Human Action Recognition Using Deep Probabilistic Graphical Models

The University Of Sheffield.

Di Wu

Department of Electronic and Electrical Engineering

The University of Sheffield

A thesis submitted for the degree of

Doctor of Philosophy

Declaration

Parts of this report have been included in the following papers:

D. Wu, L. Shao, ”Deep Dynamic Neural Networks for Gesture Segmentation and Recognition”. In ECCV ChaLearn workshop 2014, Zurich (accepted).

D. Wu, L. Shao, ”Multimodal Dynamic Networks for Gesture Recognition”. In ACM Multimedia (MM) 2014, Orlando, USA (accepted).

D. Wu, L. Shao, ”Leveraging Hierarchical Parametric Network for Skeletal Joints Action Segmentation and Recognition”. In CVPR 2014, Columbus, USA.

L. Shao, D. Wu and X. Li, Learning Deep and Wide: A Spectral Method for Learning Deep Networks, In IEEE Transactions on Neural Networks and Learning Systems, 2014, doi: 10.1109/TNNLS.2014.2308519.

D. Wu, L. Shao, ”Multi-Max-Margin Support Vector Machine for Multi-Source Human Action Recognition”. Neurocomputing, vol. 127, pp. 98-103, Mar. 2014.

D. Wu, L. Shao,H. Zhang, ”One Shot Learning Gesture Recognition with Kinect Sensor”. ACM International Conference on Multimedia (MM), Nara, Japan, 2012.

D. Wu, F. Zhu, and L. Shao. ”One shot learning gesture recognition from RGBD images”. In IEEE Conference on CVPR 2012 workshop on gesture recognition (oral), Rhode Island.

D. Wu and L.Shao, Silhouette Analysis Based Action Recognition via Exploiting Human Poses, IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, pp. 236-243, Feb. 2013.

L. Shao, D. Wu and X. Chen, Action Recognition Using Correlogram of Body Poses and Spectral Regression, IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, September 2011.

Acknowledgements

First I would like to thank my advisor Ling Shao for his dedications towards work, professionalism, and allowing me to freely pursue and explore the field of computer vision and machine learning.

I am indebted to Chris Mower for his proofreading of this thesis. His dedication far surpasses my expectation and makes the revision process fun and enjoyable.

My thanks also go to my lab mates. It was very enlightening and entertaining working with Simon Jones whom I am constantly in awe of his programming skill sets and his broad interests in the field of machine learning. I would also like to express my thanks to Li Liu for all the refreshing conversations, Fan Zhu for the fun project in the year of 2012, Ruomei Yan for her witty insights and Bo Dong for his kind assistance.

My research life highlight should be the 3 months spent in Microsoft Research Cambridge where I was fortunate to meet some of the most talented minds and make acquaintance with some incredible friends. I am very grateful for my mentors Ben Glocker and Antonio Criminisi for their hugely positive and enriching working environment. During the development of the project, I benefited a lot working with Ben from the pair-coding, model visualization and slides presentation. I also thank Jakov Vogel, Joeri de Ruiter, Katja Hofmann, Matteo Venanzi, Martin Kiefel, Bhaskar Mitra, who, apart from being amazing intellectuals, make daily research a very fun job.

At last, I would like to thank my parents for their unfailing support, mutual respection and unconditional love.

Abstract

Building intelligent systems that are capable of representing or extracting high-level representations from high-dimensional sensory data lies at the core of solving many A.I. related tasks. Human action recognition is an important topic in computer vision that lies in high-dimensional space. Its applications include robotics, video surveillance, human-computer interaction, user interface design, and multi-media video retrieval amongst others.

A number of approaches have been proposed to extract representative features from high-dimensional temporal data, most commonly hard wired geometric or bio-inspired shape context features. This thesis first demonstrates some *ad-hoc* hand-crafted rules for effectively encoding motion features, and later elicits a more generic approach for incorporating structured feature learning and reasoning, *i.e.* deep probabilistic graphical models.

The hierarchial dynamic framework first extracts high level features and then uses the learned representation for estimating emission probability to infer action sequences. We show that better action recognition can be achieved by replacing gaussian mixture models by Deep Neural Networks that contain many layers of features to predict probability distributions over states of Markov Models. The framework can be easily extended to include an ergodic state to segment and recognise actions simultaneously.

The first part of the thesis focuses on analysis and applications of hand-crafted features for human action representation and classification. We show that the "hard coded" concept of correlogram can incorporate correlations between time domain sequences and we

further investigate multi-modal inputs, *e.g.* depth sensor input and its unique traits for action recognition.

The second part of this thesis focuses on marrying probabilistic graphical models with Deep Neural Networks (both Deep Belief Networks and Deep 3D Convolutional Neural Networks) for structured sequence prediction. The proposed Deep Dynamic Neural Network exhibits its general framework for structured 2D data representation and classification. This inspires us to further investigate for applying various graphical models for time-variant video sequences.

Contents

Chapter 1

Introduction

In recent years, human action recognition has drawn increasing attention of researchers, primarily due to its growing potential in areas such as video surveillance, robotics, human-computer interaction, user interface design, and multimedia video retrieval.

Previous works on video-based motion recognition focused on adapting handcrafted features and low-level hand-designed features have been heavily employed with much success. These methods usually have two stages: an optional feature detection stage followed by a feature description stage. Well-known feature detection methods ("interest point detectors") are Harris3D [1], Cuboids [2] and Hessian3D [3]. For descriptors, popular methods are Cuboids [4], HOG/HOF [1], HOG3D [5] and Extended SURF [3]. In a recent work of Wang *et al.* [6], dense trajectories with improved motion-based descriptors epitomized the pinnacle of handcrafted features and achieved state-of-the-art results on a variety of "in the wild" datasets. Given the current trends, challenges and interests in action recognition, this list would probably continue to spread out extensively.

In the evaluation paper of Wang *et al.* [7], one interesting finding is that there is no universally best hand-engineered feature for all datasets, suggesting that learning features directly from the dataset itself may be more advantageous. Albeit hand-crafted features are still the dominant approaches for visual recognition tasks, the approaches that derive from the learning perspectives [8, 9, 10] are gaining more and more momentums.

With the recent resurgence of neural networks invoked by Hinton and others [11], deep neural architectures have been proposed as an effective solution for extracting high level features from data. Deep artificial neural networks (including the family of recurrent neural networks) have won numerous contest in pattern recognition and representation learning. Schmidhuber [12] compiled a historical survey compactly summarising relevant works with more than 850 entries of credited works. Such models have been successfully applied to a plethora of different domains: the GPU-based cuda-convnet [13] classifies 1.2 million high-resolution images into 1000 different classes; multi-column Deep Neural Networks [14] achieve near-human performance on the handwritten digits and traffic signs recognition benchmarks; 3D Convolutional Neural Networks [15, 16] recognize human actions in surveillance videos; Deep Belief Networks combining with Hidden Markov Models [17, 18] for acoustic and skeletal joints modeling outperform the decade-dominating paradigm of GMM+HMM. In these fields, deep architectures have shown great capacity to discover and extract higher level relevant features.

However, direct and unconstrained learning of complex problems is difficult, since (1) the amount of required training data increases steeply with the complexity of the prediction model and (2) training highly complex models with very general learning algorithms is extremely difficult. It is therefore common practice to restrain the complexity of the model and this is generally done by operating on small patches to reduce the input dimension and diversity [10], or by training the model in an unsupervised manner [9], or by forcing the model parameters to be identical for different input locations (as in convolutional neural networks [13, 14, 15]).

Uncertainty is unavoidable in real-world applications, "as far as the laws of mathematics refer to reality, they are not certain, as far as they are certain, they do not refer to reality"(Albert Eistein, 1921). The Probabilistic Graphical Models, which generally incorporate models from Bayesian Networks(directed graphs) and Markov Random Fields(undirected graphs), derive the ideas from discrete data structure from the computer science to effectively encode, decode, transfer uncertainties in high-dimensional variables. Graphical Representation is both intuitive and compact for a data structure, it induces efficient reasoning

using general-purpose algorithms. Furthermore, by introducing conditional reason, sparse parameterisations of the model can be achieved via feasible elicitation or learning from data.

In this thesis, Deep Learning is represented by Probabilities Graphical Models framework as a building block for modeling time series data. This thesis has two main parts. In the first part, some novel hand crafted features for human action recognition are described. The second part of the thesis focuses on data driven analysis of acyclic video sequence labeling problems, *i.e.* video sequences are non-repetitive as opposed to longer repetitive activities, *e.g.* jogging, walking and running. With the immense popularity of Kinect [19], there has been renewed interest in developing methods for human gesture and action recognition from 3D skeletal data and depth images. A number of new datasets [20, 21, 22, 23] have provided researchers with the opportunity to design novel representations and algorithms and test them on a much larger number of sequences. The tasks of action recognition using 3D skeletal joints, at the first sight, seems trivial. However, due to the high dimensional space that 3D skeletal joints reside in and the amount of variation in human motion, the learning for the skeletal model requires latent states to empower the expressiveness of the model. This thesis also discussed the tasks of continuous action recognition, which are the real world's scenario and have been mostly ignored by researches. This thesis proposed a novel framework by introducing an ergodic states to achieve continuous action/gesture recognition.

Summary of Remaining Chapters

Chapter 2: Hand-crafted Features. This chapter describes hand crafted features for describing human actions in video sequences. A Bag-of-Correlated-Poses with soft-assignment scheme is proposed to encode the correlation within an action sequence. Later, the Motion History Images representation is extended as a holistic descriptor to compensate the semi-local Bag-of-Correlated-Poses scheme. The one-shot-learning scenario for gesture recognition is also studied. The Multi-view Spectral Embedding is proposed to fuse the information from RGB and depth images. With

the appearance-based HOG temporal segmentation and extended-MHI as feature representation, the proposed system is able to recognize gesture token in an unsegmented video sequence.

Chapter 3: Deep Belief Dynamic Networks. This chapter starts solving the action recognition problem from a machine learning perspective. It begins with a brief technical overview of Restricted Boltzmann Machines (RBMs) from an energy-based model aspect. The Gaussian Bernoulli visible layer for modeling real valued data is provided as the building block for Deep Belief Networks. The pre-training procedure for better initialization of the Deep Neural Networks and the learning hyper-parameters are also discussed. Then, a hierarchical dynamical framework that first extracts high level skeletal joints features is introduced for estimating the emission probability for the Hidden Markov Models in the place of Gaussian Mixture Models. The framework can also be easily extended to include an ergodic state to segment and recognize actions simultaneously. It's worth pointing out that the model has been designed with human action recognition using skeletal joints as input in mind, but it should also lend itself well to other high-dimensional time series.

Chapter 4: Deep 3D Convolutional Dynamic Networks. This chapter is built upon the powerful framework of Deep Convolutional Neural Network which achieves the state-of-the-art result in large scale image classification tasks. A Probabilistic Graphical Model unifies the Deep Neural Nets and Markov Field in a factor graph representation. A generalized hierarchical dynamic framework that first extracts high level features from contextual frames is proposed as the spatio-temporal learning representation. The 3D Deep Convolutional Neural Network is driven directly by the objective function, negates the time and energy consuming human effort in designing problem specific, sometimes suboptimal, hand crafted features. Experiments across various sensory input, *i.e.* RGB and depth, shows the Deep 3D Convolutional Dynamic Networks consistently perform on par with a wide variety of handcrafted features and other learning based methods.

Chapter 5: Multimodal Deep Dynamic Networks. This chapter unifies multi-modal dynamic networks from various sensory inputs. Two fusion pipelines, *i.e.* early fusion and late fusion scheme are experimented and both exhibit the evidence that the multi-modal dynamic networks enables share representation learning, outperforming individual modalities. The 3D ConvNet is also studied in more detail in this section and the visualization of the 3D ConvNet filters and the convolved spatio-temporal cuboids show that both shape pattern and motion pattern have been learnt by the 3D Deep ConvNet, reinforcing the conjecture that problem specific features could be learnt automatically.

Chapter 6: Conclusion and Future Directions. This chapter briefly summarises the contributions and discusses possible future research directions.

Chapter 2

Hand-crafted Features

2.1 Introduction and Related Works

This chapter describes the contributions for improving human action recognition by introducing frame-based correlation and multiview spectral embedding. The special case of one-shot-learning is also discussed.

Recent research has been local-feature focused. In their pioneering work, Dollar *et al.*[2] introduced an efficient approach for detecting spatio-temporal interest points (STIPs) using a temporal Gabor filter and a spatial Gaussian filter. Later, a number of other STIP detectors and descriptors have been proposed. The mainstream STIP detectors include Harris3D by Laptev *et al.*[24], Cuboid by Dollar *et al.*[2], 3D-Hessian by Willems *et al.*[25], Dense Sampling by Fei-Fei and Perona [26], Spatio-Temporal Regularity Based Feature (STRF) by Goodhart *et al.*[27]. And the mainstream STIP feature descriptors are HOG/HOF[24], HOG3D[28], Extended SURF[25] and MoSIFT[29]. The differences of various feature detectors and descriptors can be found in the survey paper [30]. Spatio-temporal features [2, 24] have shown success for many recognition tasks when pre-processing methods such as foreground segmentation and tracking are not possible, *e.g.* in the Hollywood dataset [31] or the UCF sports dataset [32]. However, their computational complexity hinders their applicability in real-time applications. Wang *et al.* [7] showed that the average time for spatio-temporal feature extraction varies from 0.9 FPS to 4.6 FPS, which makes the

spatio-temporal interest points (STIPs) features too time-consuming in computation. Another major limitation of the local feature based methods is that the sparse representation such as bag-of-visual-words (BoVW) discards geometric relationship of the features and hence is less discriminative. Hard-assignment quantization during the codebook construction for BoVW, which is usually realized by the k-means clustering algorithm, also makes the sparse representation less informative.

A human action can be viewed as a set of sequential silhouettes over time. Each silhouette records a pose of this action at a particular instant. Davis and Bobick [33] found that a human action can be recognized even when it is projected onto a single frame by incorporating partial time element. Gorelick *et al.* [34] treated human actions via silhouettes as three-dimensional shapes and the Poisson equation properties is adopted to obtain to space-time key features. Wang and Leckie [35] fused the global body shapes and local temporal motions from silhouettes and encoded human actions using the quantized dictionary from space-time windows. Shao and Chen [36] employed body poses sampled from silhouettes which are fed into a bag-of-features model. Similarly, Qu *et al.* [37] calculated the differences between frames and used them as intermediate features. Incorporating the local and holistic features, Sun *et al.* [38] unified the local 3D-SIFT descriptors and holistic Zernike motion energy image features.

