that it can alleviate the demand of a vast amount of training data for deep CNNs [178].

We also apply drop-out layers together with each of the convolution layers in all deep CNN models to avoid feature co-adaptation. The idea of drop-out is proposed by Hinton *et al.* [47] to randomly zero some of the neuron units during training phases. The drop-out layers can effectively avoid the overfitting caused by complex co-adaptations, where feature detectors are only helpful with a certain internal context. CNNs with random drop-out layers show improvements on speech and object recognition benchmarks, and better generalization without using very large training data. The CNNs in our work are trained in a supervised manner, while the ground truths are action labels. Different with the framework in [178], which directly learns CNNs on attribute labels, the CNNs of our framework are trained to learn action discriminative deep activations without the involvement of action attributes. The main reasons are two-fold. On one hand, training CNNs directly on action labels can ensure the learned activations are action discriminative. On the other hand, semantic relations between attributes are difficult to be directly embedded into a CNN. More favored structure must be designed to learn action attributes. The deep activations are the activations in the middle layer of MLP in each CNN. We collect all deep activations together as the final output of the multi-stream deep CNNs for each depth video sequence. For instance, as illustrated in Figure 6.3, the final output of the tri-stream model is a $200 + 1024 + 1024 = 2248$ dimensional activation vector.

6.2 Semantic Preserving Multi-task Action Attribute Learning

This section firstly discusses the characteristics of relations among action attributes and the similarity/difference with object attributes. Then the formulation of the joint semantic preserving action attribute learning problem together with an efficient solution is introduced.

6.2.1 Semantic Relations among Action Attributes

Attribute learning is a popular topic in object recognition and face recognition [34, 69, 131]. While modeling co-occurrence between attributes is helpful in object recognition, attribute learning with de-correlating attribute pairs can prevent excessive biasing the likelihood function on the training set [55]. In action recognition, the benefits of using action attributes have also been initially explored in recent years [83]. However, as most of previous methods in object attribute recognition, action attributes are often learned independently without considering the relations among action attributes. In this chapter, we resolve this problem by embedding the relationships among action attributes into a joint multi-task attribute learning formulation.

As object attributes are often fine-grained and have simple semantic relations, simple grouping is often enough to capture the essential information. However, action attributes have more complicated semantic relationships than object attributes, thus need a more suitable structure. Human action attributes often involve one or more body parts, therefore a natural connection would be built on the body parts that the attributes involve. For example, as illustrated in Figure 6.1, the attribute “arm below torso” is related to body parts “torso” and “arm”, so it is related to attributes “torso motion” and “arm motion”. In addition, since “arm below torso” is an attribute describing “*the position of upper limbs*”, it is related to other attributes of the same topic, such as “hand above head”. In this work, the relationships among attributes are represented by an undirected graph. An example of such graph is given in Figure 6.1 and more detailed semantic graphs can be found in the appendices.

6.2.2 Joint Attribute Learning

As suggested in [83], we manually define a number of attributes as well as their correspondences between each action class. The protocol to label these attributes is based on motions and relative positions of body parts. Therefore from the ground-truth action class labels, we obtain the attribute labels for each training sample. 1 is used to indicate that the attribute is “active” and -1 otherwise. In the following, the formulation the joint attribute learning problem as a multi-task learning problem is proposed.

Formulation: Suppose there is a set of training samples $X \in \mathbb{R}^{M\times N}$ and corresponding attribute labels $Y \in \{-1,1\}^{K\times N}$, where each column $X_{i\in[1,N]}$ of X is a learned deep activation and each column Y_i in Y is X_i's attribute label. M is the deep activation dimension, N and K are the numbers of training samples and number of defined attributes, respectively. The objective is to learn a matrix $W \in \mathbb{R}^{M\times K}$. Each column $W_{k\in[1,K]}$ in W is the parameter of the corresponding attribute predictor where $W_k^T X_i = Y_i^k$.

Therefore, learning the optimal W is to minimize the following problem:

$$W^* = \underset{W}{\text{argmin}}\ \mathcal{L}(X,Y,W) + \mathcal{O}(W), \tag{6.1}$$

where $\mathcal{L}(X,Y,W)$ is the empirical loss function of predicting attribute labels. In this work, we use $\|W^T X - Y\|_F^2$ as our loss. And $\mathcal{O}(W)$ is a regularization term on W to pursue some prior structures such as sparsity.

What are the desired properties of W? Since deep activation vector is discriminative on action labels and each one has the potential to describe a semantic concept, so an attribute should have sparse response to the deep activation vector. As suggested by [55] and [32], the group sparsity enforced by $l_{2,1}$ norm plays an important role in feature selection. Secondly, to preserve the semantic relationships among attributes, the attributes that are semantically close should share features while distant ones should compete for features. We advocate this property by using the *graph Laplacian* of a predefined attribute graph.

By putting all the concerns aforementioned together, the problem of semantic preserving joint attribute learning can be formulated in the following shape:

$$W^* = \underset{W}{\text{argmin}}\ \|W^T X - Y\|_F^2 + \lambda\|W\|_{2,1} + \beta tr(WLW^T), \tag{6.2}$$

where λ and β are the weights for the row-sparsity and semantic preserving regularizers, respectively. The first term is the empirical loss for predicting attribute labels. The second term introduces row-sparsity to the learned weight matrix, which avoids overfitting and includes feature-selection. The third term models the relationships among attribute weight vectors based on the graph.

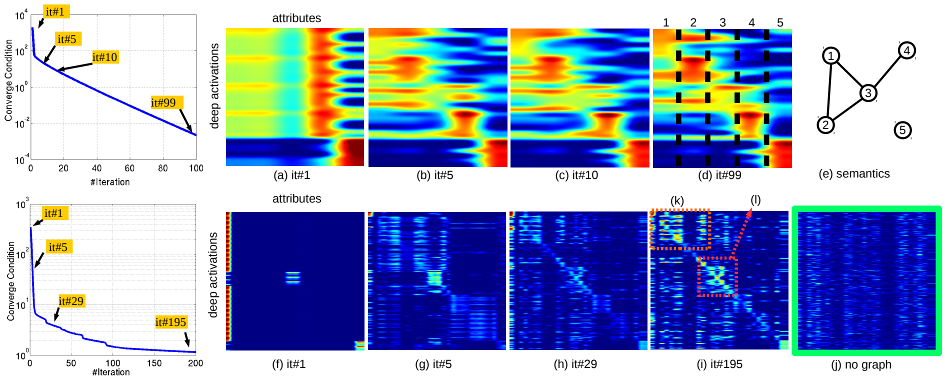
Optimization: To efficiently and effectively solve the problem (6.2), two auxiliary variables are introduced to make the problem separable, which give the following program:

$$\begin{aligned}&\min_{W,P,Q} \|P^TX - Y\|_F^2 + \lambda\|Q\|_{2,1} + \beta tr(WLW^T)\\ &s.t. \quad W = P, \quad W = Q.\end{aligned} \tag{6.3}$$

The program in Eq. (6.3) can be solved in an unconstrained form by the dual ascent method. To bring robustness to the dual ascent method, we use Augmented Lagrangian methods (ALM) to generate the augmented Lagrangian for (6.3):

$$\begin{aligned}\mathcal{L}_\rho(X, Y, W, P, Q) = \|P^TX - Y\|_F^2 &+ \lambda\|Q\|_{2,1}\\ &+ \beta tr(WLW^T) + \langle Z_1, P - W\rangle\\ &+ \frac{\rho}{2}\|P - W\|_F^2 + \langle Z_2, Q - W\rangle\\ &+ \frac{\rho}{2}\|Q - W\|_F^2,\end{aligned} \tag{6.4}$$

where Z_1 and Z_2 are Lagrangian multipliers associated with the two constraints in Eq. (6.3), and ρ is a positive penalty. Since the program in Eq. (6.4) is separable, we can apply the *alternating direction method of multipliers* (ADMM) [10] strategy. The solutions of the sub-problems based on ADMM are shown as follows:



(a) it#1 (b) it#5 (c) it#10 (d) it#99 (e) semantics

(f) it#1 (g) it#5 (h) it#29 (i) it#195 (j) no graph

Figure 6.4: Illustration of the effect of Algorithm 4 on a synthetic dataset with 5 attributes (the top row) and MSR Action dataset with 30 attributes (the bottom row). (a-d) and (f-i) are learned weights for sampled iterations. Columns correspond to attributes and rows correspond to features or deep activations. Warmer colors indicate higher absolute values in weight matrix, the more the attribute relies on the feature. (e) The underlying semantic graph of the synthetic dataset. (j) The result generated without graph involved for comparison. (k) and (l) show two examples of graph-guided effects, please see text for details.

W sub-problem: With unrelated terms discarded, this sub-problem becomes a classic least squares problem and the optimal $W^{(t+1)}$ can be calculated easily by:

$$\begin{aligned} W^{(t+1)} &= \underset{W}{\text{argmin}}\ \mathcal{L}_\rho(W, P^{(t)}, Q^{(t)}) \\ &= \underset{W}{\text{argmin}}\ \beta tr(WLW^T) + \frac{\rho}{2}\|P^{(t)} - W + u_1^{(t)}\|_F^2 \\ &+ \frac{\rho}{2}\|Q^{(t)} - W + u_2^{(t)}\|_F^2 \\ &= \rho(P^{(t)} + Q^{(t)} + u_1^{(t)} + u_2^{(t)})(2\beta L + 2\rho I)^{-1}, \end{aligned} \tag{6.5}$$

where $u_1^{(t)} = (1/\rho)Z_1^{(t)}$ and $u_2^{(t)} = (1/\rho)Z_2^{(t)}$ are scaled dual variables which make the representation more compact by combining linear and quadratic terms. Note that the matrix inverse $(2\beta L + 2\rho I)^{-1}$ only needs to be computed once.

P sub-problem: Similar to the W sub-problem, the P sub-problem is also a classic least squares problem:

$$\begin{aligned} P^{(t+1)} &= \underset{P}{\text{argmin}}\ \mathcal{L}_\rho(X, Y, W^{(t+1)}, P, Q^{(t)}) \\ &= \underset{P}{\text{argmin}}\ \|P^TX - Y\|_F^2 \\ &+ \frac{\rho}{2}\|P - W^{(t+1)} + u_1^{(t)}\|_F^2 \\ &= (2X^TX + \rho I)^{-1}[2X^TY + \rho(W^{(t+1)} - u_1^{(t)})]. \end{aligned} \tag{6.6}$$

Please note that the terms $(2X^TX + \rho I)^{-1}$ and $2X^TY$ also need to be computed only once.

Q sub-problem: The closed form solution of $Q^{(t+1)}$ can be obtained by:

$$\begin{aligned} Q^{(t+1)} &= \underset{Q}{\text{argmin}}\ \mathcal{L}_\rho(W^{(t+1)}, P^{(t+1)}, Q) \\ &= \underset{Q}{\text{argmin}}\ \lambda\|Q\|_{2,1} + \frac{\rho}{2}\|Q - W^{(t+1)} + u_2^{(t)}\|_F^2 \\ &= \mathcal{S}_{\frac{\lambda}{\rho}}^{2,1}(W^{(t+1)} - u_2^t), \end{aligned} \tag{6.7}$$

where $\mathcal{S}_{\epsilon>0}^{2,1}(\cdot)$ represents the shrinkage operator [82].