2.2 From Local to Global

This section describes the contributions for improving human action recognition by introducing frame-based correlation from local to global as follows:

1. Correlogram of human poses in an action sequence is introduced to encode temporal structural information. We first extend the bag-of-features model to treat the silhouette in each frame as a feature. In the original bag-of-features representation, features are assigned to their closest cluster centers, also called visual words, and an entire video sequence is represented as an occurrence histogram of visual words. The traditional bag-of-features representation disregards structural information among the

visual words. To encode the structural information, Leibe *et al.* [39] proposed an implicit shape model using general Hough forest as a statistical method. We propose an explicit model to encode its temporal-structural information by constructing a correlogram which is a structurally more informative version of the histogram. The undesirable increase of dimension is suppressed by two stages of dimensionality reduction, *i.e.*, unsupervised principal component analysis (PCA) and supervised linear discriminant analysis (LDA).

2. Soft-assignment scheme/kernel codebook is extended for circumventing the quantization error penalty. Traditional bag-of-features model designates a feature by the cluster number it belongs to. And this hard assignment scheme may incur penalty for its quantization error. Gemert *et al.* [40] experimented four types of soft-assignment schemes for visual words encoding and demonstrated that explicitly modeling visual word assignment ambiguity improves classification performance compared to the hard-assignment of the traditional codebook model. Boureau *et al.* [41] investigated the relative significance of mid-level features in every single step of the system pipeline through extensive experimentation and evaluation of different types of encoding schemes for object and scene recognition. The adoption of *Mahalanobis distance* achieved the state-of-the-art performance for scene classification [42]. We extend their idea in our approach to maximize the preservation of information after the k-means clustering by assigning a feature proportionally according to its Mahalanobis distance to different cluster centers, so that a feature is no longer a discrete addition to the histogram bin but a continuous voting to multiple bins.

3. A holistic representation extension is proposed as a complimentary descriptor for local representation. As found by Sun *et al.* [38], local descriptors and global features emphasize distinctive aspects of actions and share complimentary properties. Motivated by their finding, we fuse the

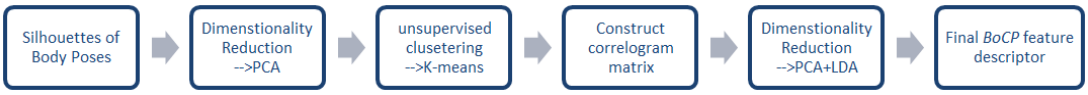


Figure 2.1: Flowchart of the *BoCP* model. Note the two phases of dimensionality reduction and their corresponding methods.

above temporally local descriptor with an extension of the holistic descriptor: motion history image *(MHI)* by adding gait energy information *(GEI)* and inversed recording *(INV)*. These two additional holistic descriptors serve as the compensation for the loss of information due to sequentially overlapping frames that is discarded in the original motion history image *(MHI)* representation. Then, a unified frame work is proposed to combine two distinct descriptors by early fusion (feature vector concatenation) and achieve further improvement over the separate methods.

2.2.1 Bag-of-Correlated-Poses *(BoCP)*

Fig.2.1 shows the flowchart of the construction process for the bag-of-correlated-poses *(BoCP)* representation. An action sequence is a series of pictorial frames. Most current approaches [2, 24, 28, 34] bundle action frames together as a monocular 3D volume representation. Unlike traditional 3D volume representation, we treat each frame individually as an atomic input. The notion of body poses in our approach is represented by the silhouettes as in Fig.2.2. A bounding box, *i.e.* the smallest rectangle containing the human figure, is applied to each frame of the silhouette sequence and then is normalized to a fixed size. The prepossessing steps reduce the original dimension and remove global scale and translation variations. The interpolation during the normalization process suppresses the noise as the morphological transformations (dilation and erosion) can isolate individual elements and join disparate elements in an image. The rectangular region of interest (ROI) mask serves as image unification in each action frame, making recognition invariant to body size as well as scale and translation variations resulting from perspective changes.

Figure 2.2: Left: Illustration of the "cross arms" action; Right: A normalized silhouette.

2.2.1.1 Codebook Creation

The extracted normalized silhouettes are used as input features for the *BoCP* model. The traditional BoVW model [2], [24] in action recognition is based on the detection and description of STIPs as input whilst our approach is based on holistic silhouettes. Due to the usage of pose silhouettes, the local feature detection and description steps in a traditional BoVW method are not required. In the interest point based action recognition method, each feature vector is a 3D descriptor calculated around a detected interest point in an action sequence. In our method, each feature vector is converted from the 2D silhouette mask to a 1D vector by scanning the mask from top-left to bottom-right pixel by pixel. Therefore, each frame at the time $\mathbf{t}$ in an action sequence is represented as a vector of binary elements $\mathbf{F_t}$, the length of which is $\mathbf{L} = \mathbf{m} \times \mathbf{n}$, where $\mathbf{m}$ and $\mathbf{n}$ are dimensions of the normalized pose silhouette. Suppose the $\mathbf{i^{th}}$ action sequence consists of $\mathbf{S_i}$ frames, then an action sequence can be represented as a matrix $\mathbf{X} \in \mathbb{R}^{\mathbf{m} \times \mathbf{n}}$. Each row of the matrix stands for a single frame. Therefore, for a training set with $\mathbf{n}$ action sequences, the whole training dataset can be represented as:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_n \end{bmatrix} \tag{2.1}$$

The total number of rows, which is also the total frame number in the training dataset, is $\mathbf{S} = \mathbf{S_1} + \mathbf{S_2} + \ldots + \mathbf{S_n}$. Because features are in high-dimensional space, we first use PCA for dimensionality reduction. Hence, each frame $\mathfrak{F}_t$ is projected into a lower dimension $\widetilde{\mathbf{F}}_\mathbf{t}$.Then visual vocabulary can be constructed by clustering feature vectors obtained from all the training samples using the

k-means algorithm. The center of each cluster is defined as a codeword, and the size of the visual vocabulary is the number of clusters $\mathbf{k}$.

2.2.1.2 Soft Assignment Scheme

In the traditional BoVW model [2], [24] each feature vector can be assigned to its closet codeword based on the Euclidean distance. The visual word vocabulary in the code-book framework can be composted in various methods, *e.g.* GMM, spectral clustering, etc. One typical way of constructing a code book is k-means clustering. However, the crucial presumption is that the feature (from an image, or a video sequence) could be well represented as a discrete visual word. The k-means algorithm minimizes the variance between the data and the clusters. Hence, the most frequent appearing features will be assigned as the clusters. Nevertheless, the most frequent features are not necessarily the most discriminative [40] and visual appearance, instead of being discrete, is naturally continuous and the discretizing the representative visual word would be problematic. Assigning a feature to its single cluster gives rise to loss of information due to quantization errors, especially for features residing on boundaries of neighboring clusters. Thus, in our approach, we model our visual words by a kernel codebook to integrate the visual word ambiguity. Kernel density estimation is an alternative to the discrete histogramming which is inherently more robust for estimating a probability density function. In the case of the soft-assignment scheme, the most common Gaussian kernel $\mathbf{K}_\sigma = exp(-\frac{1}{2}\frac{x^2}{\sigma^2})$ assumes that normal distribution between a visual feature and a codeword with a smoothing parameter σ. Therefore, we adopt this statistically viable kernel function and a visual word $\mathbf{W_{i,t}}$ can be described as:

$$\mathbf{W_{i,t}} = exp(-\frac{\left\|\widetilde{\mathbf{F}}_\mathbf{t} - \mathbf{C_i}\right\|^2}{2\sigma^2}) \qquad \text{i=1,2,...,k;} \tag{2.2}$$

where $\widetilde{\mathbf{F}}_\mathbf{t}$ is the projected low dimension frame vector at time $\mathbf{t}$, $\mathbf{C_i}$ is the $\mathbf{i^{th}}$ cluster center, $\mathbf{k}$ is the number of clusters and the smoothing parameter σ determines the degree of similarity between data samples. Note that this degree of affinity is dataset dependent with respect to dimensionality of the features,

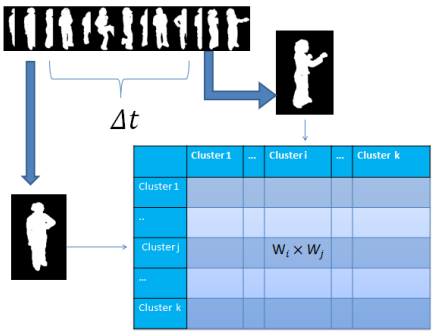
and the range of the feature values. The goal is to find the kernel size that discriminates best between classes. This pesky hyper-parameter is generally set by cross-validation.

2.2.1.3 Correlogram Of Poses

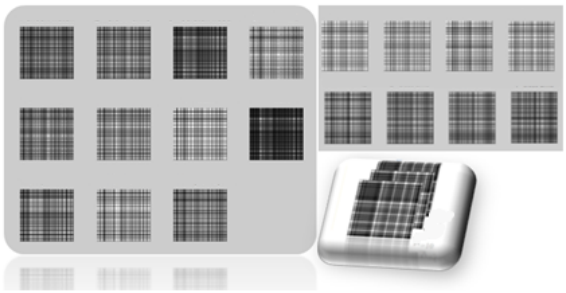
As the current BoVW model discards all geometric information and based on the 2D picture correlogram[43], we extend the original BoVW model to *BoCP* which takes advantage of the temporally local features whilst maintaining the holistic geometric information. The concept of correlogram was first introduced by Huang *et al.*[44], where they used colour correlogram for image indexing. A colour correlogram is a three-dimensional matrix where each element indicates the co-occurrence of two colours which are at a certain distance from each other. Therefore a correlogram encodes more structural information than a flat histogram. Similarly, for action representation, each element in *BoCP* denotes the probabilistic co-occurrence of two body poses taking place at a certain time difference from each other. Note that we use the word "probabilistic" here because our visual word is not a discrete codebook number but a probabilistic distribution of multiple poses. Fig. 2.3a illustrates the construction of a correlogram at a single temporal distance in our *BoCP* model. Since the poses are divided into $\mathbf{k}$ clusters, the dimensionality of the correlogram matrix at a fixed time offset Δt is $\mathbf{k} \times \mathbf{k}$, where $\mathbf{k}$ represents the codeword number of the constructed codebook. Each entry in the BoCP matrix can be defined as:

$$\mathbf{E}(i, j; \delta t) = \sum_{t=1}^{\mathbf{S_T}-\delta t} \mathbf{W}_{i,t} * \mathbf{W}_{i,t+\delta t}; \tag{2.3}$$

where δt specifies the time offset, $\mathbf{W_{i,t}}$ is the frame $\widetilde{\mathbf{F}}_\mathbf{t}$'s visual word probability to cluster $\mathbf{i}$ in Eq.(2.2), $\mathbf{S_T}$ is the number of frames in the action sequence. Note that the correlogram matrix in Eq.(2.3) can be obtained by assigning a number of different time offsets δt. Multiple time offsets scheme will accordingly enhance distinctiveness of correlogram representation but at the cost of the increase of dimensionality of the feature descriptor and the computational time. In our implementation, four time offsets of 2, 4, 6 and 8 are employed. Fig.2.3b



(a) Illustration of the correlogram in the *BoCP*. **W** correspond the visual words defined in Eq.(2.2). Inputs are a series of silhouettes. For every pair of two frames with time interval δt, each entry in the correlogram is the multiplication of two weights corresponding to these two poses.



(b) Left: Correlogram matrices of different actions, *i.e.* *"check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave punch, kick, pick up"*, performed by the same person. Top Right: two rows of two actions: *"walk"* and *"cross arms"* performed by different people; Bottom Right: correlogram matrices with different time offsets.

Figure 2.3

depicts examples of correlogram matrices. On the left side are correlogram matrices of different actions performed by the same subject. It is even visually possible to distinguish the difference in texture between different actions' correlograms. On the top right, row 1 and row 2 are correlogram matrices of 2 actions performed by different people. We can observe that the correlogram matrices of the same action look much more similar than those of different actions, which makes correlogram a discriminative representation for human actions.

2.2.1.4 Dimensionality Reduction

Both the original binary silhouette features and the final correlogram representations are in very high dimensionality. Due to the "curse of dimensionality", it is impractical to use the original long feature vectors for classification. Therefore, the use of a dimensionality reduction method is necessary. There are two stages where dimensionality reduction is needed in our algorithm. The first stage is before feeding silhouette feature vectors into the k-means clustering process and the second stage is for the reduction of the final correlogram rep-

resentations. We adopt the unsupervised principal component analysis (PCA) at the first stage and the combination of PCA and supervised linear discriminant analysis (LDA) at the second stage. PCA seeks projection directions that maximize the variance of the data and LDA maps the features to make them more discriminative. Our argument for adopting different dimensionality reduction methods is as follows: at the first stage, a certain silhouette pose may appear in different action classes and the class label information is not very relevant. Therefore, an unsupervised method, *i.e.*, PCA, is used. At the second stage, each correlogram matrix is at a high dimension (in a scale of 10^3D), we first project the correlogram matrix into a lower dimension of 100D using PCA and because each individual correlogram matrix belongs to one unique action class, we further reduce the dimension to the number of *action class -1* using LDA.

2.2.2 Extended-MHI

As found by Sun *et al.* [38], local features and holistic descriptors emphasize different facets and share complimentary properties. Motivated by their finding, we fuse the above temporally local descriptor *BoCP* with an extension of the holistic descriptor: *MHI* by adding Gait Energy Information (*GEI*) and Inverse Coding (*INV*). These two additional holistic descriptors serve as the compensation for the loss of information due to sequentially overlapping frames lost in the original *MHI* representation. We deduce our approach by first introducing motion templates:

Motion Templates: motion energy images (*MEI*) and motion history images (*MHI*) proposed by Davis and Bobick [33] are used to represent the motions of an object in video. All frames in a video sequence are projected onto one image (MHI/MEI) across the temporal axis. As to where and how motion happens are recorded in the images, *MHI* captures the temporal information of the motion in a sequence. Assume $I(x, y, t)$ is an image sequence and let $B(x, y, t)$ be a binary image sequence indicating regions of motion, which can be obtained from image differencing. The binary *MEI* $E_\tau(x, y, t)$ with the temporal extent

of duration τ is defined as:

$$E_\tau = \cup_{i=1}^{\tau-1} B(x, y, t-1) \tag{2.4}$$

The *MHI* $H_\tau(x, y, t)$ is used to represent how the motion image is moving, and is obtained with a simple replacement and decay operator:

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if} \quad B(x, y, t) = 1, \\ max(0, H_\tau(x, y, t-1) - 1) & \text{otherwise.} \end{cases} \tag{2.5}$$

It is stated in [38] that "empirical experimentations demonstrate that the cycle of a single action in two benchmark datasets: the KTH [45] and the Weizmann [34] datasets can be as short as 5 frames". Hence, in their experiments the duration of a single action is set to 5 frames. However, for non-repetitive actions, such as those in the IXMAS [46] dataset, choosing a τ is impractical. Notwithstanding, we observe that the larger τ, the more information is encoded. Therefore, we set τ as the duration of the whole action T, and generally it's up to 100 frames in most action sequences. The re-defined version of MHI is:

$$\widetilde{H}(x, y, t) = \begin{cases} T & \text{if} \quad B(x, y, t) = 1, \\ \widetilde{H}(x, y, t-1) - 1 & \text{otherwise.} \end{cases} \tag{2.6}$$

Note that there is no maximum operator in front of $\widetilde{H_\tau}$ cf. Eq. (2.11) because setting τ as the sequence duration will lead to non-negativity of $\widetilde{H_\tau}$.