In addition, the two scaled dual variables u_1 and u_2 need to be updated using corresponding residuals:

$$\begin{aligned} u_1^{(t+1)} &= u_1^{(t)} + P^{(t+1)} - W^{(t+1)} \\ u_2^{(t+1)} &= u_2^{(t)} + Q^{(t+1)} - W^{(t+1)}. \end{aligned} \tag{6.8}$$

For clarity, we summarize the optimization procedure of the deep activation-based attribute learning algorithm (DAAL) in Algorithm 4. The algorithm terminates when $(\|P^{(t)} - W^{(t)}\|_F + \|Q^{(t)} - W^{(t)}\|_F) \leq \delta(\|P^{(0)} - W^{(0)}\|_F + \|Q^{(0)} - W^{(0)}\|_F)$ where $\delta = 10^{-5}$, or when the predefined maximal number of iterations is reached.

```
Algorithm 4: DAAL
  Input: Deep Activation Matrix $X$, Attribute Ground-truth $Y$
  Initialization: Randomly initialize $W^{(0)}$,$P^{(0)}$,$Q^{(0)}$, Set $u_1^{(0)}$ and $u_2^{(0)}$ to
                  be zero matrices. $\rho = 1.5$,$t = 0$
  while not converge do
      update $W^{(t+1)}$ via Eq. (6.5)
      update $P^{(t+1)}$ via Eq. (6.6)
      update $Q^{(t+1)}$ via Eq. (6.7)
      update $u_1^{(t+1)}$, $u_2^{(t+1)}$ via Eq. (6.8)
      $t = t + 1$
  end
  Output: Optimal solution $W^*$ = $W^{(t)}$
```

Table 6.1: Attribute detection scores (mean average precision) and zero-shot action recognition rates on three benchmark datasets, higher is better.

Tasks	Detection scores (MAP)						Zero-shot learning (%)		
Datasets	MRA		UTA		MRP		MRA	UTA	MRP
Methods	S	U	S	U	S	U	l2o	l2o	l2o
no-regularize	.4057	.4913	.4880	.3620	.5305	.5254	50.27	53.40	55.89
lasso	.8283	.5105	.9473	.4293	.9894	.6414	67.82	80.94	93.28
all-sharing [32]	.4291	.4794	.6085	.3989	.5590	.5809	49.48	73.07	81.82
group-lasso [55]	.9356	.5236	.9051	**.4329**	.9985	.6405	70.81	81.53	93.27
proposed	**.9667**	**.5356**	**.9687**	.4304	**.9994**	**.6426**	**72.03**	**81.89**	**94.69**

6.3 Experimental Results

Effectiveness of the Algorithm: To better understand our joint attribute action attribute learning process, a simulation is conducted on five attributes with 1000 features. The semantic relationships among these attributes are shown in Figure 6.4 (e). One can consider the attributes to be {1: *"arm-upward motion"*, 2: *"arm-downward motion"*, 3: *"arm-motion"*, 4: *"arm below torso"*, 5: *"leg motion"*}. Learned weights for sample iterations are shown in Figure 6.4 (a) to (d), from which we can observe that the semantic relationship among attributes are more obvious with more iterations, note that warmer colors indicate higher absolute weights and each column corresponds to the weight vector for an attribute. In (d), attribute "1","2" and "3" share many features, "3" and "4" share some features and "5" barely shares features with other attributes.

In addition, a similar experiment is conducted on a real dataset, MSR Action Dataset [79]. 284 samples are used with 2248 deep activations and the activation-attribute map is visualized for sampled iterations in Figure 6.4 (f) to (i). In (k), we show the learned pattern showing that *"arm in-front-of torso"* and *"arm above head"* tend to share features with arm-related motions while (l) *"arm below torso"* tends to share features with torso related motions. For comparison, the weights learned on the same set of features without graph involved are illustrated in (j).

6.3.1 Experiment Setup and Datasets

Datasets: There are three datasets for depth based action recognition used in the experiments, including the MSR Action 3D dataset [79] (**MRA**), the UTA Action 3D dataset [162] (**UTA**) and the MSR Action Pairs dataset [103] (**MRP**). The MRA dataset contains 20 gaming actions, such as "two arms waving" and "golf swing". Each action is performed by 10 different subjects and the subjects perform each action 2 to 3 times in the same location. The UTA dataset contains 10 actions which cover movements of hands, arms, legs and upper torso. Each action is performed by 10 different persons. The MRP dataset contains 6 pairs of actions that each pair of actions has opposite temporal orders, such as "push chairs" and "pulling chairs". Different from the MRA dataset, UTA and MRP allow the subjects moving around while performing actions. We define 30 action attributes for MRA, 19 for UTA and 16 for MRP, where they share some common attributes such as "arm-motion", *etc*.

Deep Activations: For the MRA and MRP datasets, since the skeleton joint locations are available, we apply all three streams of deep CNNs as illustrated in Figure 6.3. For the UTA dataset, only the DMM-based 2D CNN and the video-volume-based 3D CNN streams are applied because the skeleton joints are not available. For the MRP and UTA datasets, since the temporal order plays an important role and the 2D representations is temporal order invariant, multiple CNNs are trained for 2D representations following the idea of temporal pyramid. For thorough lists of action attributes, their relationships, and CNNs used in each dataset, please refer to the appendices.

Baselines: For attribute detection and zero-shot learning, the proposed method is compared to four related baselines. All empirical loss functions are same as in Eq. (6.2) for uniformity. The four baselines include (1) "non-regularize": is single-task learning using least-squares loss without any regularization term. (2)"lasso" is l_1-regularized. (3)"all-sharing" is a multi-task learning method with $l_{2,1}$-regularized. (4) "group-lasso" is using the same regularize terms as in [55]. We set the default parameter values of λ and β for each baseline (if existed) to 1.

6.3.2 Attribute Detection and Zero-shot Action Recognition

This section shows the evaluation of the proposed joint attribute learning method on all three datasets with two tasks: 1) attribute detection and 2) zero-shot action recognition using only the learned attributes. For the first task, we employ two splitting ways for training and testing sets: 1) "Seen": this is the same as the "cross-subject" splitting protocols as in [154], [162] and [103], where half of the subjects are used for training and the remaining half for testing. All action classes appear during training. 2) "Unseen": the protocol introduced in [83] as "leave-two-out" scheme, where all combinations of action classes are considered. Since some combinations may contain attributes that do not appear in the training set, we leave these combinations out and keep the rest. For the MRA dataset, there are total of 104 combinations which fulfill this condition. For the UTA dataset, there are 20 such combinations and for the MRP dataset there are 64 combinations. Since the training on the 3D volumetric deep CNNs is time-consuming, we

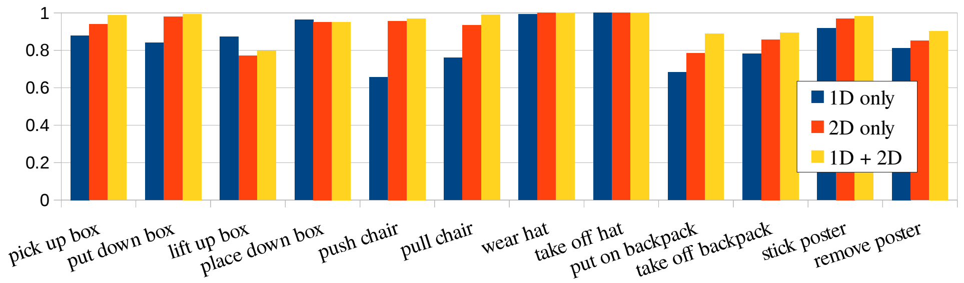


Figure 6.5: The average accuracies of zero-shot action recognition test on the MRP dataset using deep activations based on 1D, 2D and both representations.

only train CNNs for each combination using 2D spatial and 1D spatial models for "Unseen" tasks, if available.

Table 6.1 shows the action attribute detection results in terms of mean average precisions (MAP) and zero-shot action recognition in terms of recognition rates. Our method outperforms other baselines in most tests. The poor results obtained by "no-regularize" indicate that the training process is easy to overfit. We observe that "group-lasso" performs stably during all tests while "all-sharing" and "lasso" do not always perform well, which suggest that solely pursuing sparsity may result in biased attribute estimations. Preserving the semantics in attributes is beneficial for attribute detection. In most cases, our method significantly outperforms "group lasso" which is proposed in [55]. This is because our method is more suitable in modeling relationships among action attributes. The right panel of Table 6.1 lists the zero-shot learning action recognition results. Our method generalizes well in all datasets when dealing with zero-shot action learning, which demonstrates that our method learns better and more discriminative attribute vectors. By comparing attribute learning results and zero-shot learning results, we notice that higher MAP scores in attribute detection may not necessarily lead to better classification results in zero-shot action recognition, especially when they are very close.

Figure 6.5 shows the class-wise average accuracies comparison of using deep activations learned from 1D, 2D, and 1D+2D CNNs. We observe that deep activations learned from multi-stream CNNs perform better than single streams. It is interesting to observe that 2D model performs better than 1D model except for action pair "lift up box" and "place down box", since this pair of actions involves drastic motion "bent", which is easy for joint-based models.

6.3.3 DAAL Boosting Action Recognition

In this section, we compare the action recognition accuracies with some state-of-the-art methods. All the results for three datasets are shown in Tables 6.2,6.3, and 6.4. In all experiments, the same protocols used in [154], [162] and [103] is followed, where half of the subjects are used for training and the other half of subjects for testing. We evaluate deep activations, learned action attributes and their combination.

From Tables 6.2,6.3, and 6.4, one can notice that the deep activation vectors are very discriminative, which demonstrate the effectiveness of our

Methods	Accuracy
Bag of 3D points [79]	74.70%
HOJ3D [163]	79.00%
STOP [150]	84.80%
ROP [155]	86.50%
Actionlet [154]	88.20%
HON4D [103]	88.89%
DSTIP [162]	89.30%
Pose Set [153]	90.00%
SNV [168]	91.64%
Moving Pose [173]	91.70%
deep act. (Ours)	92.30%
attr. (Ours)	87.18%
deep act. + attr. (Ours)	**93.40%**

Table 6.2: Comparison of action recognition rate on MSR Action 3D with other methods using the protocol in [154].

Methods	Accuracy
Posture Word[162]	79.57%
DSTIP [162]	85.80%
deep act. (Ours)	86.87%
attr. (Ours)	78.79%
deep act. + attr. (Ours)	**87.88%**

Table 6.3: Comparison of action recognition rate on UTA Action 3D dataset with other methods using the protocol in [162].

multi-stream deep architectures. Compared to previous features, the learned attributes are very compact (only 16∼30 dimensions) and discriminative in action labels. By combining the learned activations and attributes together, our proposed framework achieves the best performances on all three datasets, because the attributes transfer knowledge from other classes to further complete the information for action classification.

6.3.4 Evidence of Learning the Right Things

In this section, we conduct an experiment to show what the attribute learner learned from deep activations. For visualization purpose, only 2D representations are used in this experiment. Given a 2D representation, we use a patch with random values to occlude it and use pre-trained deep CNN to generate the deep activation vector. Then the learned attribute classifier is employed to generate an attribute vector. By comparing the generated attribute vector with the ground-truth attribute vector, we propagate the deviation to the locations where the patch is. By densely sampling multi-scale occlusion patches, we can accumulate an error map, which implies the responsible region of every attribute. Some results are shown in Figure 6.6. By comparing results generated by “group lasso” (top) and ours, our method locates more accurately for regions responsible for a specific action attribute than “group lasso”. For example, in action “hand clapping”, the attribute detector for “arm-motion” in “group-lasso” concentrates on the

Methods	Accuracy
Skeleton + LOP [154]	63.33%
[154] + Pyramid	82.22%
HON4D[103]	97.67%
SNV [168]	98.89%
deep act. (Ours)	98.89%
attr. (Ours)	87.22%
deep act. + attr. (Ours)	**99.44%**

Table 6.4: Comparison of action recognition rate on MSR Action Pairs dataset with other methods using the protocol in [103].