We further extend motion templates that include two more elements: *GEI* and *INV*.

GEI is to compensate for the non-moving regions and the multiple-motion-instants regions of the action. The summation of all binary silhouette images and normalization of the pixel value define *GEI*:

$$G(x, y) = \frac{1}{\tau} \sum_{t=1}^{\tau} B(x, y, t) \tag{2.7}$$

INV is used to recover the loss of initial frames' action information when

setting τ as the whole action duration and is defined as follows:

$$\widetilde{I}_{\tau}(x,y,t) = \begin{cases} \tau & \text{if} \quad B(x,y,t) = 1, \\ \widetilde{I}_{\tau}(x,y,t+1) - 1 & \text{otherwise.} \end{cases} \tag{2.8}$$

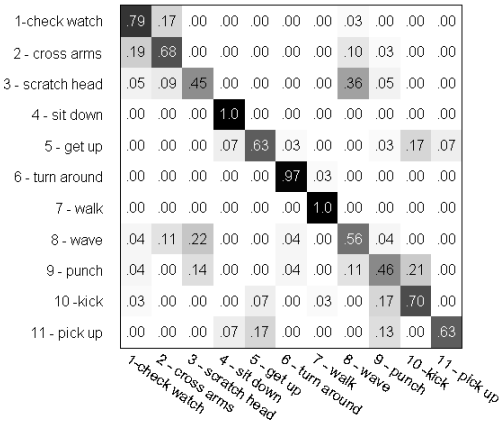
Note that its subtle difference to Eq. (2.12) is the time variable becomes $t+1$ instead of $t-1$ from which we get the name *Inversed Recording*.

We reason that *MHI* is poor at representing repetitive actions from Fig. 2.4a and Fig. 2.4b: the confusion matrix of the original *MHI* on the IXMAS dataset with τ as the whole action duration shows that "wave" and "scratch head" are the two most difficult actions to be distinguished, because they have similar motion patterns, *i.e.*, repetition of hand movement at similar spatial locations. *INV* provides complementary information by emphasizing (assigning larger value) at initial motion frames instead of the last motion frames as in *MHI*. Fig. 2.4b illustrates the similarities and differences between *MHI*, *INV* and *GEI* of these two action sequences. The first columns are the *MHI* projections, second are the *INV* projections and the last are the *GEI* projections. The top row corresponds to the "wave" action and the bottom "scratch head". Again, the projection graphs show that *MHI* emphasizes recent motion (ending frames) whilst *INV* displays the opposite. Hence the combination of the two is complementary. Furthermore, *GEI* encodes the supplementary information in repetitive actions where both *MHI* and *INV* are poor at representing. The experimental results in Section Roman4 prove the viability of our conjecture.

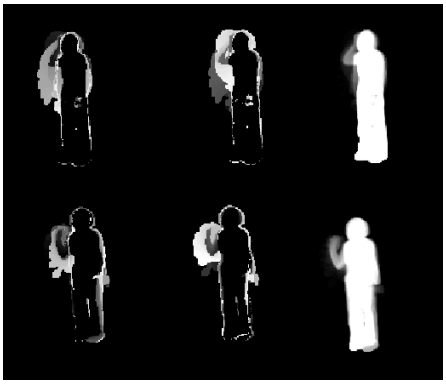
2.2.3 Experimental Results

Datasets and Experimental Setup

We evaluate our approach on two public action recognition datasets: Weizmann [34] and Inria Xmas Motion Acquisition Sequences (IXMAS) [46]. Details of the two datasets can be found at 6.1.7 and 6.1.8. For the IXMAS dataset, we only use single camera's data for training/testing and follow the widely adopted leave-one-actor-out testing strategy. For the Weizmann dataset, we adopt the leave-one-sequence-out testing strategy. We choose the following



(a) Setting τ as the whole action duration in *MHI*, the confusion matrix of the IXMAS dataset (Camera 3). Actions "wave" and "scratch head" are difficult to distinguish because they have similar motion patterns, *i.e.* repetition of hand movement at similar spatial location.



(b) Illustration of the *MHI*, *INV* and *GEI*– top: "scrath head"; bottom: "wave". The projection graphs show that *MHI* emphasizes recent motion (ending frames) whilst *INV* the opposite. *GEI* encodes the average gait information and is supplementary in repetitive actions where both *MHI* and *INV* are poor at representing.

Figure 2.4

parameter settings: the bounding box of silhouettes is 30×20 pixels; and feature vectors are reduced to the dimension of 30 using PCA. During visual vocabulary construction, $k = 30$ is used for the k-means clustering, which results in 30 codewords. Four different time offsets for $\delta t = 2, 4, 6, 8$ frames, are used for the construction of correlogram; hence the dimensionality of a *BoCP* representation is $30 \times 30 \times 4 = 3600D$. Each *BoCP* representation is then reduced to the dimension of the number of action *class-1* using the combination of PCA (100D) and LDA. Then a unified framework is proposed to combine the two distinctive descriptors: *BoCP* and *Extended-MHI* by early fusion based on a very intuitive notion: local descriptor *(BoCP)* and holistic descriptor *(Extended-MHI)* are complementary to each other. For final classification, the Gaussian kernel SVM classifier is adopted.

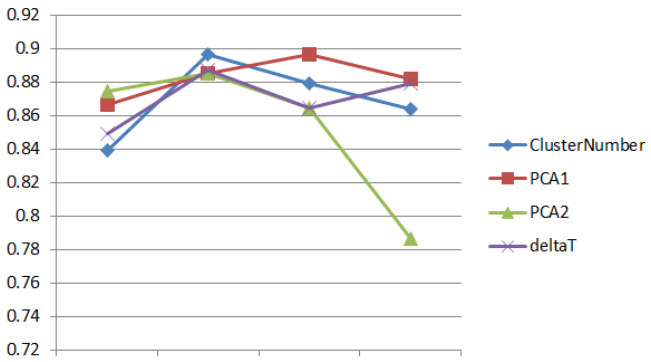


Figure 2.5: Parameters sensitivity test of *BoCP*. The choices of four parameters (from left to right): cluster number: 10,20,30,40; PCA (stage 1) dimension: 10,20,30,40; PCA (stage 2) dimension:50,100,200,400; δt (frames): 2,2-4,2-4-6,2-4-6-8. It can be seen that the proposed model is insensitive to various hyperparameters.

2.2.3.1 Parameter Sensitivity Test

There are a few parameters in the construction process of our *BoCP* model and we demonstrate that our *BoCP* model is insensitive to the choice of parameters in Fig. 2.5 using the IXMAS dataset. Four main parameters are tested: time offset δt in Eq. (2.3); cluster number k during the k-means clustering; the reduced dimension after applying PCA at the first stage; the reduced dimension after applying PCA at the second stage. It can be seen that the *BoCP* model is rather robust to the choice of different parameters as the overall accuracy varies in the range of 2%-7%. The most error-prone parameter is the second stage dimension of PCA: excessively large dimension may lead to worsen performance.

2.2.3.2 Visual Word Ambiguity Effect

Through Fig.2.5 we are also able to examine the effect of the *soft assignment strategy*. Note that the cross-validation-optimal number of clusters in our model is quite small: 30 comparing with the traditional BoVW model [2, 7, 24, 25, 28, 31], which usually surges up to 1000 and more. The graph also shows that a surprisingly small cluster number, *i.e.* a cluster number of 10, still achieves comparable result but a larger cluster number does not improve the overall

Methods	Cam1	Cam2	Cam3	Cam4	Cam5
BoCP+Extended-MHI	83.6	**90.3**	**89.4**	**89.8**	**78.8**
BoCP	81.4	87.6	84.9	88.5	71.3
Shao and Chen [36] ***HBP***	63.7	70.2	67.2	68.1	66.9
Weinland *et al.* [47] Local Product	**85.8**	86.4	88.0	88.2	74.7
Varma et Babu [48] GMKL	76.4	74.5	73.6	71.8	60.4
Wu *et al.* [49] AFMKL	81.9	80.1	77.1	77.6	73.4
Junejo *et al.* [50] PMK-NUP	76.4	77.6	73.6	68.8	66.1

Table 2.1: Performance comparison of different methods in five cameras on the IXMAS dataset.

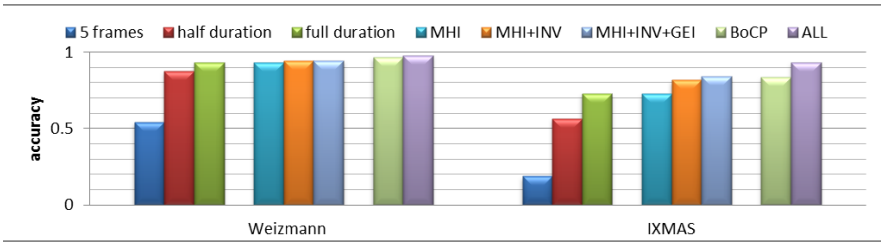


Figure 2.6: Algorithms for comparison. The first three columns of two datasets correspond to three values of τ in Eq.(2.11): 5 frames, half duration of the action, and the whole duration of the action. The following three columns correspond to *MHI* only, *MHI+INV* and *MHI+INV+GEI*. The last two columns evaluate *BoCP* and the effect of combining two descriptors: *BoCP* and *Extended-MHI* ("All" stands for the combination of the two desecriptors).

recognition accuracy. The reason behind the scene is that in our *BoCP* model, a visual word is described by a vector whose elements are related to the distances to the cluster center, *i.e.* Eq.(2.2), instead of the quantized class membership. Moreover, we further utilize this probabilistic visual word descriptor for the entries in Eq. (2.3) so that a frame spans over all the columns in the correlogram matrix. Observe that our *BoCP* model is a matrix of dimension $\mathbf{k} \times \mathbf{k}$ where $\mathbf{k}$ is the number of clusters. Again, a larger cluster number increases the computational complexity during k-means clustering $O(k)$ and *BoCP* correlogram construction $O(k^2)$ exponentially. By adopting the soft assignment scheme, we tactically circumvent the situation where k can be easily up to 1000, making the *BoCP* dimension up to what is simply impractical computationally and unnecessary for action representation. Thus, we gracefully achieve an action descriptor with a dimension in the scale of thousands while

encoding temporal structural information.

Algorithms for Comparison

We first make a quantitative comparison between our correlogram based method *(BoCP)* and the histogram based method *(HBP)* [36] in Table 2.2 and it can be seen that our proposed method *BoCP* consistently outperforms the histogram based method. We then verify our choice of time duration in Eq. (2.12) over Eq. (2.11) through Fig. 2.6. The first three columns of two datasets correspond to three values of τ in Eq. (2.11): 5 frames (suggested by [38]), half duration of the action and the whole duration of the action. Setting τ as the whole action duration achieves the best results among the three schemes. The following three columns summarize the efficacy of concatenating different versions of motion templates described in Section Roman3. The consistent improvement of accuracy validates that both *GEI* and *INV* are complementary to the original *MHI*. Then, by early fusion (direct concatenation) of two descriptors: *BoCP* and *Extended-MHI*, we show in the last two columns that the combination of temporal-local and holistic descriptors further improve the overall accuracy by 1% for the Weizmann dataset (because the improvement for the Weizmann dataset's accuracy is close to saturation: 97.78%, 88/90) and 3.6% for the IXMAS dataset. Again, this explains the complementary characteristics between the *BoCP* model and the *Extended-MHI* representation: *BoCP* is good at representing temporal pose correlations and the *Extended-MHI* excels at holistic motion representation. In addition, *BoCP* compensates for MHI's ineffectiveness in representing repetitive movements. Table 2.2 shows the comparison to the state-of-the-art methods on the IXMAS dataset using a single camera view. Our algorithm outperforms the state-of-the-art methods on the challenging free viewpoint IXMAS dataset and this demonstrates that our model is robust to large variations of viewpoint, position and orientation. The experiments were done on an Intel 2-core 3GHz CPU and 4GB memory PC in a single thread running MATLAB. The speed of our method is approximately 25 FPS (excluding the silhouette extraction process).

2.3 RGBD Images and One-Shot-Learning

In this section, the one shot learning scenario is studied with respect to depth images' special idiosyncrasies.

With the revolutionizing affordable Kinect sensor and the graceful core algorithm of , a single input depth image can be segmented into a dense probabilistic body part labeling whilst simultaneously achieving two goals: computational efficiency and robustness. One key component of their success lies in the huge and highly varied training data: both from realistic and synthetic depth images, a total number of 1 million images were used to train the deep randomized decision forest classifier in order to avoid overfitting. However, even the best existing depth image-based systems still exhibit limitations: in the system of [19], an effective depth sensor distance is required from 1.2m to 3.5m. Outside the effective range, meaningful skeletal joints are unlikely to be generated. Various applications spawned after the inception of this consumer priced 3D camera: scene flow estimation using a particle filter was formulated in [51]; human activity detection from RGBD images based on a hierarchical MEMM was studied in [52]; CHALEARN Gesture Challenge in [53], *etc* .

We focus on the CHALEARN Gesture Challenge [53]. There are some unique distinctions in this dataset from other action/gesture recognition datasets [45, 54]. We reinstate the major easy/difficult aspects of the dataset and present our analysis and reasoning to solve/circumvent the problems as follows:

1.Availability of depth camera: depth cameras significantly reduce the huge colour and texture variability induced by clothing, hair and skin. However, some imperfection/noise of various sources still exists [55] in current depth sensors: *e.g.* reflectance and mismatched patterns. *cf* to Figure 2.7, strong existence of "salt and pepper" noise is detected as real motion information. A spatial filtering and a morphological preprocessing step are required for noise reduction.

2.Multiple gestures in testing set: temporally unsegmented action sequences are real-world scenario. However, present action/gesture recognition datasets almost universally dodge this difficulty by providing training/testing sequences in a manually segmented manner. In the dataset of [53], however, the number

Figure 2.7: Noise in depth image.

of gestures contained in a testing video sequence varies from 1 to 5. Therefore, temporal segmentation is a precondition for gesture recognition. Note that current action localization methods [56, 57] also provide solutions for effectively retrieving action sequence from video sequences. We argue, however, that because of the unique property of this dataset, *i.e.* hands return to a resting position between each pair of neighboring gestures, the temporal segmentation as a preprocessing step is more effective than the action localization approach. Weinland *et al.* [58] presents a semi-supervised action recognition system that breaks down action sequences into primitive actions based on a motion history volume descriptor and automatically discovers the action taxonomies. Similarly, as suggested by [53], since the assistants hands return to a resting position between each pair of neighboring gestures, segmentation points occur near the peaks of hand motions in the lower part of an image. In our system, instead of using motion information for action segmentation, we adopt the appearance-based approach and achieve 5% error in the metrics of *Levenshtein distance* for the verification of segmentation. Also note that the accuracy for our whole gesture recognition system is upper bounded by this temporal segmentation performance.

3.One-shot-learning: only one training example of each class is considered as the unique trait of this challenge whilst using more examples per sign typically improves accuracy (see, *e.g.* [59, 60]). The standard tools of statistical machine learning, *e.g.* classification and regression, have a chance to be equally matched to modeling purposeful behavior in a poor manner; an agent's goals often succinctly, but implicitly, encode a strategy that would require tremendous amounts of data to learn. We discuss our experimental result on two

classes of machine learning models: generative algorithms, and discriminative algorithms and conclude that discriminative classifiers are more suitable for solving this one batch one lexicon one learning token dataset due to lack of training data. More specifically and surprisingly, the simplest correlation coefficient discriminating method, which defines the statistical independence, works best among other popular classifiers, *e.g.* nearest neighbor, SVM, Random Forest, *etc* .

4.Depth & RGB camera decision fusion: how to effectively utilize multiple inputs to generate an informed decision is sometimes under-appreciated. Currently, the most commonly adopted approach when encountering different types/spectra of features is to concatenate multiple features into a long vector before the classification stage and feed this long feature vector into a classifier [7, 38, 61]. In [31], a handcrafted weights was adopted for merging the classification score. Mostly, for the sake of simplification, different view features are treated independently and have been ignored by their intrinsic relationships [47, 62, 63]. We argue that the interleaving relationship between different feature vectors is lost during this brute-concatenation process, and the interdependent relationship between different feature decisions could be better incorporated in an ensemble system. Moreover, the benefit of multi-spectrum video fusion always comes with a certain cost and complexity in the analysis process due to the fact that the involved modalities have different characteristics. On one hand, the more pronounced the independence between difference modalities, the more complementary information can be gleaned from each of them. On the other hand, there need to be a sufficient amount of correlations in order to be able to link features in one modality. We study the Multiview Spectral Embedding (MSE) in [64] and its derivative of spectral clustering. Then we present our discovery of the intrinsic property during the embedding process. With the brief theoretical analysis, we demonstrate the effectiveness of the proposed approach by embedding information acquired from both depth and RGB cameras to further improve the recognition rate.

2.3.1 Experimental Results

In this section we detail our approach towards solving the general four issues aforementioned and present both quantitative results and qualitative evaluations of our method on the CHALEARN dataset [53]. The details of the dataset can be found at 6.1.2.

Error Metrics. We quantify our recognition rate by computing the *Levenshtein distance* between the list of predicted labels R and the corresponding list of true labels T, that is the minimum number of edit operations (substitution, insertion or deletion) that one has to perform to go from R to T (or vice versa). This error metrics measurement is also in accordance with the Leaderboard in [53] and we refer this error metrics as $\mathfrak{LD}$ from now on.

2.3.1.1 Preprocessing: Background Separation and Noise Reduction for Depth Images

We take advantage of the depth sensor unique property from which human silhouettes can be easily segmented. Firstly we segment human from the background using Otsu's method of global image threshold [65] as shown in Figure 2.8 (top left). The resulting noise pattern in depth images resembles salt and pepper noise. We then use a spatial filtering and a morphological process for noise reduction. A median filter provides excellent "salt and pepper" noise reduction with considerably less blurring. As in [55], we adopt a 5×5 aperture median filter. Then, morphological process is used for further noise reduction. Specifically, we use opening operation which consists of erosion followed by dilation to smooth the outers, split the narrow region and remove the thin perimeter. Thus, the opening operation removes randomly generated noise and smooths the original image. The resulting depth image is shown in Fig.2.8 (top right). When the noise reduction method is applied to the motion image generated from the depth sensor, the resulting motion description is less prone to faulty defects from the depth sensor. These operations are highly effective for the depth image noise reduction especially if the action descriptor is motion-based as in our system. Experimental result shows that the noise reduction method can improve the performance in terms of $\mathfrak{LD}$ up to 9%.

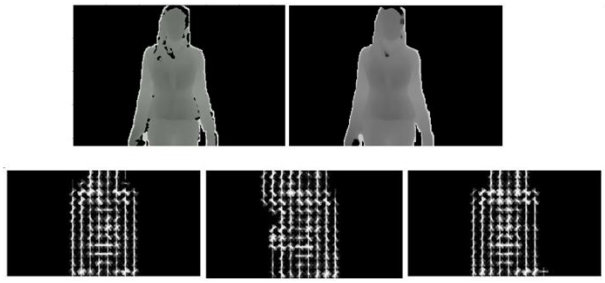


Figure 2.8: Top left: background segmentation; top right: depth image after noising; Bottom row: HOG descriptor for temporal segmentation. As it can be seen that the starting frame(bottom left) and the ending frame(bottom right) are quite similar to each other whereas in the midst of action (bottom middle), there is a substantially spatial difference.

2.3.1.2 Temporal Segmentation

For temporal segmentation, we adopt the *appearance based* approach. Because hands return to a resting position between each pair of neighboring gestures, we aspire to find the frames that are similar to the beginning and ending frame in the unsegmented testing video sequence and define them as the interval frames between two gestures in a video sequence. A simple but effective option to retrieve similar frames is to divide a single frame into an $N \times N$ lattice and use the histogram of oriented gradients (HOG) [66] as the cell descriptor with B number of bins. Hence, a single frame can be represented as a feature vector of $N \times N \times B$ dimension. Then we use the k nearest neighbor approach to search for frames that are similar to the beginning and ending frame. Some implementation details worth mentioning: first how many similar frames should we search in this unsegmented video? Our solution is to first store the training example's average frame number L and when there is a test sequence, we make a rough estimation of gesture number as the quotient Q of test frame number F and the average training tokens' frame number L. Then, the estimated frame number to retrieve is $8 \times Q$. The reason we choose 8 here is rather arbitrary as it makes little influence to the final performance. After the similar frames being retrieved, a max pooling approach [67] is used to aggregate interval frames. We then merge minimal segmented sequences if the total action tokens segmented exceed the number of 5, which is the maximum gesture number for one test sequence in this dataset. Generally, the more lat-

tices that one frame picture is divided into, the more accurate it is to segment action sequence. However, through our experiments, the varying of HOG grid size has little impact on temporal segmentation performance. Consequently, taking the computational cost into consideration, there is no significant point to set the size of the HOG grid into small values. In our experiment, bin number B is 9 and two lattice types were tested, *i.e.* 8×8 and 16×16. In the case of 8×8, $\mathfrak{LD}$ is 6.764% and for 16×16 is 5.235%. Note that as we mentioned in Section 2, accuracy for our whole gesture recognition system is upper bounded by this temporal segmentation performance.

2.3.1.3 Motion Descriptors and Scheme for Classifier

We experiment extensively on different motion descriptors and classifiers and via comparison we discuss our methodological insights. Our final adopted approach is *Extended-MHI* for action descriptor and *Maximum Correlation Coefficient* for classifier. The results are reported on the first 20 development batches unless otherwise we explicitly state on the validation dataset.

Cons for local method: Spatio-temporal features [2, 24] have shown success for many recognition tasks where preprocessing methods such as foreground segmentation and tracking are not possible. However, their computational complexity hinders their applicability in real-time applications. Wang et al.[7] showed that the average time for spatio-temporal feature extraction varies from 0.9 FPS to 4.6 FPS, which makes the STIP features too time-consuming in computation. Another major limitation of the local feature based methods is that the sparse representation such as *BoVW* discards geometric relationship of the features and hence is less discriminative. We experiment on depth image using Dollar's method [2] for STIP detection, HOG3D [28] for cuboid descriptor, kernel codebook [40] for encoding and SVM [68] for classifying BoVW model. The result shows that the $\mathfrak{LD}$ is merely 0.7232 and is even worse than the baseline of 0.5998. We argue that the reasons behind local BoVW method's ineffectiveness in gesture recognition lie in the following two aspects: 1) low interclass variation between different gestures make local methods and their corresponding descriptors less discriminative. *cf* Figure 2.9, although motion

Figure 2.9: Spatial temporal interest points in white bounding box of three different gesture tokens.

interest points have been successfully detected around arms and hands area, similarity of interest points around bending elbows could hinder the discriminative power of local patch; 2) one-shot-learning renders it difficult to distinguish the most informative local patch in a BoVW model, especially temporal sequence has been discarded through the construction process of histograms. Insufficient training examples would be very likely to lead to the failure in this histogram based approach.

Cons for generative methods: Under the one-shot-learning configuration, for some discriminative models, one-shot-learning also restrains their discriminative power: *e.g.* for SVM, a single training example can not effectively define its hyperplane for discriminating multi-class; for Adaboost, certain quantity of positive and negative examples are needed to train the weak classifiers; the decision trees methods, *e.g.* Random Forest [69], require hundreds of thousands of training samples to avoids overfitting [19]. Comparatively, nonparametric methods, *e.g.* nearest neighbor, maximum correlation coefficient, *etc* work surprisingly well for one-shot-learning because they are intrinsically template matching metrics and will not suffer from overfitting problems.

Our approach: *Extended-MHI* and Maximum Correlation Coefficient

Motion Templates: *MEI* and *MHI* proposed by Davis and Bobick [70] are used to represent the motions of an object in video. All frames in a video sequence are projected onto one image across the temporal axis. As where and how motion happens are recorded in the images, MHI captures the temporal information of the motion in a sequence. Assume $I_t = (I_1, I_2, \ldots, I_{nFrames}) \in \Re^3$ is a gray scale image sequence and let $B_t = (B_1, B_2, \ldots, B_{nFrames-1}) \in \Re^3$ be

a binary image sequence indicating regions of motion, which can be obtained from image differencing and thresholding:

$$B_t = \begin{cases} 1 & \text{if } (I_{t+1} - I_t) > Threshold, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

where threshold is defined as:

$$Threshold = \sqrt{\sum_{t}^{nFrames} \sigma_t / (h \times w \times nFrames)} \tag{2.10}$$

where σ_t is the second moment (variance) of a single frame I_t; $h, w, nFrames$ are the height, width and frame number of that video sequence.

The *MHI* $H(t;\tau)$ is used to represent how the motion image is moving, and is obtained with a simple replacement and a decay operator:

$$H(t;\tau) = \begin{cases} \tau & \text{if } B_t = 1, \\ max(0, H((t-1);\tau) - 1) & \text{otherwise.} \end{cases} \tag{2.11}$$

We observe that the larger τ, the more information is encoded. Therefore, we set τ as the duration of the whole action T to preserve the whole sequence motion trail. The re-defined version of MHI is:

$$\widetilde{H}(t) = \begin{cases} T & \text{if } B_t = 1, \\ \widetilde{H}(t-1) - 1 & \text{otherwise.} \end{cases} \tag{2.12}$$

Note that there is no maximum operator in front of $\widetilde{H_\tau}$ *cf* Eq. (2.11) because setting τ as the sequence duration will lead to non-negativity of $\widetilde{H}(t;\tau)$.

We further extend motion templates that include two more elements: *GEI* and *INV*:
GEI is to compensate for the non-moving regions and the multiple-motion-instants regions of the action. The summation of all image pixels and normal-

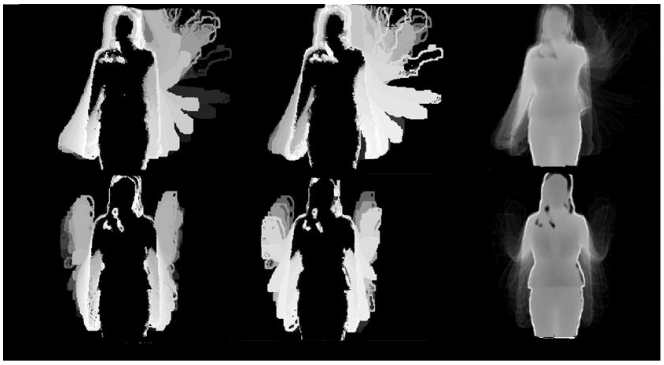


Figure 2.10: Illustration of the *MHI*, *INV* and *GEI* in two tokens (top row and bottom row). The projection images show that *MHI* emphasizes recent motion, *i.e.* ending frames whilst *INV* the beginning frames. *GEI* encodes the average gait information and is supplementary in repetitive actions where both *MHI* and *INV* are poor at representing.

ization of the pixel value define *GEI*:

$$G = \frac{1}{\tau}\sum_{t=1}^{\tau} I_t \tag{2.13}$$

INV is used to recover the loss of initial frames' action information when setting τ as the whole action duration and is defined as follows:

$$\widetilde{I}(t;\tau) = \begin{cases} \tau & \text{if } B_t = 1, \\ \widetilde{I}(t+1;\tau) - 1 & \text{otherwise.} \end{cases} \tag{2.14}$$

Note that its subtle difference to Eq. (2.12) is the time variable becomes $t+1$ instead of $t-1$ from which we get the name *Inversed Recording*.

We reason the complementary property of our proposed *extended-MHI* as *MHI* is poor at representing repetitive actions and *INV* provides complementary information by emphasizing (assigning larger value) at initial motion frames instead of the last motion frames. Figure 2.10 illustrates the similarities and differences between *MHI*,*GEI* and *INV* of two gesture tokens. The first columns are the *MHI* projections, second are the *INV* projections and the last are the *GEI* projections. Again, the projection graphs show that *MHI* emphasizes recent motion, *i.e.* ending frames whilst *INV* the opposite. Hence the combination of the two is complementary. Furthermore, *GEI* encodes the supplementary in-

Methods	*GEI*	*MHI*	*INV*	*Extended-MHI*
$\mathfrak{LD}$	0.2761	0.3010	0.3022	**0.2600**

Table 2.2: Performance comparison of three elements in *Extended-MHI*.

formation in repetitive actions where both *MHI* and *INV* are poor for the representation. Then, we reduce the dimensionality of each projection by dividing the projection into a 16×16 lattice using HOG as the feature descriptor and concatenate three vectors into a long feature vector. Supervised linear discriminant analysis (LDA) is adopted for the final stage of dimensionality reduction. The experimental results in Table 2.2 prove the viability of our conjecture. Note that in order to have a fairer comparison between different algorithms, we use the action boundaries provided by [53] for development batch instead of using the temporal segmentation results in Section 3.2 and *MSE* in Section 3.4 is also used for RGB and depth camera fusion so that irrelevant influences can be reduced to a minimum.

For the matching metric, nonparametric methods is more advantageous by avoiding the issue of overfitting. In our experiment, Maximum Correlation Coefficient works best. The correlation coefficient is defined as:

$$\rho = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \tag{2.15}$$

where σ_{xy} is the covariance of two feature vectors x and y, and σ_x, σ_y are the respective variances.