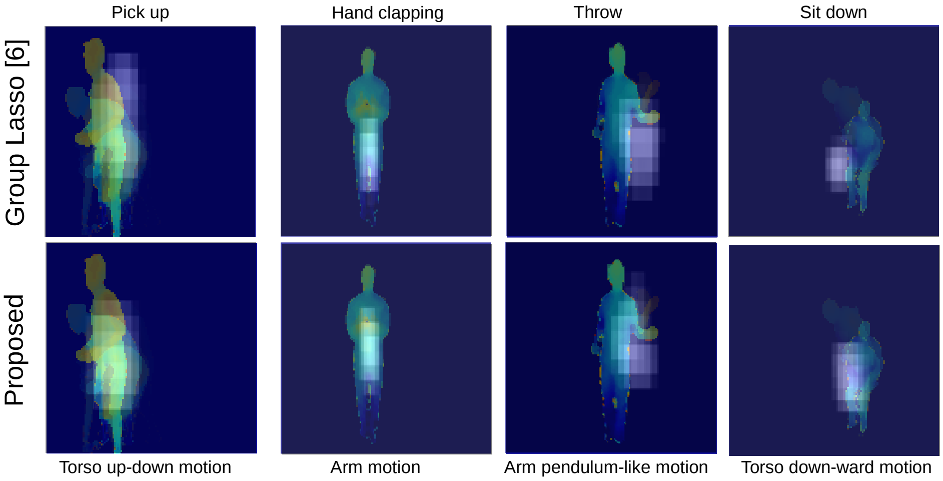


Figure 6.6: Sample results showing the responsible regions for attributes from UTA dataset. The top row shows the results generated by [55], the bottom row shows ours.

lower-body and ours covers more on the hand area. This experiment further demonstrates that our proposed method is more suitable for feature selection in action attribute learning and it can locate the right part for a specific attribute.

6.4 Discussion

The main direction of our future work will be to develope a more sophisticated and suitable cost function which can integrate the attribute learning cost to the network training. Therefore the whole network can be re-trained more effectively. In addition, the architecture of the networks is still not deep enough, which limits the potential of learning more powerful representations. In the future work, the deep activation learning phase and the attribute learning phase will be integrated together into a holistic neural network which can be tuned end-to-end. Finally, the deconvolutional techniques will be also integrated into our framework to shed light on the learned saliency maps as shown in Figure 6.6.

Chapter 7

Deep Learning-based Video Content Description

In previous chapters, we have discussed about how to model the mappings from visual signals (images or videos) to a discrete set of pre-defined labels. In this chapter, we focus on automatically generating meaningful and informative human-level English descriptions of input visual data. More specifically, we propose a multi-channel sequence-to-sequence video captioning framework based on recurrent neural networks.

Automatic visual content understanding and describing have become a fast-growing research area in computer vision for the recent decade. Effective understanding visual medias can significantly improve the performance of computer programs to automatically analyze and organize the online media. With the recent ground-breaking progress in large-scale visual recognition and deep neural networks, an explosive amount of techniques have been proposed in object recognition [23, 122], scene understanding [33, 41] and action recognition [74, 143]. These findings successfully broaden the horizon of visual recognition research. Combining with the rapid progress of natural language processing, visual content describing has drawn more and more attention in the field of computer vision and machine learning. How to bridge the gap between visual content and natural human language has become the motivation of many research topics, such as image and video captioning.

Automatic image captioning deals with both images and textual data and generates natural sentences to summarize input image content. Generating descriptive sentences for images requires knowledge from multiple domains such as computer vision, natural language processing, and machine learning. Inspired by the recent renewed interests in deep learning techniques, there are many image captioning frameworks proposed [171, 15, 90, 59, 151, 14]. The paradigm for generating captions for images takes two steps: 1) **Encoding stage:** the visual input (an image) is processed by a feature extraction layer (encoder). 2) **Decoding stage:** a language model is applied to decode the input feature encoding to a pre-defined vocabulary. The output sentence is generated based on the probabilistic distribution over the vocabulary using the language model. Recurrent neural network (RNN) has been proven to be an effective choice for the decoder because RNN is capable to address the temporal dynamics in output sentences.

Video captioning is a similar problem with image captioning and the encoder-decoder framework is also applicable for this problem. However,

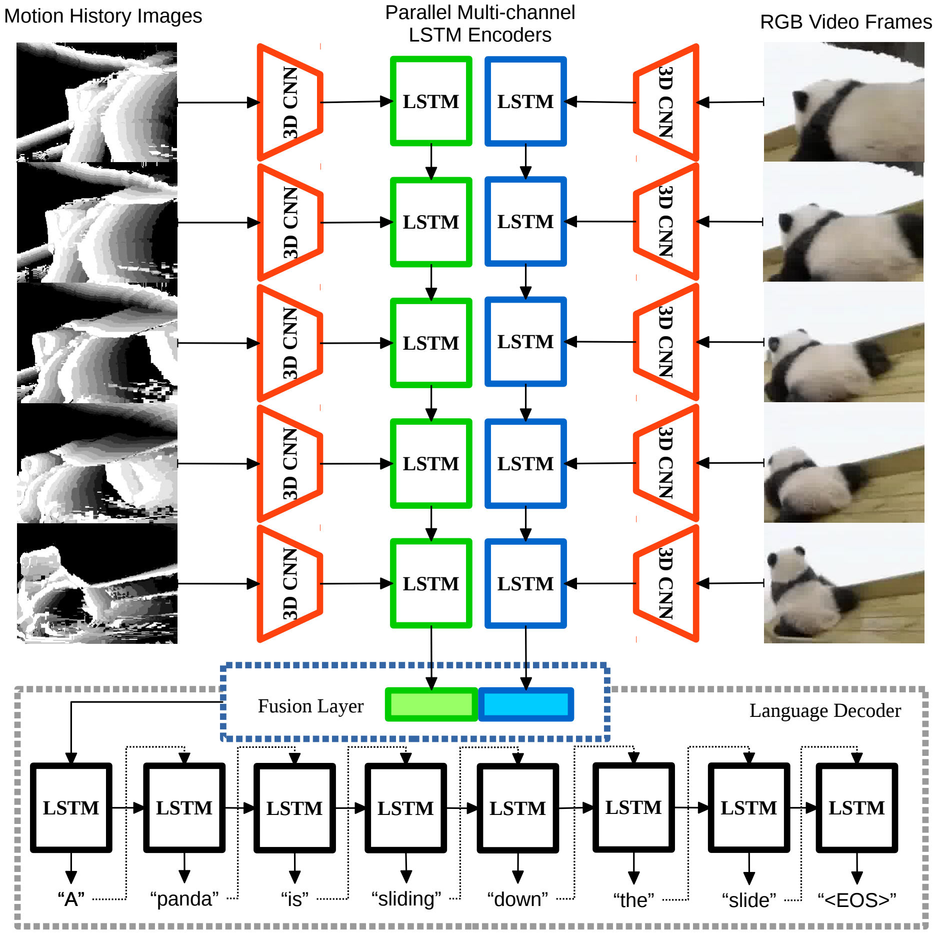


Figure 7.1: Illustration of our proposed video captioning framework. Two channels of input frames are utilized: motion history images (MHIs) and RGB video frames. Firstly, raw features are extracted from each input channel frames using 3D convolutional neural networks. The feature extraction phase generates sequential features of arbitrary lengths. Secondly, the sequence of features is encoded using RNNs with LSTM cells for each channel. Then a fusion layer is employed to combine the encoded features from both LSTM encoders. Finally the fused features are fed into a LSTM-based language decoder to be decoded into a sequence of words. "<EOS>" represents the "*end of sentence*" token.

different from static images, videos contain much more semantic information related to temporal dynamics. In [1], the authors explored to automatically assign conceptual tags to YouTube videos by learning from both visual and audio features. The authors of [42] treated the problem as an activity-recognition problem. They built hierarchical semantic trees to organize detected entities such as actors, actions, and objects. Zero-shot-learning-based language models were applied on the learned hierarchies to assign a short sentence to summary the detected potentials. Similarly, semantic triplets (*subject-verb-object*) were also used in [65] to organize detections of objects and activities for sentence inferencing. Quadruples were utilized in [138] to include more information from the context and scene for more accurate descriptions. Other efforts made to improve the performance of automatic tagging include video tag augmentation [98], video clustering [51], and video re-ranking [157]. Inspired by the successful utilization of LSTM-based RNNs in image captioning, there has been a lot of work using

RNNs for video captioning. In [149], Venugopalan *et al.* proposed to apply average pooling over image features extracted from each video frame to obtain a video feature. Then the video feature was encoded to feed into a LSTM-based RNN language model for sentence decoding.

In this chapter, we propose a novel framework for video captioning task. The main idea is illustrated in Fig. 7.1. To include more temporal motion-related information from the input video sequences, two channels (motion history images and raw video frames) are employed as video inputs. Our proposed framework integrates three different types of neural networks to perform automatic video captioning:
1) **3D-CNN:** instead of using object-detection-oriented feature extraction networks (such as VGG and AlexNet), we employ 3D convolutional neural networks (3D CNNs) to extract spatial-temporal features from video clips.
2) **RNN Encoder:** since the length of each video is arbitrary, the generated 3D CNN features are also of arbitrary lengths. A recurrent neural network (RNN) with long-short-term-memory (LSTM) cells is employed to map the sequential inputs to a fixed-dimensional encoding space. To jointly learn the encoding from two input channels, one LSTM encoder is assigned to each channel and the two encoders forms a parallel system. The fusion layer is a fully-connected layer which maps the LSTM internal states to the encoding space and the encoded vectors are concatenated.
3) **RNN language model:** the RNN language model defines a probability distribution of the next word in a sequence based on both the context and the current word. In our model, the context encoded in the form of LSTM internal state and initialized by the learned encoding vector.

In addition, we also explore the potential utilization of the proposed video captioning framework in automatic video-based American Sign Language (ASL) translation. ASL is a visual gestural language which is used by many people who are deaf or hard-of-hearing. Automatically generating textual descriptions from ASL videos can significantly benefit the ASL-using population to communicate with non-ASL users. To the best of our knowledge, there has been no such effort to link ASL translation with video captioning before. We have collected a large-scale dataset from YouTube uploaded by ASL signers and gained annotations by aligning the video clips with subtitles. The proposed network is able to gain ASL-oriented knowledge from the dataset and to generate meaningful sentences from ASL videos.

The contributions of this work have three aspects:

- A sequential LSTM encoder framework is proposed to learn to embed video sequences addressing both spatial and temporal information.

- Our framework can handle multiple streams of input sequences and automatically learn how to combine.

- We are the first to explore video captioning in the area of ASL translation and provide a novel dataset in this area.

The rest of this chapter is organized as the following. Section 7.1 elaborates the architecture of the proposed framework. Then the datasets used and proposed by this chapter are described in Section 7.2. Section 7.3 discusses the experiments.

7.1 Method

The framework of our proposed method is illustrated in Fig. 7.1. The whole framework is composed of four core modules: 1) 3D CNN-based feature extractor. 2) Sequential feature encoder. 3) Parallel fusion layer. 4) Sentence-generation language module. Both the feature encoder and the language module are based on RNNs with LSTM cells.

7.1.1 LSTM-based RNNs

Recurrent neural network (RNN) is a category of neural network containing an internal state. RNN is able to encode a dynamic temporal behavior due to its connections between units form directed cycles. The internal state of RNN can be treated as a state of memory, which contains information of both current input and the previous memory. Therefore, RNN has the capability to "remember" the history of both previous inputs and outputs. RNN is widely applied in prediction frameworks which is dependent on context, such as machine-translation [2]. A RNN cell can be formatted as:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t), \tag{7.1}$$

where h_t and x_t denote the hidden state and input encoding at time step t, respectively; W_h and W_x denote the parameters assigned to each state vector. $\sigma(\cdot)$ denotes the sigmoid function.

However, RNN often suffers from modeling long-term temporal dependencies [7]. A modification called *long-term-short-memory* (LSTM) is proposed for better long-term temporal dependency modeling with more sophisticated internal states and connections. A typical LSTM cell can be formatted as:

$$\begin{aligned}
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
\hat{C}_t &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\
C_t &= f_t \odot C_{t-1} + i_t \odot \hat{C}_t \\
h_t &= o_t \odot tanh(C_t),
\end{aligned} \tag{7.2}$$

where $\odot$ is element-wise product; $\sigma(\cdot)$ denotes the sigmoid nonlinearity-introduce function; x_t is the input encoding at each time step t to the LSTM cell; W_i, W_f, W_c, W_o, U_i, U_f, U_c, and U_o are weight matrices assigned to parameters of input gate, forget gate, cell state and output gate, respectively; b_i, b_f, b_c and b_o are bias vectors for corresponding gates and states; i_t, o_t, f_t, C_t and h_t denote the state values of input gate, output gate, forget gate, cell state and hidden state, respectively. $\hat{C}_t$ represents the candidate cell state before combining with the previous cell state (C_{t-1}) and the forget gate.

In our work, the LSTM cells are the building blocks of two types of RNNs: 1) feature encoding RNN and 2) sentence decoding RNN (language model). The illustrations of both RNNs are shown in Fig. 7.2. The two types of RNN cells are connected as illustrated in Fig. 7.1. The feature encoding RNN is responsible to encode the sequential inputs from video features;

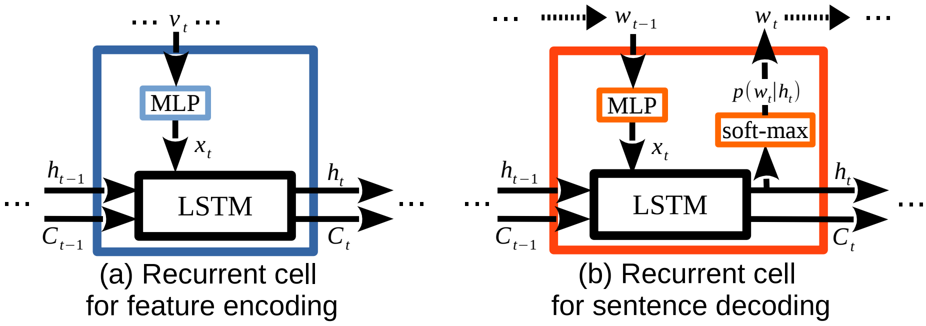


Figure 7.2: Illustration of two types of recurrent cells for feature encoding and sentence decoding, respectively. Both cells contain an internal LSTM cell. At each time step, feature encoding recurrent cell takes an input video feature (v_t) and sentence decoding cell takes an input as the word-prediction (w_t) from the previous time step. Note that the MLPs in both cells act as look-up tables which map the input vector to the internal input vector (x_t).

and the sentence decoding RNN is responsible to decode the output from encoding RNN to a sequence of words.

7.1.2 Feature encoder

Suppose the input video sequence $V = \{c_1, c_2, ..., c_T\}$ is composed of T short video clips. Without loss of generality, the length of each video clip $\|c_i\|$ could equal to 1 to represent individual frames. The video sequence can be encoded with a feature extractor ϕ (such as C3D [143] and VGG-net [128]), thus the video can be represented as: $\phi(V) = \{v_1, v_2, ..., v_T\}$, where $v_t = \phi(c_t)$ denotes a video feature vector for a video clip.

Therefore, the input video can be encoded into a sequence of feature vectors $\{v_t\}$. For the feature encoding RNN as illustrated in Fig. 7.2 (a), one video feature v_t is fed into the RNN cell with a multiple-layer-perceptron (MLP). The MLP can represent any multi-layer neural network, and in our case the MLP indicates a fully-connected layer followed by a ReLU layer. Note that the MLP acts like a look-up table, mapping the input feature vector into a continuous RNN embedding space. At each time step t, the RNN cell takes input from both the previous cell and the video sequence; it encodes the input vectors using an internal LSTM cell and output hidden state h_t and cell state C_t to the next cell. The behavior of the internal feature encoding LSTM cell ($LSTM_{FE}$) can be formatted as:

$$[h_t, C_t] = LSTM_{FE}(h_{t-1}, C_{t-1}, MLP(v_t)). \tag{7.3}$$

Parallel fusion layer. Our framework is designed to handle video encodings from multiple channels of the input video, such as RGB frames and motion history images (MHI) as shown in Fig. 7.1. Because different channel of video encoding contains different information, each channel should have its own feature encoding so that the intrinsic characteristics can be encoded. In our framework, to connect the output encoding vectors from feature encoding RNNs and the input of sentence decoding RNN, a parallel paradigm to conduct the mapping is employed:

$$ENC(V) = MLP(h_T) \oplus MLP(h_T^{'}), \tag{7.4}$$

where $ENC(V)$ denotes the final video encoding of the input video V and $\oplus$ denotes vector concatenation; h_T and $h_T^{'}$ denote the final state vector of two streams of RNN encoders. Note that the dimension of $ENC(V)$ matches with the dimension of RNN encoding space in the language model decoder.

7.1.3 Language model

A general language model is usually designed to compute the probability of a sequence of words:

$$p(w_1, w_2, ..., w_K) = p(w_K|w_{K-1}, , , w_1) \cdot ... \cdot p(w_2|w_1) \cdot p(w_1), \tag{7.5}$$

where w_i is the i^{th} word in the output sentence.

In video captioning scenario, the language model is designed to compute the modified probability:

$$p(w_1, w_2, ..., w_K, Y) = p(w_K|w_{K-1}, , , w_1, Y) \cdot ... \cdot p(w_2|w_1, Y) \cdot p(w_1, Y), \tag{7.6}$$

where $Y = ENC(V)$ represents the encoded video.

In our framework, the language model is implemented with a RNN-based sentence decoder, as shown in Fig. 7.2 (b). More specifically, the RNN decoding cell at each time step computes the probability by providing the previous output words and the video encoding as following:

$$\begin{aligned} p(w_t|w_{t-1}, , , w_1, Y) &= p(w_t|h_t) = SM(h_t) \\ [h_t, C_t] &= LSTM_{LM}(x_t, h_{t-1}, C_{t-1}) \\ x_t &= \begin{cases} Y, & \text{if } t = 1 \\ MLP(\mathbf{1}(w_{t-1})), & \text{otherwise,} \end{cases} \end{aligned} \tag{7.7}$$

where $SM(\cdot)$ represents a soft-max layer and $\mathbf{1}(\cdot)$ denotes the 1-hot-vector representation of the word index. Note that the MLP learns the mapping from word-index to the RNN internal space. The output word w_t is sampled according to the probability distribution computed by the soft-max layer.

7.1.4 Video representation

In this section, the procedure of obtaining video representations, *i.e.* $\phi(V)$, is discussed.

Spatial-temporal feature extraction. In [149], the video representation is obtained from mean-pooling of static image feature vectors of each frame. However, videos are more than combinations of individual frame. Only including static image features can capture the visual appearance such as objects and scenes, but discard the information of temporal motions. For example, in the example of Fig. 7.1, information about "panda" could be included in visual appearance features, but information about "sliding" will more likely be included in motion features. To capture sufficient spatial-temporal features, our framework employs two strategies: 1) two channels of raw video representations are included: motion history images and RGB video frames. MHI focus on temporal motions and RGB frames focus on spatial appearances. 2) For each short clips in each channel (16 frames), temporal-spatial features are computed via a 3D convolutional neural network (C3D [143]). The C3D networks are pre-trained on action recognition dataset so that they are capable to capture discriminative spatial-temporal features.

Context embedded video representation. Before feeding the extracted C3D features into video encoding RNNs, an additional pooling layer is added to provide more context information to the video representation:

$$\begin{aligned} \phi(V) &= \{v_0, v_1, ..., v_T\} \\ v_t &= \begin{cases} max\_pool(v_1, ..., v_T), & \text{if } t = 0 \\ C3D(c_t), & \text{otherwise}, \end{cases} \end{aligned} \tag{7.8}$$

where v_t represents the input for video encoding RNN at each time step t and c_t represents the corresponding video clip.

Therefore, at time step $t = 0$, the encoding RNN is fed with the "context" vector, which is the max pooling vector over all C3D feature vectors. In this way, the video encoding RNN starts with the holistic knowledge about the whole video before taking the sequential inputs representing each video clip.

7.2 Datasets

7.2.1 Microsoft video description corpus

The Microsoft video description (MSVD) corpus is a video snippet-based dataset, which focuses on describing simple interactive events, such as driving, cooking, *etc.* Each video snippet is collected from YouTube. There are about $1,658$ video clips in this corpus which are available by the time of our experiments. Each video snippet lasts from multiple seconds to several minutes. Human annotators were asked to describe the video snippet using one sentence from any language. Since each video snippet was assigned to multiple annotators, there are multiple sentences for one video snippet. Here, our chapter only focuses on English descriptions. Among the $1,658$ video snippets, 300 are used as testing and the rest are for training.

7.2.2 Movie Description Datasets

In this chapter, two movie description datasets are employed: Max Planck Institute for Informatics Movie Description Dataset (MPII) and Montreal video annotation dataset (MVAD). Both of the MPII dataset [118] and MVAD dataset [141] are collected from Hollywood movies. MPII dataset contains over $68,000$ video snippets from 94 High-definition movies and MVAD dataset contains $49,000$ video snippets from 92 movies. The text annotation from the MVAD dataset is from Descriptive Video Service (DVS), a linguistic description that allows visually impaired people to follow the movie. Besides DVS, the MPII dataset also employs movie scripts to enrich the text annotations. Both datasets are very challenging compared to the MSVD dataset in several aspects: 1) movie videos have more complex scenes and varied backgrounds. 2) The text annotations are sourced from a combined corpus, therefore the linguistic complexity is much higher than well-structured sentences as in the MSVD dataset. The MVAD and the MPII datasets belong to the recent Large Scale Movie Description Challenge (LSMDC). We report evaluation on the public testing set, where the MPII dataset has $3,535$ testing video/sentence pairs and the MVAD has $6,518$.

7.2.3 American Sign Language video description corpus

To the best of our knowledge, previous automatic ASL recognition frameworks only focus on hand gesture or facial expression recognition. We further explore the utilization of video captioning framework for ASL recognition. Since there is no proper public dataset for this task, we propose a new dataset, **ASL-TEXT**, collected from YouTube. This proposed dataset is focused on describing videos of ASL signing, and it contains about $20,000$ video-sentence pairs. The ASL-TEXT dataset is very challenging in two aspects: 1) the scenes are complex but irrelevant, and the only relevant information is from human facial expressions and body gestures. 2) The sentences are extracted from YouTube subtitles, some of which are generated by automatic voice recognition. Therefore the language complexity and variation are even higher than the previous mentioned movie description datasets.

The resource of ASL on YouTube comes in several categories, such as *ASL lessons, ASL songs,* and *ASL instructions provided by public institutes*. We manually search on YouTube with multiple textual queries such as "ASL", "American Sign Language", and "ASL Lessons", *etc*. The search results are further manually filtered using several criteria: 1) the search results should be correct ASL signing. 2) The subtitles associated with the video snippets should be available. 3) There should be only one frontal-view signer in the video. To further rule out unnecessary background noises, face detection is applied on each video frame and the video frames are then centered and cropped according to the face detection results. Some examples of the dataset are shown in Fig. 7.3 (d).

Table 7.1: Comparative statistics of the propose ASL-TEXT dataset with the MSVD and MPII datasets.

	#-sentences	#-words	vocab. size	avg. length
MPII	68,375	679,157	21,700	3.9s
MVAD	56,634	568,408	18,092	6.2s
ASL-TEXT	22,527	178,637	11,193	5.4s

Following the convention in MPII and MVAD datasets, each video is segmented into several short snippets. Since each video in our dataset has caption (or subtitle) available, we segment the videos so that each video clip corresponds to one sentence in the caption text. As a result, the ASL-TEXT dataset contains $22,527$ video/sentence pairs and the average length of video clips is $5.4s$. The sizes of vocabularies in the three datasets are comparable but the ASL-TEXT dataset has less words. The ASL-TEXT dataset is more challenging because the averaged word frequency is much lower than in the other two datasets.