2.3.1.4 Multiview Spectral Embedding (*MSE*) for Data Fusion

To effectively and efficiently learn the complementary nature of different views, we adopt the spectral methods in [64] to search for a low dimensional representation and sufficiently smooth embedding over all views simultaneously. Luxburg [71] elegantly presents the intuition behind why spectral clustering works. We briefly state the core algorithm in *MSE* and further cast light on the unaddressed dimensionality problem in [64] by Graph Cut point of view. For notational details, please refer to the paper [64]. Firstly, we construct the graph

Laplacian L^i for each view i. The normalized graph Laplacians we choose for the system is L_{sys} as the matrix is symmetric. Then we introduce a weight α_i to encode the significance for each view i. We try to find the low-dimensional embedding by solving:

$$\underbrace{argmin}_{Y,\alpha} \sum_{i=1}^{m} \alpha_i^r tr(YL^iY^T) \tag{2.16}$$

$$\text{s.t. } YY^t = I; \quad \sum_{i}^{m} \alpha_i = 1, \quad \alpha_i \geq 0. \tag{2.17}$$

where Y is the multiview fused embedding feature vector in a dimension of d, exponent r is the coefficient for controlling the interdependency between different modalities/views and should satisfy $r \geq 1$. Pronounced independence between difference modalities prefers smaller r while rich complementary prefers larger r. In our system, the value r has trivial influence over low dimensional embedding and is set to be 1.5. In our system, we only fuse RGB and depth camera, hence the number of views m is 2.

Eq. (2.17) is a nonlinearly constrained nonconvex optimization problem and an expectation-maximization (EM) like iterative algorithm can be used to obtain a local optimal solution. The alternating optimization iteratively updates Y and α in an alternating fashion. By introducing Lagrange multiplier λ to take the constraint $\sum_i^m \alpha_i = 1$ into consideration, we get the Lagrange function

$$L(\alpha, \lambda) = \sum_{i=1}^{m} \alpha_i^r tr(YL^iY^T) - \lambda(\sum_{i}^{m} \alpha_i - 1) \tag{2.18}$$

By setting the derivative of $L(\alpha, \lambda)$ with respect to α_i and λ to zero, we have

$$\alpha_i = \frac{(1/tr(YL^iY^T))^{1/(r-1)}}{(\sum_{i=1}^{m} \alpha_i tr(YL^iY^T))^{1/(r-1)}} \tag{2.19}$$

Here, we cast light on the choice of lower embedding dimension d and the interpretation of weights α_i dispatched to different views where the original paper [64] fails to accomplish. In the paper of [64], the value of d is acquired by cross-validation. However, we argue that the low dimension d should be

fixed to be the number of gesture *class-1*. According to the Graph Cut theorem, the multiplicity k[1] of the eigenvalue 0 of Graph Laplacian L equals the number of connected components in the graph. Similarly, *MSE* finds d smallest eigenvalues in the spectrum of L which corresponds to the smallest variation of the cluster. The smallest eigenvalue of L is always 0 [71] and the corresponding eigenvector is the constant one vector *1*. Therefore, the veritable number of d should be the number of cluster/gesture *class-1*. And the experiments in [64] are in agreement with our reasoning. Secondly, we explicitly express the physical meaning of the weights α_i as a measurement of the "closeness" of intra-class distance from each individual view. From Eq. (2.19), we can see that α_i is proportional to the inverse trace of YL^iY^T, and

$$tr(YL^iY^T) = \sum \lambda_i \tag{2.20}$$

where λ_i are the eigenvalues of the Graph Laplacian L^i. Hence, $\alpha_i \propto 1/(\sum \lambda_i)$. In Spectral clustering [71], small eigenvalues (closer to 0) represent the the "closeness" of intra-class distance from each individual view. A well clustered view, *i.e.* , easier to be classified, is more significant than other views. So a larger α_i assigns larger significance to that view.

We then use the low dimensional multiview fused representation Y as the feature vector for Correlation Coefficient comparison. Note that this approach unsupervisedly clusters the test set, however it does not violate the competition rule that allows using unlabeled examples for training the system. We compare the performance between our approach against the approach which directly concatenates the RGB and depth camera feature vector and there is a consistently 4% improvement in $\mathfrak{LD}$.

2.3.1.5 Performance Evaluation

The performance of our system on validation data batch is **0.29685** and among the top entries on the public leader board in [53] with $\mathfrak{LD}$ less than 0.3. Figure 2.11 shows our system's performance on the first 20 development batches. It can be observed that our system performs well when there is large amount of

[1] multiplicities: the number of eigenvectors belonging to λ_i

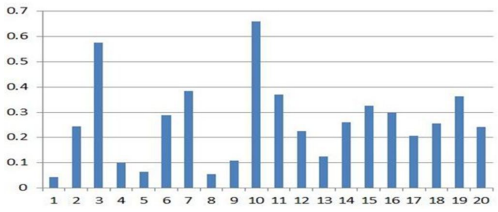


Figure 2.11: Performance on the 20 development data batches.

motion presents in a gesture token, *e.g.* batch $01, 05, 08, 09$ whereas the performance suffers if the gestures are rather static, *e.g.* batch $03, 10$. We reason that our gesture descriptor is motion based so that little motion and subtle appearance differences in gesture tokens will degenerate our system's discriminative power.

The experiments were done on a Intel 2-core $3.0GHz$, $4GB$ memory desktop in a single thread running MATLAB and the average training and testing time for a single batch is around 220 seconds (approximately 20 fps) which is faster than real time requirement.

2.4 Discussion

In this chapter, two new representations, namely *BoCP* and the *Extended-MHI* for action recognition were presented. *BoCP* is a temporally local feature descriptor and the *Extended-MHI* is a holistic motion descriptor. In the *BoCP* model, the unique way of considering temporal-structural correlations between consecutive human poses encodes more information than the traditional bag-of-features model. Also a soft-assignment strategy was utilized to preserve the visual word ambiguity which is usually disregarded during the quantization process after k-means clustering. The extension of *MHI* compensates for information loss in the original approach and later the conjecture that local and holistic features are complementary to each other was verified.

A one-shot-learning gesture recognition system was proposed to utilize both RGB and depth information from Kinect sensor. Depth sensor has the unique property to enable segmenting human silhouettes and a morphological was performed to denoise the depth images. Temporal segmentation was

performed on the appearance-based approach and an *extended-MHI* representation was adopted as the motion descriptor. The intrinsic property was explored between different spectra and made a physically meaningful embedding of multiviews through *Multiview Spectral Embedding*.

In the current work, the system shows promising performance and produces better results than any published paper on the *IXMAS* dataset using only low-level features and a simple dimensionality reduction method with the early fusion strategy. With more sophisticated feature descriptors and advanced dimensionality reduction methods, it's reckoned better performance should be achieved. Moreover, the utilization of the state-of-the-art skeleton tracker [19] could better assist the system to conquer its ineffectiveness in discriminating static gestures by relying on a more advanced appearance-based descriptor.

Chapter 3

Deep Belief Dynamic Networks

3.1 Introduction

This chapter presents a hierarchial dynamic framework that first extracts high level skeletal joints features and then uses the learned representation for estimating emission probability to infer action sequences. Currently gaussian mixture models (GMMs) and hidden Markov models (HMMs) are the typical pair and GMMs are the standard paradigms for modeling the emission distribution of HMMs. It can be shown that better action recognition using skeletal features can be accomplished by substituting GMMs by Deep Belief Networks (DBNs) that consist of many layers of features to predict probability distributions over states of hidden Markov models. The framework can be easily extended to include an ergodic state to segment and recognize actions simultaneously.

This chapter first provides a brief overview of Restricted Boltzmann Machines (RBMs), generalizations of RBMs to modeling real-valued data, and by stacking RBMs together as a pre-train procedure for DBNs, the better initialization can be achieved. After introducing the preliminaries, the deep belief dynamic networks are presented to model the higher level temporal data.

The model has been designed with action/gesture recognition in mind, but should lend itself well to other high-dimensional time series data.

3.2 Boltzmann machines

3.2.1 Energy-Based Model

Energy-based model assigns a scalar energy to corresponding composition of the variables associated to the model. The task of learning from the perspective of energy-based model can be explained as adjusting the energy function of the model so that its configuration has fitting traits. One fitting trait is that probable compositions to have low energy. Extending to the probabilistic graphical models, the energy-based model is characterized by a probability distribution in the form of an energy function:

$$p(x) = \frac{e^{-E(x)}}{Z} \tag{3.1}$$

where the normalizing factor Z is called the **partition function** termed by physical systems as the summation of all possible configurations of the variables:

$$Z = \sum_{x} e^{-E(x)} \tag{3.2}$$

Generally, the learning strategy for an energy-based model can be (stochastic) gradient descent with the cost as the observational negative-log-likelihood of the training instances. The log-likelihood and the corresponding cost (negative-log-likelihood) with parameter θ and input dimension D are defined as follows:

$$\mathscr{L}(\theta, D) = \frac{1}{N} \sum_{x^{(i)} \in D} \log p(x^{(i)}) \tag{3.3}$$

$$l(\theta, D) = -\mathscr{L}(\theta, D) \tag{3.4}$$

3.2.1.1 EBMs with Hidden Units

In order to boost the expressiveness of the model, or in many scenarios, the variables x are not fully observable, it's common to introduce a **hidden** part h into the energy-based model with the observed variables (still denote x).

Hence, the energy-based model with hidden units can be written as:

$$P(x) = \sum_h P(x,h) = \sum_h \frac{e^{-E(x)}}{Z} \tag{3.5}$$

The introduction of hidden unites h for image classification tasks can be seen as the higher level feature detectors and for action recognition can be seen as the higher level spatial temporal feature detectors.

Motivated by the field of physics, to unify aforementioned establishment analogous to Eq.3.1, the introduction of **free energy** could be served as the intermediate step for gradient calculation and is denoted as:

$$\mathcal{F}(x) = -\log \sum_h e^{-E(x,h)} \tag{3.6}$$

Hence, Eq.3.1 can be similarly defined as:

$$\mathcal{P} = \frac{e^{-\mathcal{F}(x)}}{Z} \tag{3.7}$$

with normalization constant as:

$$Z = \sum_x e^{-\mathcal{F}(x)} \tag{3.8}$$

Applying gradient descent method, a peculiarly intriguing gradient with respect to the negative log-likelihood of the data has the pattern:

$$-\frac{\partial \log \mathcal{P}(x)}{\partial \theta} = \frac{\partial \mathcal{F}(x)}{\partial \theta} - \sum_{\tilde{x}} \mathcal{P}(\tilde{x}) \frac{\partial \mathcal{F}(\tilde{x})}{\partial \theta} \tag{3.9}$$

The interesting property of the right hand side of the equation is explained as follows: the two terms are denoted as the **positive phase** and **negative phase**. The phrase positive and negative are not invoked by the sign of individual term, however mirror their outcomes on the configuration of the energy-based model. The **positive phase** (the first term) raise the probability of the observed data (by lowering the reciprocal free energy) whilst the **negative**

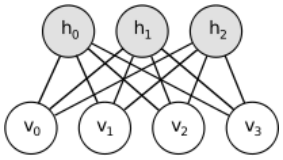


Figure 3.1: Restricted Boltzmann machines.

phase (the second term) lowers the probability of the instances spawned by the model.

Because the gradient of the second term can not be computed in a closed form ($E_p[\frac{\partial \mathcal{F}(x)}{\partial \theta}]$ has no analytical solution which is the expectation of all possible configurations of the observable x times their corresponding probability distribution P), approximation approach is required.

3.3 Restricted Boltzmann machines

Boltzmann Machines (BMs) are a special structure of Markov Random Field (MRF), *i.e.* the energy function is linear in term of its corresponding free parameters. To empower the expressiveness of the model so as to encode complex distributions, the hidden variables are introduced to enhance the modelling capability of the Boltzmann Machines

Restricted Boltzmann Machines (RBMs) is a subtype of BMs in that there is no connections between visible to visible or hidden to hidden variables. RBM, as a special type of Markov random field with a two-layer structure, has the visible binary stochastic units $v \in \{0,1\}^D$ connected to the hidden binary stochastic units $h \in \{0,1\}^F$. A graphical depiction of an RBM is show as in Fig3.1.

The energy of the state $\{v, h\}$is:

$$E(v,h;\theta) = -v^\top W h - b^\top v - a^\top h \tag{3.10}$$

$$= -\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j \tag{3.11}$$

where $\theta = \{W, b, a\}$ are the free parameters: $W_{i,j}$ serves as the symmetric syn-

ergy term between visible unit i and hidden unit j; b_i is the bias term of the visible units and a_j is the bias term of the hidden units. The joint distribution over the visible and hidden units is defined by:

$$P(v,h;\theta) = \frac{1}{Z(\theta)}\exp(-E(v,h;\theta)); \tag{3.12}$$

$$Z(\theta) = \sum_v \sum_h exp(-E(v,h;\theta)) \tag{3.13}$$

The conditional distributions needed for inference and generation are given by:

$$P(h_{j=1}|\mathbf{v}) = g(\sum_i W_{ij}v_i + a_j)); \tag{3.14}$$

$$P(v_{i=1}|\mathbf{h}) = g(\sum_j W_{ij}h_j + b_i)) \tag{3.15}$$

where $g(x) = \frac{1}{1+exp(-x)}$ is the logistic function.

The derivative of the log-likelihood with respect to the model parameter from Eq. 3.13 is expressed as: $E_{P_{data}}[vh^\top] - E_{P_{model}}[vh^\top]$ where E denotes the expectation. Due to the intractability of the second term, an approximation is generally required. This approximation is called the "Constrative Divergence":

$$\Delta W = \alpha(E_{P_{data}}[\mathbf{v}\mathbf{h}^T] - E_{P_T}[\mathbf{v}\mathbf{h}^T]). \tag{3.16}$$

where α is the learning rate and P_T is the distribution obtained by running a Gibbs chain, initialized with the visible units given by the data, for T full steps.

3.3.1 Gaussian Bernoulli Restricted Boltzmann machines

If input features (*a.k.a.* observation domain $\mathcal{X}$) are continuous instead of binomial features, we use the Gaussian RBM (*GRBM*) to model the energy term of first visible layer:

$$E(v,h;\theta) = -\sum_{i=1}^{D}\frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij}h_j\frac{v_i}{\sigma_i} - \sum_{j=1}^{F} a_j h_j \tag{3.17}$$

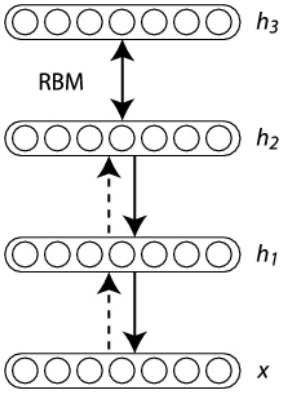


Figure 3.2: Deep Belief Networks.