7.3 Experimental Results

7.3.1 Experimental setup

Metric. In this chapter, we mainly evaluate the proposed framework using the METEOR evaluation metric [24]. Compared to other n-gram-based metrics such as BLEU [105], METEOR is more appropriate to evaluate sequential predictions. METEOR scores the predictions by aligning them to more than one reference sentences, which are based on exact, stem, synonym, and paraphrase matches between words and phrases. Therefore METEOR takes more linguistic and semantic information into consideration.
Loss function. In each iteration during the training process, a batch of images is fed into the neural networks, and the language decoder generates a sequence of probability distributions. A log-likelihood function is applied for each probability vector and corresponding ground-truth vector (1-hot-vector). The losses and gradients are then computed by maximizing the likelihood function. The losses and gradients are averaged and back-propagated to the preceding network modules for parameter updates.
Training and optimization. For computational efficiency, we assign the weights for the C3D networks with a pre-trained network and do not apply fine-tuning. The rest of the modules (LSTM feature encoder, fusion layer, and LSTM language decoder) are trained end-to-end using stochastic gradient descent. The learning rates for all modules are set to 0.0001. Each iteration contains a batch of 16 samples. All RNN sizes are set to 1024. The drop-out rates for both encoder and decoder are set to 0.5. We implement the networks using Torch7 [18] and CuDNN. It takes about 1 to 3 days to converge on the training set using a GeForce TitanX core, depending on the sizes of datasets.

7.3.2 Video Description Results

MSVD dataset. The comparative METEOR scores of the proposed and other methods are shown in Table 7.2. The proposed method significantly outperforms the baseline factor graph model (FGM [138]) by 6.3%. Comparing with *mean-pooling* methods [149], the improvements are $1.1\%{\sim}3.3\%$, which demonstrate that including more temporal dynamic information is beneficial. Comparing with the current sequential modeling state-of-the-arts, temporal attention (TA) [172] and S2VT [148], our proposed method performs slightly better (30.2% *vs.* $29.0{\sim}29.8\%$). Some qualitative results are shown in Fig. 7.3 (a).

MSVD dataset is more focused on describing static human-object interactions and scenes, such as "someone is doing something in somewhere". Comparing temporal-based methods (the proposed, TA [172] and S2VT [148]) and static-based methods (mean-pooling [149]), there are improvements but limited.

MPII and MVAD datasets. To further comparative evaluate our proposed method with the state-of-the-arts on more temporal-focused datasets, two movie-based datasets (MVAD and MPII) are employed for comparison. The proposed framework and other state-of-the-arts are compared in Table 7.3. Despite the scores on each of the MPII and MVAD datasets, we also report the overall scores (weighed by the sizes of testing set). Our result

Table 7.2: METEOR scores on the MSVD dataset.

Method	METEOR (%)
FGM [138]	23.9
AlexNet [149]	26.9
VGG [149]	27.7
AlexNet-COCO [149]	29.1
GoogleNet [172]	28.7
GoogleNet + TA [172]	29.0
GoogleNet + 3D-CNN + TA [172]	29.6
AlexNet(Flow) + S2VT [148]	24.3
AlexNet + S2VT [148]	27.9
VGG + S2VT [148]	29.2
VGG + AlexNet(Flow) + S2VT [148]	29.8
Proposed	**30.2**

(7.06) outperforms Visual-Labels (6.55) and VGG (6.31) by 0.51 and 0.75, respectively. It is beneficial to explicitly model the temporal dynamics of the input videos.

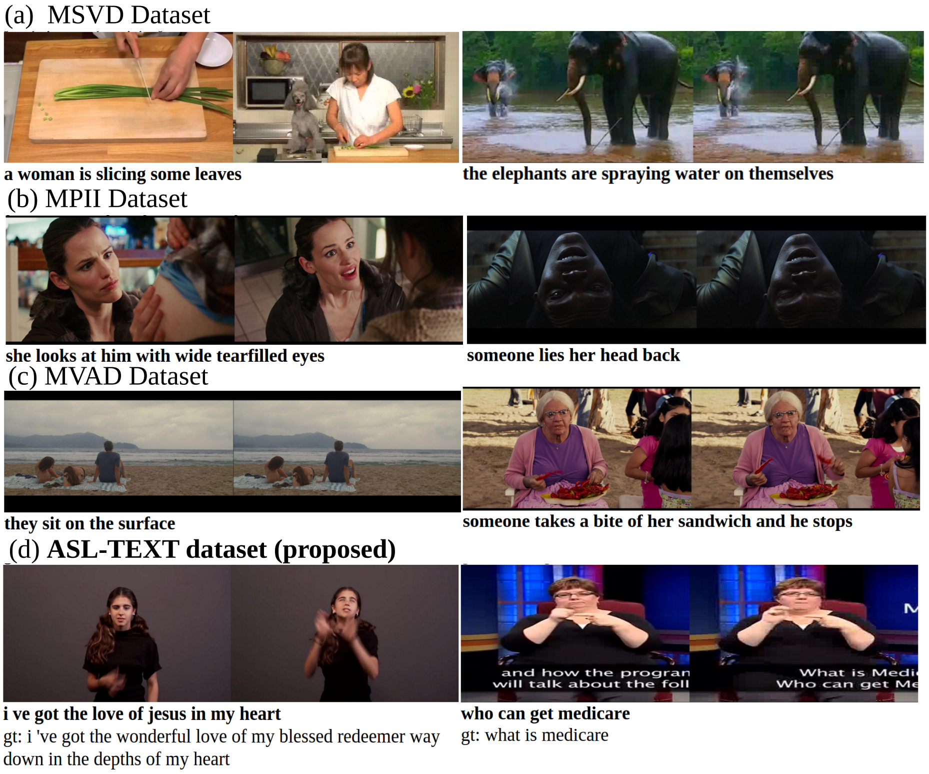
Compared to the previous state-of-the-art sequence-to-sequence model (S2VT [148]), our framework outperforms by 0.25. The experimental results demonstrate that our framework can avoid feature entanglement so that it can better model the temporal structures of videos.

Table 7.3: METEOR scores (%) on the Movie Description datasets, higher is better.

Method	MPII [118]	MVAD [141]	Overall
SMT [118]	5.6	–	–
Visual-Labels [117]	7.0	6.3	6.55
VGG [149]	6.7	6.1	6.31
Temporal Attention [172]	–	4.3	–
S2VT [148]	7.1	6.7	6.81
Proposed	7.0	7.1	**7.06**

7.3.3 ASL-TEXT

Since there is no other result available on our ASL-TEXT dataset, we evaluate the proposed framework on this new dataset comparing among different network configurations. There are two aspects to be investigated in this comparative evaluation. Firstly, since our fusion layer can assign different dimensions to each feature channel, the impact of assigning different portions to RGB and MHI is discussed. Secondly, the impact of RNN sizes for both feature encoders and language decoders is discussed. $20,527$ training samples and $2,000$ testing samples from ASL-TEXT are used and the METEOR scores of different configurations are shown in Table 7.4. In Table 7.4, $(RGB)\%$ denotes the parameter of how much percent of the encoding feature dimensions is assigned to RGB channel; (RNN_{ENC}, RNN_{DEC}) denotes the RNN sizes for encoder and decoder. There are two observations can be made from Table 7.4: 1) for each row, the METEOR score increases as the RNN sizes increases but after an optimal size setting, the performance



(a) MSVD Dataset

a woman is slicing some leaves

the elephants are spraying water on themselves

(b) MPII Dataset

she looks at him with wide tearfilled eyes

someone lies her head back

(c) MVAD Dataset

they sit on the surface

someone takes a bite of her sandwich and he stops

(d) **ASL-TEXT dataset (proposed)**

i ve got the love of jesus in my heart
gt: i 've got the wonderful love of my blessed redeemer way down in the depths of my heart

who can get medicare
gt: what is medicare

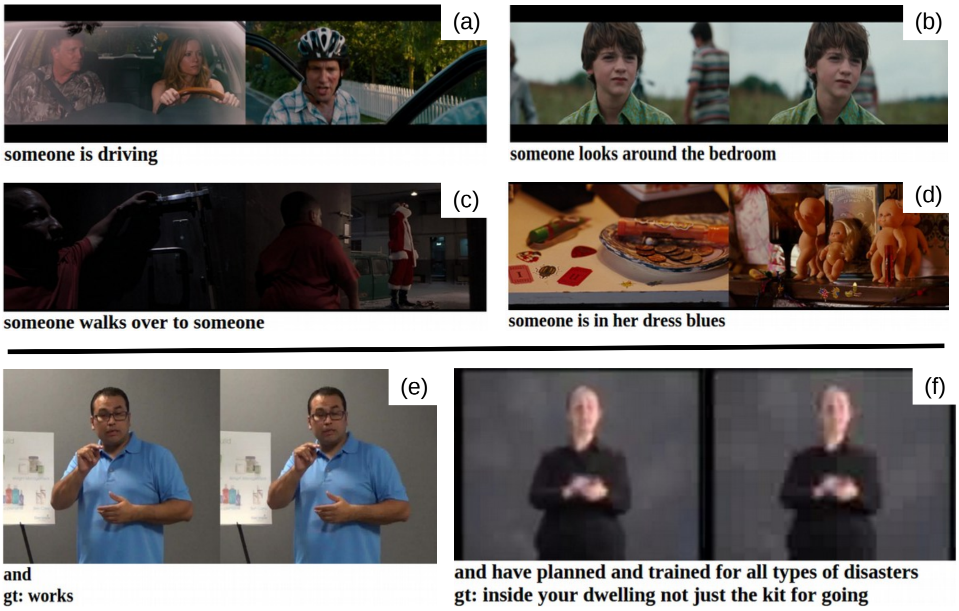
Figure 7.3: Qualitative results of the proposed video captioning framework on four datasets: (a) MSVD, (b) MPII, (c) MVAD and (d) ASL-TEXT. The bold sentence under each pair of images is the predicted caption and for ASL-TEXT the ground-truth text is also attached.

starts to decrease. 2) Assigning different dimensions to different feature channels has little impact on the performance. Observation 1 shows that the ASL-TEXT dataset is more complex than other datasets because even moderate RNN sizes such as $(512, 512)$ is sufficient to over-fitting. Observation 2 demonstrates that our framework can automatically learn an optimal combination of multiple feature channels. Therefore there is no need to manually tune the weight of different feature channels.

Table 7.4: METEOR scores on the ASL-TEXT dataset of different configurations.

		(RNN_{ENC}, RNN_{DEC})				
		(128,128)	(256,128)	(256,256)	(512,256)	(512,512)
(RGB)%	10%	3.9	4.7	4.3	4.2	3.6
	30%	4.1	3.8	4.7	3.5	3.9
	50%	3.7	4.7	3.5	3.9	3.9
	90%	3.7	3.7	3.5	4.5	4.0

Some qualitative results of the proposed framework have been shown in Fig. 7.3. For simple scenes and interactive actions in Fig. 7.3 (a), our system can accurately generate descriptive sentences. For more complex scenarios as in movies (Fig. 7.3 (b) and (c)), our system can predict well on the main actions (such as “sit”, “eat” and “enter”) but make errors in objects. For ASL recognition, it is promising to observe that the system has the potential to build relationships between key words (such as "love", "medicare", "WH-sign" and "single/married") and videos. The results demonstrate that exploring temporal structures and combining multiple feature channels are potentially beneficial for video captioning even in complex visual content and sentence structures.



someone is driving

someone looks around the bedroom

someone walks over to someone

someone is in her dress blues

and
gt: works

and have planned and trained for all types of disasters
gt: inside your dwelling not just the kit for going

Figure 7.4: Some failed examples. Upper panel (a-d) is from the movie description datasets. Lower panel (e,f) is from the ASL-TEXT dataset. We can observe that complicated static scenes, too-blurry frames, and too-short clips can lead to bad results.