The conditional distributions needed for inference and generation are given by:

$$P(h_{j=1}|\mathbf{v}) = g(\sum_i W_{ij}v_i + a_j)); \tag{3.18}$$

$$P(v_{i=1}|\mathbf{h}) = \mathcal{N}(v_i|\mu_i, \sigma_i^2). \tag{3.19}$$

where $\mu_i = b_i + \sigma_i^2 \sum_j W_{ij}h_j$ and $\mathcal{N}$ is normal distribution. In general, we normalize the data by mean subtraction and standard deviation division in the preprocessing phase. This type of structure sometimes is called a mean covariance RBM (mcRBM) in some literature. Hence, in practice, instead of learning the σ_i^2, it's commonly to adopt a fixed, predetermined unit value ***1*** for σ_i^2.

The aforementioned graphical model results in the following free energy representation for RBM and Gaussian RBM:

$$\mathcal{F}(v) = -b^T v - \sum_i \log \sum_{h_i} e^{h_i(a+v^T W)}$$

$$\mathcal{F}(v) = -\frac{(v-b)^2}{2} - \sum_i \log \sum_{h_i} e^{h_i(a+v^T W)}$$

3.4 Deep Belief Networks

In the seminal paper of [72], Hinton and Salakhutdinov showed that by stacking multiple layers of *RBMs* and pre-training in a greedy "Constrative Divergence" manner, *Deep Belief Networks (DBN)* in the form of multi-layer perceptron can be better initialized. A graphical depiction of an RBM is shown in Fig 3.2.

The joint probability distribution between observed vector x and the l hidden layers h^k can be written as:

$$P(x, h^1, \ldots, h^l) = (\prod_{k=0}^{l-2} P(h^k|h^{k+1}))P(h^{l-1}, h^l) \tag{3.20}$$

Justifying Greedy-Layer Wise Pre-Training: Neal and Hinton [73] demonstrated that the negative log-probability of an observed input vector, $\mathbf{v}^0$, in the multi-layer perceptron framework, is bounded by a variational lower bound. This lower bound, in the form of free energy, is the summation of two terms: 1) expected energy under the approximating distribution, $Q(\mathbf{h}^0|\mathbf{v}^0)$ and 2) the negative entropy of the corresponding distribution. Using RBMs as the building block of DBNs, the architecture is a directed graphical model. And the "energy" of the configuration $\mathbf{v}^0$ and $\mathbf{h}^0$ is expressed as $E(\mathbf{v}^0, \mathbf{h}^0) = -[\log p(\mathbf{h}^0) + \log p(\mathbf{v}^0|\mathbf{h}^0)]$. Hence, the variational lower bound is:

$$\begin{aligned}\log p(\mathbf{v}^0) \geqslant & \sum_{\mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0)[\log p(\mathbf{h}^0) + \log p(\mathbf{v}^0|\mathbf{h}^0)] \\ & - \sum_{\mathbf{h}^0} Q(\mathbf{h}^0|\mathbf{v}^0) \log Q(\mathbf{h}^0|\mathbf{v}^0)\end{aligned}$$

Bengio et al. [74] further demonstrated that deep architectures have the merits over shallow architectures in terms of model expressiveness and efficiency. This exponential efficiency which is required to represent energy functions stands out as the major contribution of the deep architecture. Moreover, with the greed layer-wise unsupervised pre-training, the weights of the model will be better initialized in a region in the vicinity of a good local optimum. This strategy helps the optimization and generalization, giving rise to energy

function that are high-level abstractions of the lower layers.

Stochastic Gradient Methods: If the training data mainly comprise of a large quantity of *iid* samples, which is mostly true for the large scale image classification or spatio-temporal features this thesis mainly interested with. Rather than sweeping through all the training samples to have the gradient update, it is be more desirable to update the gradient after observing only a few number of samples. *Stochastic Gradient Descent (SGD)* is a commonly adopted optimization strategy to exploit the large training set with *iid* samples. The key concept is that for each iteration, to choose a set of training samples at random, and move a small stride along the direction indicated by the gradient. In the batch update setup, *SGD* is usually a suboptimal optimization strategy, because the global optimum could be in a very contrasting direction than the direction obtain by the local steepest descent. Hence, *SGD* is generally associated with the tradeoff: global methods such as *L-BFGS* could give rise to better direction than the individual step, however, the locally generated direction by *SGD* can be computed much faster.

The standard backpropagation can be adopted for adjusting the weight W:

$$W = W - \alpha_m \frac{\partial}{\partial W} J(W) \tag{3.21}$$

where $\alpha_m > 0$ is the annealed learning rate for epoch m that controls how fast the parameters move to the direction of the gradient and $J(W)$ is the cost function (cross-entropy) of the last layer perceptron. This step size α_m is one crucial hyper-parameter for *DBN*: if it is too large, the update of the parameters will swing violently and may not converge at al, and contrarily, if too small, the training process will progress in a much slower manner and in the case of some extreme cases, the numerical values of the gradient may signal the optimization process has converged whereas the local optimal is far from the erroneously stopping point.

The main issue in gradient descent is: how should we set the step size α_m? This proves to be a difficult task. If the learning rate is constant and relatively small, convergence will be very slow, however if it is large, the method might fail to converge at all. This point is illustrated in Fig 3.3 by the following (con-

vex) function:

$$f(\boldsymbol{\theta}) = (\theta_1^2 - \theta_2)^2 + (\theta_1 - 1)^2, \tag{3.22}$$

with initial guess$(0, 0)$.

In Fig 3.3, the step size η is fixed. For fixed step $\eta = 0.1$, note the path of the descent moves slowly along the valley. For fixed step size $\eta = 0.3$, the algorithm starts oscillating along the sides of the valley and never converges to the optimum even though the learning rate difference is smaller than an order of magnitude.

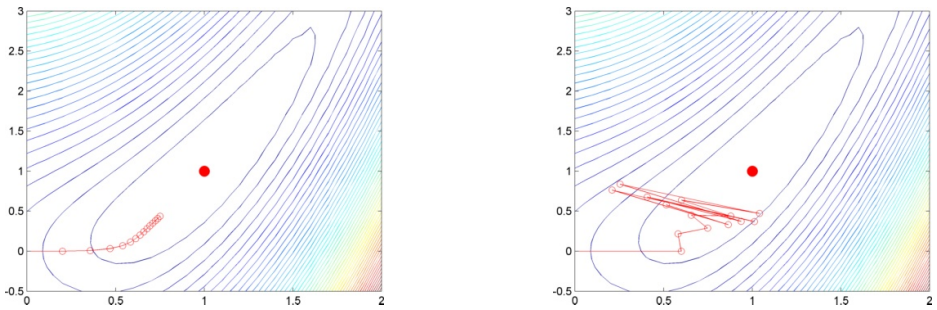


Figure 3.3: Gradient descent on a simple convex function, starting point is $(0, 0)$, for 20 steps, using a fixed learning rate (step size) η. The global minimum is at $(1, 1)$. (a) $\eta = 0.1$. (b) $\eta = 0.6$.

In general, α_m should decrease as epoch m increases. In the hope that the optimization process will converge to a local optimum. The typical approach is to select a strategy in the form of $\alpha_m \sim \frac{1}{m}$ or $\alpha_m \sim \frac{1}{\sqrt{m}}$. However, simply taking $\alpha_m = \frac{1}{m}$ will result in too large step size for the first few update. Alternatively, a "common trick" is demonstrated as in[75]

$$\alpha_m = \frac{1}{\sigma^2(m_0 + m)} \tag{3.23}$$

or

$$\alpha_m = \frac{m_0}{(1 + \sqrt{m})} \tag{3.24}$$

where m_0 is a a hyper-parameter that is required to be initialised. A typical proposition for choosing this parameter is to run several epoch update over a subset of the training examples with different size α. The optimum m_0 is chosen such that $\alpha_0 = \alpha^*$ with the highest resulting likelihood.

Historically, back-propagation algorithms in multi-layer deep neural networks have been associated with*SGD*. Because the non-convexity of the objective function in multi-layer deep neural networks, SGD are infamously difficult to debug. The algorithm may seemingly working in spite of incorrect gradient. Giving rise to the misbelief of the flawed system. The major drawback of *SGD* is its hyper-parameter tuning, contrary to the off-the-shelf solvers as *L-BFGS* or conjugate gradient. It is also worth pointing out that *SGD* is not favourable in the setups where training samples are not *i.i.d.*, or on small data sets. Nevertheless, *SGD* is still the dominant method for updating gradient for deep neural networks.

3.5 Deep Belief Dynamic Networks

Inspired by the state-of-the-art performance achieved for speech recognition [17], the proposed model borrows the idea of a data driven learning system, relying on a "pure learning method that all the information gain in the model derives from the data without sophisticated pre-processing or dimensionality reduction. The proposed framework can be seen as an extension to [76] in that instead of using the conditional Restricted Boltzmann Machine, a type of shallow model, to model human motion, we add layers to learn higher level features justified by a variational bound [11]; for modeling temporal information, rather than explicitly binding limited adjacent frames (3 past frames as in [76], we resort to hidden Markov model which can be well extended to long term temporal information, spanning hundreds of frames in our system.

We demonstrate that consistently better action recognition performance can be achieved using skeleton information by "pretraining" a multi-layer neural network in which the greedy pre-training is done by "contrastive divergence" method used for RBMs as generative models. This pre-training better initiates the weights of deep neural networks that have multiple layers of hidden units and exponentially increases the capacity of the model before overfitting. The generative pre-training generates many layers of feature detectors and feature descriptors that grow into progressively more abstract and complex. An ensuing process of discriminative fine-tuning by the standardized backprop-

agation to slightly modify the feature detectors and descriptors in each layer rendering them more effective for the discriminative tasks. The advantage of pre-training multi-layer deep neural nets is that the limited volume of knowledge from the labels is not devoted to design features detectors and descriptors from scratch. It is particularly devoted to adjust the weights of the deep neural nets so slightly that the better class hyperplane could be obtained. The features detectors and descriptors themselves are designed by the pre-training process as a generative model.

The Deep Belief Dynamic Networks make three major presumptions about the innate of the relationship between the input data, which in this chapter is a set configurations of skeletal joints, and the label information, which are action class HMM hidden states produced by a forced alignment. The first assumption is that the discriminative tasks is more relevant to the underlying relationship of the data rather than to the singular elements of the data itself. Previous hardwired techniques [77, 78, 79, 80] have shown that multiple joints relational features, *e.g.* hands approaching each other, feet moving towards each other, *etc* , are more relevant for action recognition rather than a single joint spatial-temporal position). The second assumption is that the aforementioned underlying relationship of the data can be presented by modeling its high-order statistical model. Third, feature-vector produced hidden states are mostly unique, meaning sequences are non-repetitive actions as opposed to longer repetitive activities, *e.g.* walking, running, jogging, *etc* , spanning minutes or hours.

Given the structure of our model, our framework is also suited for detection of "action points" [79] for accurate temporal localizing of gestures. Action point could be perceived as spotting a particular pose under the hypothesis of in what ways the performer fixate into that pose. This evaluation metrics explicitly lay out the tradeoff between latency and accuracy.

3.5.1 Methodology

3D joint data obtained through the skeletal detection from the depth images are generally more noisy than the 3D joints generated from the MoCap data. It is

often the case that the disparity between gestures is subtle, hence the accurate determination of hidden states from the observation without attentive design of features is generally challenging. The suboptimal design of the relevant features undermines the performance of such generative models and mostly serve as the bottleneck of the whole system. Furthermore, without sufficient number of training examples, the learning of a big, complex generative model is prone to overfitting. As it's pointed out by [17] the interesting relationship between the amount of constraint that the data imposes on the discriminative model and generative model: for the discriminative model, its parameters is equal to the number of bits mandatory to define the factual labels of the training data; for the generative model, the constraint is equal to the number of bits mandatory to define the input space vector. Therefore, when input space vector has much more structure information than the information from the labels which is mostly the case, a generative model is able to have many more parameters before overfitting occurs. "

Currently the model parameters are predominantly learnt by Gaussian mixture models using expectation maximization [81, 82]. We reason that replacing Gaussian mixture models by deep belief networks can better predict probability distributions over the states of hidden Markov models:

3.5.1.1 Problem formation

The seminal works of [17] concludes that "*DBN* provide several potential advantages over GMMs:

- The data distribution is not a precondition for *DBN* to estimate the posterior probabilities of HMM hidden states.
- *DBN* renders it possible to incorporate diverse input features types, either discrete or continuous variables.
- *DBN* requires far more amount of training data to adjust each parameter because the susceptibility of the output for each training instance to a broad portion of the weights.

The major advantage of *DBN* over *GMM* is in the benefit of each weight in *DBN* being learned by a broad portion of training instances. Multi-layer perceptron have traditionally been trained discriminatively, in contrast, *GMMs* are generally trained as generative models, albeit later stage of discriminative training process. Generative models render many more bits of constraints on the parameters imposed by the data. Therefore, moderately compensating for the fact that each module of a large-size *GMM* can only be trained on a very small portion of the data.

GMMs and HMMs co-evolved for speech recognition for the past decade when computational resources was a major constraints to machine learning community to explore larger, more complex architectures. GMMs are straightforward to learn with diagonal covariance matrices, and with sufficient components, GMMs is able to model any distribution. However, GMMs are statistically less efficient in modeling high-dimensional data with complex componential structure [17]. An analytical illustration is stated as follows: for the input data with $\mathcal{N}$ major distinct configurations and in each configuration, there are $\mathcal{M}$ major distinct sub-band configurations. Assume that each configuration in each sub-band is roughly independent. A GMM demands $\mathcal{N} \times \mathcal{M}$ components to represent the configuration since each component should represent both sub-band configurations. In another case when a model can represent the data via multiple share structures only requires $\mathcal{N} + \mathcal{M}$ components with each sub-band configuration has shared upper level structure. Aforementioned disadvantage result in the GMMs+HMMs system with a large number Gaussian components and each component is learnt from a small portion of the data.

With a number of new datasets [20, 21, 22, 23] and the large supply of unlabeled skeletal data, the benefit of learning a generative model is notably magnified. Even though no unlabeled data are used in the following experiments, we believe the use of unlabeled data for better model initialization could only further improve the results relative to solely discriminative methods.