7.4 Discusssion

In this chapter, we mainly focus on solving the problem of how to automatically integrate multi-modality input with LSTM-based video description framework.

However, the current framework often results in over-simplified sentences when the input video is too complicated. Some of the failure cases have been shown in Figure 7.4. As shown in Figure 7.4 (a), the output sentence is "someone is driving", which only captures the most trivial information. For video clips with almost-static scenes (b and e), the output sentences are also not accurate due to lack of motion information. As for too-complex scenes such as shown in (d), the algorithm also fails to generate useful descriptions because the objects contained in the video are too small and uncommon. Too-blurry frames as shown in (f) will also lead to inaccurate results.

To handle these difficulties, we will concentrate on combining dense image captioning framework with video captioning in the future work. The purpose of introducing dense captioning is to capture more detailed object and scene information. We believe that with more detailed information captured by dense captioning, the resulted sentences will be more diverse and descriptive.

Chapter 8

Discussions and Future Directions

This dissertation concentrates on human activity analysis. More specifically, we start by describing two kinds of depth-based descriptor, one is a local depth image-based descriptor and the other is a holistic depth video-based descriptor. The two descriptors shows their effectiveness in human action and hand gesture recognition. Then we investigated how to combine both RGB and depth, facial expressions and body gestures for combinational human affection recognition. A two-layer sparse representation-based classification scheme is proposed to eliminate subjective variances. To bridge the gap between raw visual features and abstract action labels, this dissertation then discussed a framework for human action attribute learning based on deep activations. Finally, a multi-stream sequential encoder-decoder language modeling framework is discussed for automatic generating sentences for human action and related scenarios.

Depth map-based image descriptor for human action and hand gesture recognition. We firstly proposed a novel discriminative 3D descriptor (H3DF) which can effectively capture and model the rich surface shape information of the depth maps. Applying orientation normalization, robust coding and concentric spatial pooling, our H3DF descriptor is robust to translation, view angle and scaling changes. Local H3DF is also able to evolve into denseH3DF for modeling more local patterns. To tackle the task of dynamic hand gesture and human action recognition from depth video sequences, two temporal extension approaches are developed: Dynamic Programming-based temporal partition and N-gram-based method. The two approaches are applied to build augmented descriptors with robust representative description. We have extensively evaluated the effectiveness of the proposed H3DF descriptor on four public datasets including static hand gesture recognition from single depth image, dynamic hand gesture and human action recognition from depth sequences. The experimental results demonstrate that our proposed approach outperforms or achieves comparable accuracy to the state-of-the-art for action and hand gesture recognition.

Holistic depth video descriptor for human action recognition. An edge enhanced depth motion map framework is proposed to model different hand gestures from their visual effects. Then we have designed a new dynamic temporal pyramid organization approach to capture temporal structure to compensate the information loss due to building energy map by integrating discrete energy from each frame along temporal dimension. For classification, we apply a Support Vector Machine with linear kernel. Experiments demonstrate that our method achieves better performance compared to the state-of-the-art methods while using relative simple classifier rather than involving complicate dictionary learning techniques. In addition, our proposed method is more general among different hand gesture categories.

Multi-modality human affection recognition. Affection recognition from the perspective of facial expression and body gesture combination in RGB-D videos is discussed. To address the issue that subjective variance in affection recognition is always larger than inter-class variance, we have proposed a novel subject adaptive algorithm to mining category-related variance by using sparse representation with Fisher discriminant. Instead of using all training data for each testing query, we firstly select a subject adaptive subset using sparse representation based classification. Then affection class is recognized in the selected subject adaptive subset of training data. To jointly recognize affection class from facial expressions and body gestures, we propose a confident reconstruction based joint decision making strategy. We also presented a novel dataset which contains 10 different affection categories and 12 subjects, which is challenging due to large subjective variance. Our proposed recognition framework and joint recognition approach is evaluated on the dataset. Experimental results demonstrate that joint recognition results can be improved by combing two complementary discriminative models.

Deep learning and human action attribute recognition. A novel joint action attribute learning algorithm for depth videos which is based on multistream deep neural networks is proposed. To model complex semantics in action attributes, an undirected graph is integrated in the formulation of attribute learning. Extensive experiments demonstrate that the proposed method is effective in learning action attributes for depth videos. Experiment results based on our method outperform existing state-of-the-art methods in action attribute detection, zero-shot action recognition and conventional action recognition.

Multi-stream sequence-to-sequence video description based on RNN. In this chapter, we have proposed a novel video captioning framework based on a two-stage encoder-decoder system. The encoding part is composed of a multi-channel LSTM-based RNNs which can capture the temporal dynamics in video clips by allowing arbitrary-length input sequences. The decoding part is a LSTM-based language model which can decode the input video feature vector to a sequence of English words. A fusion layer is inserted between the encoder and decoder to automatically learn the optimized combination of multiple channels. To capture spatial-temporal information in the videos, we apply 3D convolutional neural networks pretrained for action recognition (C3D) to extract features from both MHIs and raw RGB video frames. The whole network can be trained end-to-end using back-propagation. The proposed model is extensively evaluated on three public video description datasets comparing with the state-of-the-art methods and outperforms their performances. Furthermore, we collect an ASL recognition dataset and propose to apply video description framework in the area of automatic ASL recognition.

Future directions Our future work will be continuing research on human activity analysis. We believe that it is promising to focus on exploring and exploit the utilization of high-performance deep learning techniques in continuous human action segmentation, recognition and description. The future of our research directions will cover the following three topics:

- American Sign Language related recognition and human computer interaction research.
- Modeling the temporal saliency of human activities using deep learning and other techniques.
- Exploring the exploiting the utilizations of other modalities, such as audio and text.

Appendix A

Publications During Ph.D. Study

1. C. Zhang and Yingli Tian, Automatic Video Captioning via Multichannel Sequential Encoding, European Conference on Computer Vision (ECCV), 2016 (Submitted).
2. C. Zhang and Yingli Tian, Automatic Video Description Generation via LSTM with Joint Two-stream Encoding, ICPR, 2016 (Submitted).
3. C. Zhang and Yingli Tian, Multi-modality American Sign Language Recognition, IEEE International Conference on Image Processing (ICIP), 2016 (Submitted).
4. Z. Liu, C. Zhang, Y. Tian, 3D-based Deep Convolutional Neural Network for Action Recognition with Depth Sequences, Elsevier Editorial System for Image and Vision Computing, 2016.
5. C. Zhang, Y. Tian, X. Guo, J. Liu, DAAL: Deep Activation-based Attribute Learning for Action Recognition in Depth Videos, International Journal of Computer Vision (IJCV), 2016 (Submitted).
6. C. Zhang and Yingli Tian, BCA: Bi-symmetric Component Analysis for Temporal Symmetry in Human Actions, IEEE International Conference on Multimedia and Expo (ICME), 2016.
7. C. Zhang and Yingli Tian, Histogram of 3D Facets: A Depth Descriptor for Human Action and Hand Gesture Recognition, Computer Vision and Image Understanding, Vol 139, pp. 29-39, Oct, 2015.
8. C. Zhang and Yingli Tian, Subject Adaptive Affection Recognition via Sparse Reconstruction, IEEE Workshop on Vision Meets Cognition, in conjunction with CVPR, 2014.
9. S. Wang, H. Pan, C. Zhang, and Y. Tian. RGB-D Image-Based Detection of Stairs, Pedestrian Crosswalks and Traffic Signs, Journal of Visual Commu nication and Image Representation (JVCIR), 2013
10. C. Zhang, X. Yang, C. Yi, Y. Tian, Q. Yu, A. Tamrakar, and A. Divakaran. CCNY-SRI @ TRECVid 2013 intED: a Human Interactive Event Detection System, NIST TRECVID Workshop, 2013
11. C. Zhang, and Y. Tian. Edge Enhanced Depth Motion Map for Dynamic Hand Gesture Recognition, CVPR 2013 International Workshop on Human Activity Understanding from 3D Data (HAU3D)

12. C. Zhang, X. Yang and Y. Tian. Histogram of 3D Facet: A Characteristic Descriptor for Hand Gesture Recognition, IEEE International Conference on Automatic Face and Gesture Recognition (FG 2013), 2013.

13. C. Zhang, and Y. Tian. RGBD Camera-based Activity Analysis (Invited Paper, Oral), Asia Pacific Signal and Information Processing Association (APSIPA) Annual Summit and Conference, CA, 2012.

14. C. Zhang, and Y. Tian. RGB-D Camera-based Daily Living Activity Recognition. Journal of Computer Vision and Image Processing, Vol. 2, No. 4, December 2012.

15. C. Zhang, M. Shabbir, Y. Tian, and D. Stylianou. Computer Vision-based Mathematics Learning Enhancement for Children with Visual Impairments. International Workshop on Biomedical and Health Informatics (BHI), 2012.

16. X. Yang, C. Zhang, and Y. Tian. Recognizing Actions Using Depth Motion Maps-based Histograms of Oriented Gradients. International Conference on ACM Multimedia, 2012

17. C. Zhang, Y. Tian, and E. Capezuti. Privacy Preserving Automatic Fall Detection for Elderly Using RGBD Cameras. International Conference on Computers Helping People with Special Needs (ICCHP), 2012.

Bibliography

[1] Hrishikesh Aradhye, George Toderici, and Jay Yagnik. “Video2Text: Learning to Annotate Video Content”. In: *ICDM Workshop on Internet Multimedia Mining*. 2009.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[3] Marian Stewart Bartlett. “Face image analysis by unsupervised learning and redundancy reduction”. In: (1998).

[4] Sumit Basu, Nuria Oliver, and Alex Pentland. “3D modeling and tracking of human lip motions”. In: *Computer Vision, 1998. Sixth International Conference on*. IEEE. 1998, pp. 337–343.

[5] Richard Bellman. “On the approximation of curves by line segments using dynamic programming”. In: *Communications of the ACM* 4.6 (1961), p. 284.

[6] Serge Belongie, Jitendra Malik, and Jan Puzicha. “Shape matching and object recognition using shape contexts”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24.4 (2002), pp. 509–522.

[7] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *Neural Networks, IEEE Transactions on* 5.2 (1994), pp. 157–166.

[8] Michael Van den Bergh and Luc Van Gool. “Combining RGB and ToF cameras for real-time 3D hand gesture interaction”. In: *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE. 2011, pp. 66–72.

[9] G. Bobick and J. Davis. “The representation and recognition of human movement using temporal templates”. In: *IEEE Trans. on PAMI* (2001).

[10] Stephen Boyd et al. “Distributed optimization and statistical learning via the alternating direction method of multipliers”. In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[11] Peter F Brown et al. “Class-based n-gram models of natural language”. In: *Computational linguistics* 18.4 (1992), pp. 467–479.

[12] Chao-Yeh Chen and Kristen Grauman. “Efficient activity detection with max-subgraph search”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1274–1281.

[13] Hui Chen et al. “3D model-based continuous emotion recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1836–1845.

[14] Xinlei Chen and C Lawrence Zitnick. "Learning a recurrent visual representation for image caption generation". In: *arXiv preprint arXiv:1411.5654* (2014).

[15] Xinlei Chen et al. "Microsoft COCO captions: Data collection and evaluation server". In: *arXiv preprint arXiv:1504.00325* (2015).

[16] Sien W Chew et al. "Improved facial expression recognition via uni-hyperplane classification". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 2554–2561.

[17] Wongun Choi, Khuram Shahid, and Silvio Savarese. "Learning context for collective activity recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3273–3280.

[18] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. "Torch7: A matlab-like environment for machine learning". In: *BigLearn, NIPS Workshop*. EPFL-CONF-192376. 2011.

[19] Microsoft Corporation. *Kinect for Xbox 360*. `http://www.xbox.com/en-US/kinect`. 2010.

[20] Xinyi Cui et al. "Abnormal detection using interaction energy potentials". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3161–3167.

[21] Matthew N Dailey and Garrison W Cottrell. "PCA: Gabor for Expression Recognition". In: (1999).

[22] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 886–893.

[23] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. IEEE. 2009, pp. 248–255.

[24] Michael Denkowski and Alon Lavie. "Meteor universal: Language specific translation evaluation for any target language". In: *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*. Vol. 6. 2014.

[25] P. Dollar. *Piotr's Image and Video Matlab Toolbox (PMT)*. `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`.

[26] Fabio Dominio, Mauro Donadeo, and Pietro Zanuttigh. "Combining multiple depth-based descriptors for hand gesture recognition". In: *Pattern Recognition Letters* 50 (2014), pp. 101–111.

[27] Jeff Donahue et al. "Decaf: A deep convolutional activation feature for generic visual recognition". In: *arXiv preprint arXiv:1310.1531* (2013).

[28] Jeff Donahue et al. "Long-term recurrent convolutional networks for visual recognition and description". In: *arXiv preprint arXiv:1411.4389* (2014).

[29] Philippe Dreuw et al. "Speech recognition techniques for a sign language recognition system". In: *hand* 60 (2007), p. 80.

[30] Bradley Efron et al. "Least angle regression". In: *The Annals of statistics* 32.2 (2004), pp. 407–499.

[31] Alexei A Efros et al. “Recognizing action at a distance”. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE. 2003, pp. 726–733.

[32] A Evgeniou and Massimiliano Pontil. “Multi-task feature learning”. In: *NIPS* 19 (2007), p. 41.

[33] Clement Farabet et al. “Learning hierarchical features for scene labeling”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013), pp. 1915–1929.

[34] Ali Farhadi et al. “Describing objects by their attributes”. In: *CVPR*. IEEE. 2009, pp. 1778–1785.

[35] Alireza Fathi, Jessica K Hodgins, and James M Rehg. “Social interactions: A first-person perspective”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1226–1233.

[36] Ronald A Fisher. “The use of multiple measurements in taxonomic problems”. In: *Annals of eugenics* 7.2 (1936), pp. 179–188.

[37] Kai-Yin Fok et al. “A Real-Time ASL Recognition System Using Leap Motion Sensors”. In: *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on*. IEEE. 2015, pp. 411–414.

[38] Alexandre RJ François and Gérard G Medioni. “Adaptive color background modeling for real-time segmentation of video streams”. In: *Proceedings of the International Conference on Imaging Science, Systems, and Technology*. Vol. 1. 121. 1999, pp. 227–232.

[39] E Friesen and P Ekman. “Facial action coding system: A technique for the measurement of facial movement”. In: *Palo Alto* (1978).

[40] Yanwei Fu et al. “Learning multimodal latent attributes”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36.2 (2014), pp. 303–316.

[41] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[42] Sergio Guadarrama et al. “Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition”. In: *ICCV*. IEEE. 2013, pp. 2712–2719.

[43] Hatice Gunes and Massimo Piccardi. “A bimodal face and body gesture database for automatic analysis of human nonverbal affective behavior”. In: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*. Vol. 1. IEEE. 2006, pp. 1148–1153.

[44] Simon Hadfield and Richard Bowden. “Hollywood 3d: Recognizing actions in 3d natural scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 3398–3405.

[45] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. “Canonical correlation analysis: An overview with application to learning methods”. In: *Neural computation* 16.12 (2004), pp. 2639–2664.

[46] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[47] Geoffrey E Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580* (2012).

[48] Minh Hoai and Andrew Zisserman. "Talking heads: Detecting humans and recognizing their interactions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 875–882.

[49] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[50] Chung-Lin Huang and Yu-Ming Huang. "Facial expression recognition using model-based feature extraction and action parameters classification". In: *Journal of Visual Communication and Image Representation* 8.3 (1997), pp. 278–290.

[51] Haiqi Huang et al. "A multi-modal clustering method for web videos". In: *Trustworthy Computing and Services*. Springer, 2013, pp. 163–169.

[52] Matt Huenerfauth et al. "Comparing Methods of Displaying Language Feedback for Student Videos of American Sign Language". In: *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility*. ACM. 2015, pp. 139–146.

[53] Nazlı Ikizler and Pınar Duygulu. "Human action recognition using distribution of oriented rectangular patches". In: *Human Motion–Understanding, Modeling, Capture and Animation*. Springer, 2007, pp. 271–284.

[54] Dinesh Jayaraman and Kristen Grauman. "Zero-shot recognition with unreliable attributes". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3464–3472.

[55] Dinesh Jayaraman, Fei Sha, and Kristen Grauman. "Decorrelating Semantic Visual Attributes by Resisting the Urge to Share". In: *CVPR*. IEEE. 2014.

[56] Hueihan Jhuang et al. "A biologically inspired system for action recognition". In: *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. Ieee. 2007, pp. 1–8.

[57] Shuiwang Ji et al. "3D convolutional neural networks for human action recognition". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.1 (2013), pp. 221–231.

[58] Michael A Karchmer et al. "How many people use ASL in the United States? Why estimates need updating". In: *Sign Language Studies* 6.3 (2006), pp. 306–335.

[59] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *arXiv preprint arXiv:1412.2306* (2014).

[60] Andrej Karpathy et al. "Large-scale video classification with convolutional neural networks". In: *CVPR*. 2014.

[61] Cem Keskin et al. "Hand pose estimation and hand shape classification using multi-layered randomized decision forests". In: *Computer Vision–ECCV 2012*. Springer, 2012, pp. 852–863.

[62] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. "A spatio-temporal descriptor based on 3d-gradients". In: *BMVC 2008-19th British Machine Vision Conference*. British Machine Vision Association. 2008, pp. 275–1.

[63] Hiroshi Kobayashi and Fumio Hara. "Facial interaction between animated 3D face robot and human beings". In: *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*. Vol. 4. IEEE. 1997, pp. 3732–3737.

[64] Yu Kong and Yun Fu. "Modeling supporting regions for close human interaction recognition". In: *Computer Vision-ECCV 2014 Workshops*. Springer. 2014, pp. 29–44.

[65] Niveda Krishnamoorthy et al. "Generating natural-language video descriptions using text-mined knowledge". In: *NAACL HLT 2013* (2013), p. 10.

[66] Parag Kulkarni et al. "Transfer learning via attributes for improved on-the-fly classification". In: *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE. 2014, pp. 220–226.

[67] Alexey Kurakin, Zhengyou Zhang, and Zicheng Liu. "A real time system for dynamic hand gesture recognition with a depth sensor". In: *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE. 2012, pp. 1975–1979.

[68] John Lafferty, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In: (2001).

[69] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. "Learning to detect unseen object classes by between-class attribute transfer". In: *CVPR*. IEEE. 2009, pp. 951–958.

[70] Tian Lan, Leonid Sigal, and Greg Mori. "Social roles in hierarchical models for human activity recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1354–1361.

[71] Ivan Laptev. "On space-time interest points". In: *International Journal of Computer Vision* 64.2-3 (2005), pp. 107–123.

[72] Ivan Laptev et al. "Learning realistic human actions from movies". In: *CVPR*. IEEE. 2008, pp. 1–8.

[73] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *CVPR*. Vol. 2. IEEE. 2006, pp. 2169–2178.

[74] Quoc V Le et al. "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis". In: *CVPR*. IEEE. 2011, pp. 3361–3368.

[75] Yann LeCun and Yoshua Bengio. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361 (1995).

[76] Honglak Lee et al. "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 609–616.

[77] Binlong Li et al. "Activity recognition using dynamic subspace angles". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3193–3200.

[78] Liangyue Li, Sheng Li, and Yun Fu. "Discriminative dictionary learning with low-rank regularization for face recognition". In: *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE. 2013, pp. 1–6.

[79] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. "Action recognition based on a bag of 3d points". In: *CVPR Workshops*. IEEE. 2010, pp. 9–14.

[80] Weixin Li et al. "Recognizing activities via bag of words for attribute dynamics". In: *CVPR*. IEEE. 2013, pp. 2587–2594.

[81] Hui Liang, Junsong Yuan, and Daniel Thalmann. "Parsing the hand in depth images". In: *Multimedia, IEEE Transactions on* 16.5 (2014), pp. 1241–1253.

[82] Guangcan Liu, Zhouchen Lin, and Yong Yu. "Robust subspace segmentation by low-rank representation". In: *ICML*. 2010, pp. 663–670.

[83] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. "Recognizing human actions by attributes". In: *CVPR*. IEEE. 2011, pp. 3337–3344.

[84] Jingen Liu, Jiebo Luo, and Mubarak Shah. "Recognizing realistic actions from videos "in the wild"". In: *CVPR*. IEEE. 2009, pp. 1996–2003.

[85] Jingen Liu and Mubarak Shah. "Learning human actions via information maximization". In: *CVPR*. IEEE. 2008, pp. 1–8.

[86] Jingjing Liu et al. "Recognizing eyebrow and periodic head gestures using CRFs for non-manual grammatical marker detection in ASL". In: *FGR*. IEEE. 2013, pp. 1–6.

[87] Ningning Liu et al. "Associating textual features with visual ones to improve affective image classification". In: *Affective Computing and Intelligent Interaction*. Springer, 2011, pp. 195–204.

[88] Wei-Lwun Lu et al. "Identifying players in broadcast sports videos using conditional random fields". In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE. 2011, pp. 3249–3256.

[89] Bruce D Lucas, Takeo Kanade, et al. "An iterative image registration technique with an application to stereo vision." In: *IJCAI*. Vol. 81. 1981, pp. 674–679.

[90] Junhua Mao et al. "Explain images with multimodal recurrent neural networks". In: *arXiv preprint arXiv:1410.1090* (2014).

[91] Michael Marszalek, Ivan Laptev, and Cordelia Schmid. "Actions in context". In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 2929–2936.

[92] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. "Trajectons: Action recognition through the motion analysis of tracked features". In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE. 2009, pp. 514–521.

[93] Calvin R Maurer Jr, Rensheng Qi, and Vijay Raghavan. "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.2 (2003), pp. 265–270.

[94] Angeliki Metallinou, Sungbok Lee, and Shrikanth Narayanan. "Audio-visual emotion recognition using gaussian mixture models for face and voice". In: *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*. IEEE. 2008, pp. 250–257.

[95] Angeliki Metallinou and Shrikanth Narayanan. "Annotation and processing of continuous emotional attributes: Challenges and opportunities". In: *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*. IEEE. 2013, pp. 1–8.

[96] Dimitris N Metaxas et al. "Recognition of Nonmanual Markers in American Sign Language (ASL) Using Non-Parametric Adaptive 2D-3D Face Tracking." In: *LREC*. Citeseer. 2012, pp. 2414–2420.

[97] Stephen Milborrow and Fred Nicolls. "Locating facial features with an extended active shape model". In: *Computer Vision–ECCV 2008*. Springer, 2008, pp. 504–513.

[98] Nicholas Morsillo, Gideon Mann, and Christopher Pal. "Youtube scale, large vocabulary video annotation". In: *Video Search and Mining*. Springer, 2010, pp. 357–386.

[99] André Mourão et al. "Facial expression recognition by sparse reconstruction with robust features". In: *Image Analysis and Recognition*. Springer, 2013, pp. 107–115.

[100] Qutaishat Munib et al. "American sign language (ASL) recognition based on Hough transform and neural networks". In: *Expert systems with Applications* 32.1 (2007), pp. 24–37.