It is safe to assume that many of the high-level features detected, extracted from the generative model may be unrelated to the task specific discriminative jobs. Nevertheless, the pre-training process are critical for interpreting

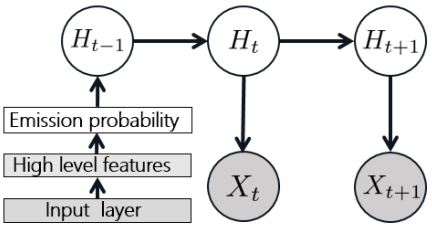


Figure 3.4: Per-action model: a forward-linked chain. Inputs (skeletal features) are first passed through a deep neural nets to extract high level features and output the emission probabilities of the hidden states. The deep neural net is first pre-trained using all skeletal features and then fine-tuned by the target class acquired by forced alignment for individual action class (10 hidden states for each action class in all our experiments).

the input data. Moreover, with gradient diminish effect, how to initialize the network properly is the key contribution and the renaissance of Deep Neural Network since 2006 [72]. Hence, with the increase variational lower bound guaranteed (see Sec.3.4), the pre-training process is a procedure worth implementing especially if the computational cost is inexpensive and the extracted high-level features are favourable for discriminating tasks.

3.5.2 Graphical Models

In order to encode higher level temporal relationships, a continuous-observation HMM is utilized with with discrete hidden states. The temporal model is constructed the same as [79]: for each time step t, there is a random observation variable variable X_t and an unobserved variable H_t in a finite set $\mathcal{H} = (\bigcup_{a\in\mathcal{A}} \mathcal{H}_a)$,where $\mathcal{H}_a$ is a group of sub-states assigned to an individual class of action a. The proposed model is motivated by the flexibility requirement for capturing the forward transitions of the chain, so that the variability of each gesture sequence is satisfied. The probabilistic graphical model is defined as an HMM:

$$p(H_{1:T}, X_{1:T}) = p(H_1)p(X_1|H_1)\prod_{t=2}^{T} p(X_t|H_t)p(H_t|H_{t-1}), \tag{3.25}$$

where $p(H_1)$ is the prior on the first hidden state and in all our experiments in this chapter, we defined the prior with a uniform distribution. $p(X_t|H_t)$ is the observation model, and $p(H_t|H_{t-1})$ is the transition dynamics model. The observation domain $\mathcal{X}$ relies on the intrinsic property of the skeleton and will be specified in the Sec. 3.7. We model the emission distribution of hidden Markov models by first pre-training a deep belief networks, greedy layer-wise training as a generative model for a window of action frames. The graphical representation of a per-action model is shown as Fig. 3.4.

Because our skeleton features (*a.k.a.*observation domain $\mathcal{X}$) are continuous instead of binomial features, we use the *GRBM* to model the energy term of first visible layer as in Eq 3.17.

Model reasoning: The intuition using deep belief networks for modeling marginal distribution in skeleton joints action recognition is that by constructing multi-layer networks, semantically meaningful high level features for skeleton configuration will be extracted whilst learning the parametric prior of human pose from mass pool of skeleton joints data. In the recent work of [83], a non-parametric bayesian network is adopted for human pose prior estimation, whereas in our framework, the parametric networks are incorporated.

Using the pair wise joints features as raw input, the data-driven approach network will be able to extract relational multi-joints features which are relevant to target frame class. E.g., for "toss" action, wrist joints is rotating around shoulder joints would be extracted from the backpropagation via target frame as those task specific, *ad hoc* hard wired sets of joints configurations as in [77, 78, 79, 80].

The outputs of the neural net are the hidden states learned by force alignment during the supervised training process. We can infer the action presence in a new sequence by Viterbi decoding as:

$$V_{t,\mathcal{H}} = P(H_t|X_t) + \log(\max_{\mathcal{H}\in\mathcal{H}_a}(V_{t-1,\mathcal{H}})) \tag{3.26}$$

where initial state $V_{1,\mathcal{H}} = \log(P(H_1|X_1))$. The inference results are the shortest path with the highest probability, and the predicted action probability is

specified as $a \in \mathcal{A}$ as $p(y_t = a|x_{1:t}) = V_{T,\mathcal{H}}$.

The overall algorithm for training and testing are presented in the following Algorithm.1 and 2.

Algorithm 1: Deep Belief Dynamic Networks – training pipeline

Data:
$\mathbf{X} = \{\mathbf{x_i}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input feature (skeletal) sequence.
$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - frame based local label (achieved by semi-supervised forced-aligment),
where $\mathbf{y_i} \in \{C * S + \boldsymbol{1}\}$ with C is the number of class, S is the number of hidden states for each class, $\boldsymbol{1}$ as ergodic state.

1 Preprocessing the data $\mathbf{X}$ (with or without time window) as in Eq.5.2.
2 Normalizing(zero mean, unit variance per dimension) the above features and feed to to Eq.3.17.
3 Pre-training the multi-layer networks using *Contrastive Divergence* as in Eq.3.16.
4 Supervised fine-tuning the deep belief networks using $\mathbf{Y}$ by standard mini-batch SGD backpropagation 3.21.

Result:
GDBN - a gaussian bernoulli visible layer deep belief networks to generate the emission probabilities for hidden markov model.
$\mathbf{p}(\mathbf{H_1})$ - prior probability for $\mathbf{Y}$.
$\mathbf{p}(\mathbf{H_t}|\mathbf{H_{t-1}})$ - transition probability for $\mathbf{Y}$, enforcing the beginning and ending of a sequence can only start from the first or the last state.

3.5.3 *ES-HMM*: Simultaneous Segmentation and Recognition

The aforementioned framework can be easily adapted for simultaneous action segmentation and recognition by adding an ergodic states-$\mathcal{ES}$ which resembles the silence state for speech recognition. Hence, the unobserved variable H_t takes an extra finite set $\mathcal{H} = (\bigcup_{a\in\mathcal{A}} \mathcal{H}_a) \bigcup \mathcal{ES}$, where $\mathcal{ES}$ is the ergodic state as the resting position between actions and we refer the model as *ES-HMM*.

Since our goal is to capture the variation in speed of performing gestures, we set the transitions in the following way: when being in a particular node n in time t, moving to time $t+1$ we can either stay in the same node (slower performance), move to node $n+1$ (the same speed of performance), or move

Algorithm 2: Deep Belief Dynamic Networks – testing pipeline

Data:
$\mathbf{X} = \{\mathbf{x_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - raw input feature sequence
GDBN - the trained gaussian bernoulli deep belief networks to generate the emission probabilities for hidden markov model
$\mathbf{p}(\mathbf{H_1})$ - prior probability for $\mathbf{Y}$
$\mathbf{p}(\mathbf{H_t}|\mathbf{H_{t-1}})$ - transition probability for $\mathbf{Y}$

1 Preprocessing and normalizing the data $\mathbf{X}$ as in Eq.5.2
2 Feedforwarding network **GDBN** to generate the emission probability $\mathbf{p}(\mathbf{X_t}|\mathbf{H_t})$ in Eq.3.25
3 Generating the score probability matrix $\mathbf{p}(\mathbf{H_{1:T}}, \mathbf{X_{1:T}})$ according to Eq.3.25
4 Finding the best path $\mathbf{V_{t,\mathcal{H}}}$ by Viterbi decoding as in Eq.3.26

Result:
$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - frame based local label where $\mathbf{y_i} \in \{C * S + \boldsymbol{1}\}$ with C is the number of class, S is the number of hidden states for each class, $\boldsymbol{1}$ as ergodic state
$\mathbf{C}$ - global label, the anchor point is chosen as the middle state frame

to node $n + 2$ (faster performance). From the $\mathcal{ES}$ we can move to the first three nodes of any video, and from the last three nodes of every video we can move to the $\mathcal{ES}$ as shown in Fig. 3.5. The *ES-HMM* framework differs from the Firing Hidden Markov Model of [79] in that we strictly follow the temporal independent assumption, forbidding inter-states transverse, preconditioned that a non-repetitive sequence would maintain its unique states throughout its performing cycle.

The emission probability of the trained model is represented as a matrix of size $N_{\mathcal{TC}} \times N_{\mathcal{F}}$ where $N_{\mathcal{F}}$ is the number of frames in a test sequence and output target class $N_{\mathcal{TC}} = N_{\mathcal{A}} \times N_{\mathcal{H}_a} + 1$ where $N_{\mathcal{A}}$ is the number of action class and $N_{\mathcal{H}_a}$ is the number of states assigned to an individual action a and one $\mathcal{ES}$ state. Result of the Viterbi algorithm is a path–sequence of nodes which corresponds to states. From this path we can infer the class of the gesture.

Performance Measure: F-score@Δ The performance of the system is mea-

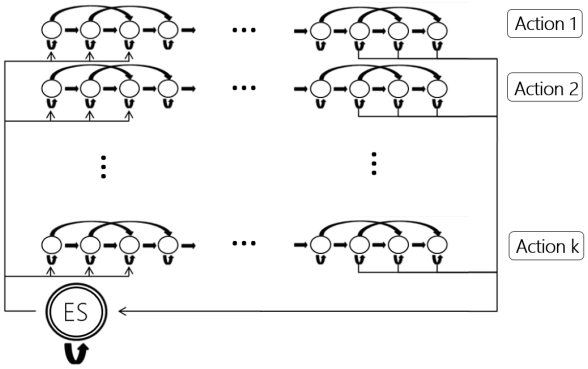


Figure 3.5: State diagram of the *ES-HMM* model for action segmentation and recognition. An ergodic states (ES) shows the resting position between action sequence. Each node represents a single frame and each row represents a single action model. The arrows indicate possible transitions between states.

sured in terms of precision and recall, defined in [21] as:

$$F - score(\Delta) = 2\frac{prec(\Delta) * rec(\Delta)}{prec(\Delta) + rec(\Delta)}$$

To achieve a high precision, the training data should only contain movements that users of the deployed system will associate with the gesture. To achieve a high recall, the training data should contain all movements that the designer wants to associate with a gesture. We assess the quality of our predictions using ground truth annotations, defining a performance measure as following 3 factors: 1) precision - how precise the system predicts the true positives against all the predictions; 2) recall - how accurate the system retrieves the performed gestures; and 3) latency - how small is the margin between the prediction by the system and the true action point. For a fixed margin of tolerated latency (Δms) we measure the precision and recall as shown in Fig. 3.6. A balanced F-score between 0 and 1 combines precision and recall. In the experiments we will examine the performance measure for a fixed $\delta = 333ms$ as in [21].

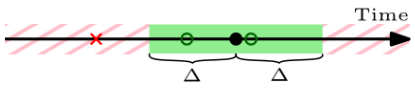


Figure 3.6: Anchor point definition according to [79]: The ground truth action point is marked as •, a pre-fixed windows of size 2Δ is centred around the ground truth annotation. There are three predictions, two correct marked in ○ as the True Positives (*TP*) and one incorrect marked in × as the False Positive (*FP*). However, there are two correct predictions within the action point window and in this case, only one correct prediction is valid and the rest are ignored. The $prec(\Delta)$ and $rec(\Delta)$ are determined by the total number of valid correct and incorrect predictions.

3.6 Related Works

Traditionally, 3D joints data are acquire by MoCap system, and a plethora of hard wired features have been proposed: by Müller *et al.* [78] introduced the relational pose features to index and retrieve motion capture data and served as the harbinger for exploring 3D joints data. Yao *et al.* [84] modified a subset of the relational pose features for action recognition and showed that by using these robust features, some imperfect poses can also suffice to perform action recognition. Lv *et al.* [85] designed feature vectors, such that each joint feature comprises of a single joint or combination of multiple joints. 7 distinct categories of hand crafted features were designed according to the analysis of the best ad-hoc combinatorial joints features distinguish different actions. Various hand-designed features describe different levels of dynamics of an action and there are in total 141 hard wired features. During their training stage, 3 sets of *ad hoc* segregation of features space required laborious human involvement. Ofli *et al.* [80] proposed the Sequence of Most Informative Joints (SMIJ) representation, an interpretive feature for human motion representation based on joint angle time series. Chaudhry *et al.* [77] introduced a bio-inspired features incorporating 3D shape context into a spherical coordinate system, they model a human activity using a hierarchy of 3D skeletal features in motion and learn the dynamics of these features using Linear Dynamical Systems (LDSs).

Alternative approaches to acquire discriminative features leverage statistical learning methods: Wang *et al.* [23] proposed a feature mining approach for

computing discriminative actionlets from a recursively defined temporal pyramid of joint configurations. Do *et al.* [86] proposed non-linear Markov model for structured prediction. Traditional conditional random fields uses shallow log-linear model for feature extraction. Their proposed framework combines the power of deep neural networks to extract high level features and leverages Markov networks for upper level structure prediction, creating a scalable and powerful probabilistic graphical model for signal labeling tasks.

Nowozin and Shotton [79] explicitly address the latency issue in action recognition tasks. In their proposed system, a single HMM comprised of multiple sub-models for each gesture is learnt by a random forest. During test-time action recognition, online filtering is adopted and the model state is reset by heuristics after per-recognized gesture.

Inspired by recent findings of [17], the automatic extraction of high level skeletal joints representation is proposed by using deep forward neural networks. This framework serves as a better model for estimation emission probability of hidden Markov models and achieves improved results for human action recognition amongst other well established methods. We also demonstrate that the framework can be easily adapted for simultaneous segmenting and recognizing gestures, discovering action points [79] which are precise temporal anchor points relative to the action performance. The proposed framework has been designed with action/gesture recognition in mind, but should lend itself well to other high-dimensional time series prediction tasks.

3.7 Experimental Results

Experimental results on three publicly available skeleton datasets are presented: the ChaLearn Italian Gesture Recognition, the MSR Action3D and the MSRC12 dataset. We first present the pre-processed skeletal features used in the experiments:

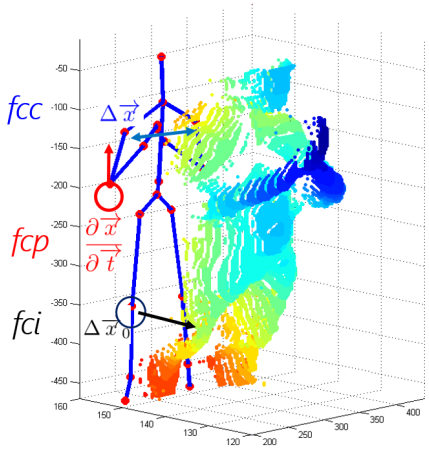


Figure 3.7: Point cloud projection of depth image from ChaLearn Italian Gesture dataset and the 3D positional features.

Features

The 3D coordinates of N joints of current frame c are given as: $X_c = \{x_1^c, x_2^c, \ldots, x_N^c\}$. We deploy 3D positional pairwise differences of joints [87] for observation domain $\mathcal{X}$. They capture posture features, motion features and offset features by direction concatenation: $\mathcal{X} = [f_{cc}, f_{cp}, f_{ci}]$ as demonstrated in Fig. 3.7.

$$f_{cc} = \{x_i^c - x_j^c | i, j = 1, 2, \ldots, N; i \neq j\} \tag{3.27}$$

$$f_{cp} = \{x_i^c - x_j^p | x_i^c \in X_c; x_j^p \in X_p\} \tag{3.28}$$

$$f_{ci} = \{x_i^c - x_j^I | x_i^c \in X_c; x_j^I \in X_I\} \tag{3.29}$$

Resulting in a raw dimension of $N_{\mathcal{X}} = N_{joints} \times (N_{joints} - 1)/2 + N_{joints}^2 + N_{joints}^2) \times 3$ where N_{joints} is the number of joints used. Note that before extracting any features, all the 3D joint coordinates are transformed from the world coordinate system to a person centric coordinate system by placing the HipCenter (or ShoulderCenter if applied) at the origin. By including temporal differences f_{cp}, f_{ci} partially overcomes the conditional independence requirement of HMMs, *i.e.* continuous frames are independent to previous frames given the current hidden state.