[101] Shay Ohayon, Winrich A Freiwald, and Doris Y Tsao. "What makes a cell face selective? The importance of contrast". In: *Neuron* 74.3 (2012), pp. 567–581.

[102] Eshed Ohn-Bar and Mohan Trivedi. "Joint angles similarities and HOG2 for action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2013, pp. 465–470.

[103] Omar Oreifej and Zicheng Liu. "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences". In: *CVPR*. IEEE. 2013, pp. 716–723.

[104] Maja Pantic and Leon JM Rothkrantz. "Toward an affect-sensitive multimodal human-computer interaction". In: *Proceedings of the IEEE* 91.9 (2003), pp. 1370–1390.

[105] Kishore Papineni et al. “BLEU: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.

[106] Glauco Vitor Pedrosa and Agma JM Traina. “From bag-of-visual-words to bag-of-visual-phrases using n-grams”. In: *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI-Conference on*. IEEE. 2013, pp. 304–311.

[107] Yigang Peng et al. “RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34.11 (2012), pp. 2233–2246.

[108] Hanspeter Pfister et al. “Surfels: Surface elements as rendering primitives”. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 2000, pp. 335–342.

[109] Rosalind W Picard and Roalind Picard. *Affective computing*. Vol. 252. MIT press Cambridge, 1997.

[110] Nicolas Pugeault and Richard Bowden. “Spelling it out: Real-time ASL fingerspelling recognition”. In: (2011), pp. 1114–1119.

[111] Kishore K Reddy, Jingen Liu, and Mubarak Shah. “Incremental action recognition using feature-tree”. In: *ICCV*. IEEE. 2009, pp. 1010–1017.

[112] Zhou Ren, Junsong Yuan, and Zhengyou Zhang. “Robust hand gesture recognition based on finger-earth mover’s distance with a commodity depth camera”. In: *Proceedings of the 19th ACM international conference on Multimedia*. ACM. 2011, pp. 1093–1096.

[113] Zhou Ren et al. “Minimum near-convex decomposition for robust shape representation”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 303–310.

[114] Microsoft Research. *MSR Action Recognition Datasets*. `http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/default.htm`.

[115] Microsoft Research. *MSR Gesture3D Dataset*. `http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/gestureData.tar.bz2`.

[116] Neil Robertson and Ian Reid. “A general method for human activity recognition in video”. In: *Computer Vision and Image Understanding* 104.2 (2006), pp. 232–248.

[117] Anna Rohrbach, Marcus Rohrbach, and Bernt Schiele. “The Long-Short Story of Movie Description”. In: *German Conference on Pattern Recognition (GCPR)*. 2015.

[118] Anna Rohrbach et al. “A Dataset for Movie Description”. In: *CVPR*. 2015.

[119] Mehmet Emre Sargin et al. “Audiovisual synchronization and fusion using canonical correlation analysis”. In: *Multimedia, IEEE Transactions on* 9.7 (2007), pp. 1396–1403.

[120] Christian Schüldt, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach". In: *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. Vol. 3. IEEE. 2004, pp. 32–36.

[121] Loren Arthur Schwarz et al. "Estimating human 3d pose from time-of-flight images based on geodesic distances and optical flow". In: *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*. IEEE. 2011, pp. 700–706.

[122] Pierre Sermanet et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229* (2013).

[123] Caifeng Shan, Shaogang Gong, and Peter W McOwan. "Beyond Facial Expressions: Learning Human Emotion from Body Gestures." In: *BMVC*. 2007, pp. 1–10.

[124] Caifeng Shan, Shaogang Gong, and Peter W McOwan. "Facial expression recognition based on local binary patterns: A comprehensive study". In: *Image and Vision Computing* 27.6 (2009), pp. 803–816.

[125] Jamie Shotton et al. "Real-time human pose recognition in parts from single depth images". In: *Communications of the ACM* 56.1 (2013), pp. 116–124.

[126] Behjat Siddiquie et al. "Affect analysis in natural human interaction using joint hidden conditional random fields". In: *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE. 2013, pp. 1–6.

[127] Nathan Silberman et al. "Indoor segmentation and support inference from RGBD images". In: *Computer Vision–ECCV 2012*. Springer, 2012, pp. 746–760.

[128] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014).

[129] Karen Simonyan and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos". In: *NIPS*. 2014.

[130] Richard Socher et al. "Parsing natural scenes and natural language with recursive neural networks". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 129–136.

[131] Fengyi Song, Xiaoyang Tan, and Songcan Chen. "Exploiting relationship between attributes for improved face verification". In: *CVIU* 122 (2014), pp. 143–154.

[132] Thad Starner and Alex Pentland. "Real-time american sign language recognition from video using hidden markov models". In: *Motion-Based Recognition*. Springer, 1997, pp. 227–243.

[133] Chen Sun and Ram Nevatia. "Active: Activity concept transitions in video event classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 913–920.

[134] M. Sun, P. Kohli, and J. Shotton. "Conditional Regression Forests for Human Pose Estimation". In: *CVPR*. 2012.

[135] Xinghua Sun, Mingyu Chen, and Alexander Hauptmann. "Action recognition via local descriptors and holistic features". In: *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE. 2009, pp. 58–65.

[136] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*. 2014, pp. 3104–3112.

[137] Shuai Tang et al. "Histogram of oriented normal vectors for object recognition with a depth sensor". In: *Computer Vision–ACCV 2012*. Springer, 2012, pp. 525–538.

[138] Jesse Thomason et al. "Integrating language and vision to generate natural language descriptions of videos in the wild". In: *Proceedings of the 25th International Conference on Computational Linguistics (COLING), August*. 2014.

[139] Christian Thurau and Václav Hlaváč. "Pose primitive based human action recognition in videos or still images". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE. 2008, pp. 1–8.

[140] Ying-li Tian, Takeo Kanade, and Jeffrey F Cohn. "Recognizing action units for facial expression analysis". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 23.2 (2001), pp. 97–115.

[141] Atousa Torabi et al. "Using descriptive video services to create a large data source for video annotation research". In: *arXiv preprint arXiv:1503.01070* (2015).

[142] Du Tran, Junsong Yuan, and David Forsyth. "Video event detection: From subvolume localization to spatiotemporal path search". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36.2 (2014), pp. 404–416.

[143] Du Tran et al. "Learning spatiotemporal features with 3D convolutional networks". In: *ICCV*. 2015, pp. 4489–4497.

[144] Hoang Trinh et al. "Hand tracking by binary quadratic programming and its application to retail activity recognition". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 1902–1909.

[145] Michel Valstar and Maja Pantic. "Fully automatic facial action unit detection and temporal analysis". In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE. 2006, pp. 149–149.

[146] Andrea Vedaldi and Brian Fulkerson. "VLFeat: An open and portable library of computer vision algorithms". In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM. 2010, pp. 1469–1472.

[147] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. "Human action recognition by representing 3d skeletons as points in a lie group". In: *CVPR*. IEEE. 2014, pp. 588–595.

[148] Subhashini Venugopalan et al. "Sequence to sequence - video to text". In: *ICCV*. 2015.

[149] Subhashini Venugopalan et al. “Translating videos to natural language using deep recurrent neural networks”. In: *NAACL-HLT*. 2015.

[150] Antonio W Vieira et al. “Stop: Space-time occupancy mail patterns for 3d action recognition from depth map sequences”. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Springer, 2012, pp. 252–259.

[151] Oriol Vinyals et al. “Show and tell: A neural image caption generator”. In: *arXiv preprint arXiv:1411.4555* (2014).

[152] Christian Vogler and Dimitris Metaxas. “Parallel hidden markov models for american sign language recognition”. In: *ICCV*. Vol. 1. IEEE. 1999, pp. 116–122.

[153] Chunyu Wang, Yizhou Wang, and Alan L Yuille. “An approach to pose-based action recognition”. In: *CVPR*. IEEE. 2013, pp. 915–922.

[154] Jiang Wang et al. “Mining actionlet ensemble for action recognition with depth cameras”. In: (2012), pp. 1290–1297.

[155] Jiang Wang et al. “Robust 3d action recognition with random occupancy patterns”. In: *ECCV*. Springer, 2012, pp. 872–885.

[156] Yang Wang and Greg Mori. “Learning a discriminative hidden part model for human action recognition”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1721–1728.

[157] Shikui Wei et al. “Multimodal fusion for video search reranking”. In: *Knowledge and Data Engineering, IEEE Transactions on* 22.8 (2010), pp. 1191–1199.

[158] Paul J Werbos. “Generalization of backpropagation with application to a recurrent gas market model”. In: *Neural Networks* 1.4 (1988), pp. 339–356.

[159] John Wright et al. “Robust face recognition via sparse representation”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.2 (2009), pp. 210–227.

[160] Qiuxia Wu et al. “Realistic human action recognition with audio context”. In: *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*. IEEE. 2010, pp. 288–293.

[161] Ying Wu and Thomas S Huang. “Vision-based gesture recognition: A review”. In: *Gesture-based communication in human-computer interaction*. Springer, 1999, pp. 103–115.

[162] Lu Xia and JK Aggarwal. “Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera”. In: *CVPR*. IEEE. 2013, pp. 2834–2841.

[163] Lu Xia, Chia-Chih Chen, and JK Aggarwal. “View invariant human action recognition using histograms of 3d joints”. In: *CVPR Workshops*. IEEE. 2012, pp. 20–27.

[164] Kelvin Xu et al. “Show, attend and tell: Neural image caption generation with visual attention”. In: *arXiv preprint arXiv:1502.03044* (2015).

[165] Yaser Yacoob and Larry S Davis. “Recognizing human facial expressions from long image sequences using optical flow”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.6 (1996), pp. 636–642.

[166] Meng Yang et al. “Fisher discrimination dictionary learning for sparse representation”. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE. 2011, pp. 543–550.

[167] Xiaodong Yang and YingLi Tian. “Eigenjoints-based action recognition using naive-bayes-nearest-neighbor”. In: *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*. IEEE. 2012, pp. 14–19.

[168] Xiaodong Yang and YingLi Tian. “Super Normal Vector for Activity Recognition Using Depth Sequences”. In: *CVPR*. 2014.

[169] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. “Recognizing actions using depth motion maps-based histograms of oriented gradients”. In: *ACM Multimedia*. ACM. 2012, pp. 1057–1060.

[170] Bangpeng Yao et al. “Human action recognition by learning bases of action attributes and parts”. In: *ICCV*. IEEE. 2011, pp. 1331–1338.

[171] Benjamin Z Yao et al. “I2t: Image parsing to text description”. In: *Proceedings of the IEEE* 98.8 (2010), pp. 1485–1508.

[172] Li Yao et al. “Describing videos by exploiting temporal structure”. In: *ICCV*. 2015, pp. 4507–4515.

[173] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. “The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection”. In: *ICCV*. IEEE. 2013, pp. 2752–2759.

[174] Matthew D Zeiler and Rob Fergus. “Visualizing and understanding convolutional networks”. In: *ECCV 2014*. Springer, 2014, pp. 818–833.

[175] Chenyang Zhang and Yingli Tian. “Edge enhanced depth motion map for dynamic hand gesture recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2013, pp. 500–505.

[176] Chenyang Zhang and Yingli Tian. “Rgb-d camera-based daily living activity recognition”. In: *Journal of Computer Vision and Image Processing* 2.4 (2012), p. 12.

[177] Chenyang Zhang, Xiaodong Yang, and YingLi Tian. “Histogram of 3D facets: A characteristic descriptor for hand gesture recognition”. In: *Automatic Face and Gesture Recognition (FG), 10th IEEE International Conference on*. IEEE. 2013, pp. 1–8.

[178] Ning Zhang et al. “PANDA: Pose Aligned Networks for Deep Attribute Modeling”. In: *arXiv preprint arXiv:1311.5591* (2013).

[179] Lin Zhong et al. “Learning active facial patches for expression analysis”. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012, pp. 2562–2569.