Admittedly, we do not completely negate human prior knowledge about

information extraction for relevant static posture, velocity and offset overall dynamics of motion data. Nevertheless, aforementioned three attributes are all very crude pairwise features without any tweak into the data set or hand-pick the most relevant pairwise, triple wise, *etc* , designed features [77, 78, 79, 80, 84, 85]. Similar data driven approach has been adopted in [21] where random forest classifiers were adapted to the problem of recognizing gestures using a bundle of 35 frames. These sets of features extraction processes resemble the *Mel Frequency Cepstral Coefficients (MFCCs)* for speech recognition community [17].

Experimental setup

For high level skeleton feature extraction, we fix network architecture as $[N_{\mathcal{X}}, N_2, 1000, 1000, 1000, 1000, N_{\mathcal{TC}}]$ where $N_{\mathcal{X}}$ is the observation domain dimension and N_2 is the number of hidden nodes in *GRBM*, depending on the used joints set and is chosen as 2000 for upper body joints and 4000 for full body skeletal joints; $N_{\mathcal{TC}}$ is the output target class. And in all our experiments number of states associated to the respective action $N_{\mathcal{H}_a}$ is chosen as 10 for modeling the states of an action class. The feed forward networks are pre-trained with a fixed learning rate using stochastic gradient decent with a mini-batch size of 100. Unsupervised layer-wise pre-training can help better initialize the relevant higher level feature detectors and we have run 100 epochs for unsupervised pre-training. For Gaussian-binary RBMs, learning rate is fixed at 0.001 while for binary-binary RBMs as 0.01. For fine-tuning, the learning rate starts at 0.1 with 0.998 scaling after each epoch. To prevent complex co-adaptations problems where a feature detector is strongly dependent in the context of several other co-evolved feature detectors, we dropout [88] half of the feature detectors. Though we believe further carefully fine-tuned parameters would lead to more competitive results, in order not to "creeping overfitting", as algorithms over time turn into too dataset-specific, virtually memorizing all the idiosyncrasies from the dataset, and eventually fail to generalize [89], we would like to treat the model as the aforementioned more generic approach.

Method \ Data Set	ChaLearn Gesture	MSR Action3D
EigenJoints+NBNN [87]	0.593	0.720
GMM+HMM [82]	0.408	0.704
NN+DTW [90]	0.599	-
DBN+HMM (this work)	***0.628***	***0.735***

Table 3.1: Baseline comparisons: first row (EigenJoints+NBNN) adopts same sets of features: showing that our model's efficacy in temporal incorporation; second row method (GMM+HMM) has the same graphical representation except that the Deep Belief Network is used to extract high level skeletal features, proving DBN is more effective for estimating the emission probability. And our model achieves better recognition rate than the winner of the challenge [90] that uses variant of nearest neighbour and dynamic time warping in the ChaLearn Gesture dataset.

Baseline

We perform the sanity check for our algorithm as an effective way of comparing against two baselines: in order to verify that the model is a more powerful alternative to GMM for relating HMM states, we compare our approach against the *GMM+HMM* paradigm [82] for modeling the observation states $p(X_t|H_t)$; to verify that the temporal incorporation in our model is a more effective approach for action recognition against the Bag-of-Visual-Word approach, we compare against the *EigenJoint-Naive Bayes Nearest Neighbour* [87] where the same set of raw features have been used.

3.7.1 ChaLearn Italian Gesture Recognition-Kaggle track

This dataset is on "multiple instance, user independent learning" [20] of gestures. The details of the dataset can be found at 6.1.1.1. We focus on the skeletal modality.

We use the subset where the label data are provided during our evaluation process. The set contains 393 labeled sequences with a total of 7754 gestures. We used 350 sequences for training and the rest 43 sequences for testing, each sequence contains 20 unique gestures. For the training set, there are in total 339,700 frames (20 fps). Note that large number of frames (up to hun-

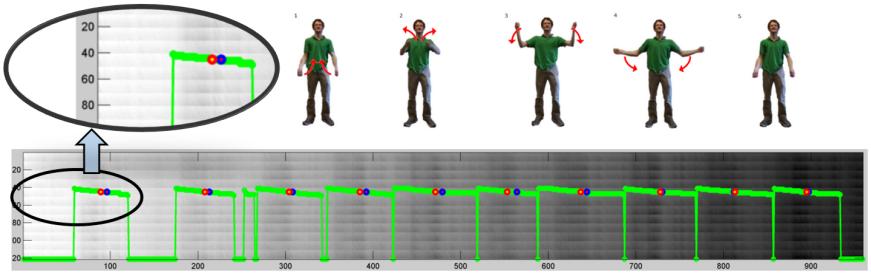


Figure 3.8: Top : a "Wind up the music" metaphoric action instance from MSRC12 dataset. Bottom: global score matrix via accumulating emission probabilities - fluorescent green is the Viterbi path from back tracking, a zoom in (top left) shows a path from states 41-50 (y-axis) indicating the gesture number 5 because we assign 10 states for each hidden Markov Model. Blue circles are the oracle ground truth action points and red circles are the predicted action points (middle frame of the Viterbi best path).

dred thousands of frames) is advantageous in our model settings over other nonparametric models for estimating skeletal human poses (*e.g.* GPLVM [91], Kernel Methods [92] could not be readily scaled up). Due to the parametric structure, once the training set is learned, testing time will be trivial compared with memory based method [87, 93]. Because this is a gesture recognition data set, only upper body joints are relevant to our discriminative tasks. Therefor, we consider only the upper 9 body for our task (full body joints have been compared, but as expected, results in inferior results compared to upper body 9 joints). The 9 upper body joints used are *"ShoulderCenter, ShoulderLeft, ElbowLeft, WristLeft, HandLeft, ShoulderRight, ElbowRight, WristRight, HandRight"*. The consistent improvement of recognition accuracy against two baseline methods under the same experimental settings in Table 3.1 shows the efficacy of the proposed framework in better estimating observation model and parsing temporal domain knowledge.

3.7.2 MSR Action3D

MSR Action3D dataset [23] is an action dataset of depth sequences captured by Kinect. This dataset includes twenty actions, each action was performed by ten subjects for three times with details at 6.1.3. We compared the methods using only skeleton joints module in Table 3.1. Though the model still consistently

Method	Classification rate
Sequence of Most Informative Joints [80]	0.29
Recurrent Neural Network [94]	0.425
Dynamic Temporal Warping [95]	0.54
Multiple Instance Learning [96]	0.657
Structured Streaming Skeletons [97]	0.817
Actionlet Ensemble [23]	*0.88*
DBN+HMM (this work)	*0.82*

Table 3.2: Recognition accuracy on MSR-Action3D dataset compared to state-of-the-art approaches.

outperforms two other baselines, the margins become smaller because only less than 10,000 frames in MSR Action3D dataset so that limited frames would not bring the advantages of generative pre-training into play.

We compare our model with the state-of-the-art methods on the cross-subject test setting as in [23] where training and testing sets are split by half of the actors. Though various idiosyncratic experimental set ups make it hard to have a fair comparison and generally render our generic 20 classes model at a disadvantage, (*e.g.* [23] with parameters that are empirically selected with data set dependent further tuning), the performance in Table 5.1 still exhibits the reasonable effectiveness of our model for this small frame number dataset.

3.7.3 MSRC12 dataset

The MSRC12 dataset [21] (details can be found at 6.1.4) is originally proposed to investigate what is the most relevant semiotic modality of instructions for delivering to human performers. We conduct our experiments on sequences with a compound semiotic modality, (*i.e.* tagstream with letter *"A"*, such as Video + Text or Image + Text) and follow a "leave-persons-out" protocol, using 14 sequences of each gesture class for training (note that each sequence contains multiple gesture instances), leaving 4-6 sequences for intra-modality testing or 29-30 sequence for inter-modality testing.

For training the network, we set the ground truth action point annotation as the middle state of the target class, encoding a window of 100 frames centered

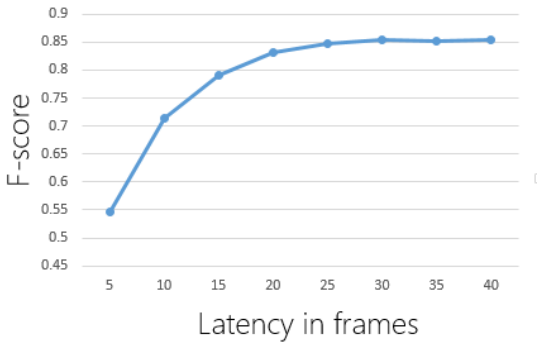


Figure 3.9: Latency profile of the MSRC 12 action point spotting task. We plot F-score as a function of the tolerated latency of the window Δ for the *DBN+ES-HMM*. Each frame lasts 33ms. The plateau is largely due to low recall (missing detection).

Modality / F-score	intra-modality	inter-modality
Randomized Forest [21]	0.621	0.576
Structured Streaming Skeletons [97]	0.718	-
DBN-ES-HMM (this work)	***0.7243***	***0.7098***

Table 3.3: F-score at $\Delta = 333ms$ for intra-modality and inter-modality test of MSRC12 dataset.

around the action point, with the rest of frames encoded as the $\mathcal{ES}$. We assess the dual-modality generalization performance for all 12 gestures and compare against the random forest recognition system [79] which has been successfully integrated into a gaming console that is currently being sold in retail stores and most recently proposed Structured Streaming Skeletons in the metric of F-score in Table 3.3. Fig. 3.8 illustrates a "Wind up the music" metaphoric action instance and the visualization of action point detection. Fig 3.9 plots the F-score at tolerated latency for the ES-HMM.

3.8 Discussion

In this chapter, feature extraction from skeletal joints data has been made an implicit approach by utilizing deep belief networks. By encoding dynamic structure into a HMM-based model, the discriminative trained, hierarchical

parametric model excelled the GMM paradigm at better estimating emission probabilities for the directed graphical model. Furthermore, the introduction of an ergodic states rendered the framework being able to anchor the precise temporal locations of actions that are voluntarily,momentary performed and discrete in essence. Experiments have confirmed the efficacy of the framework at better estimating observation model than the GMM and integration of temporal domain knowledge exceeds the Bag-of-Visual-Word approach.

Chapter 4

Deep 3D Convolutional Dynamic Networks

4.1 Introduction

This chapter presents a generalized hierarchical dynamic framework that first extracts high level features from contextual frames and then deploys the learned representation for estimating the emission probabilities to infer the label of a video sequence. It combines the power of deep neural networks, specifically, 3D Convolutional Neural Networks (3DCNN) for extracting high level spatio-temporal features with the graphical models of both Bayesian and Markov networks, yielding a scalable and powerful probabilistic graphical model that applies to acyclic video sequence labelling problems. The proposed framework labels a video sequence in a frame-to-frame mechanism, rendering it possible for online segmentation and recognition for both RGB and depth images.

To tackle the problem of learning thousands categories of objects from more than one million of images, a model with a tremendous learning capacity is required. However, colossal intricacy of the object recognition tasks means that this problem could not simply be solved by a dataset as mammoth as ImageNet. Consequently, the model should incorporate prior knowledge to make up to the gargantuan data we are not able to obtain. Convolutional neural networks (CNNs) [13] embody one such type of model. CNNs are flexible

in varying their depth and breadth, incorporating the strong and mostly veracious assumptions about the innate properties of the images, *i.e.* locality constraints of neighbouring pixels and stationarity of statistics. Therefore, in contrast with the standard fully connected feedforward neural networks with comparable-size layers, CNNs have much fewer connections. Hence, much fewer weights need to be trained whilst maintaining a sizable-scalably structure.

However, direct and unconstrained learning of complex problems is difficult, since (i) the amount of required training data increases steeply with the complexity of the prediction model and (ii) training highly complex models with very general learning algorithms is extremely difficult. It is therefore common practice to restrain the complexity of the model and this is generally done by operating on small patches to reduce the input dimension and diversity [10], or by training the model in an unsupervised manner [9], or by forcing the model parameters to be identical for different input locations (as in convolutional neural networks [13, 14, 15]). In this paper, a novel model of the latter category is proposed, which is adapted to the video sequence. We focus on data driven analysis of acyclic video sequence labeling problems, *i.e.* video sequences are non-repetitive as opposed to longer repetitive activities, *e.g.* jogging, walking and running.

Problem formulation: Giving a video sequence $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$, instead of finding the global label $\mathbf{Y}$ directly, we dissect the problem into finding the individual $\{\hat{\mathbf{y}}_\mathbf{1}, \hat{\mathbf{y}}_\mathbf{2}, \ldots, \hat{\mathbf{y}}_\mathbf{t}\}$ and reasoning with a higher level Markov field to obtain the most likely label $\hat{\mathbf{Y}}$.

Framework characteristics:

i) Relying on a pure learning approach, all the information are obtained from the data without sophisticated pre-processing or dimensionality reduction via manifold learning methods. An advantage of a fully-automatic learning-based method is that it incorporates the feature learning and classification procedures in a unified framework by minimizing the energy (*i.e.* optimizing the object function). Therefore, the proposed framework is more adaptable to different object functions or different input sensory modalities (*e.g.* RGB *vs.* depth).

ii) By unifying the deep learning paradigm with Markov fields via a factor graph representation, and investigating the learnt filters for the 3DCNN, the proposed feed forward neural networks offer several potential advantages as a better estimator for emission probabilities of the Markov field over the traditional paradigms (*e.g.* Gaussian mixture models) because its estimation of the posteriori probabilities does not rely on particular requirement about the data distribution.
iii) The labeling framework is based on contextual frames by incorporating prior knowledge of the intrinsic properties of the video sequence: instead of brutally flattening a sequence of image patches as in [98], we adopt the parameter tying scheme by the spatial-temporal constraint. Therefore, by employing a dynamic time programming scheme, the system is scalable to various time length sequences and is easily adapted for simultaneously segmenting and recognizing video sequences, discovering anchor points.

4.2 Probabilistic Graphical Model Unification

In this paper, we unify the deep neural nets and the probabilistic graphical model in a factor graph representation as in Fig. 4.1. Especially, as we incorporate the prior knowledge in modeling video sequences, *i.e.*, spatial-temporal constraint and state space constraint, by adopting 3D convolutional neural networks (ergo, weights sharing), the model can be scaled to real-sized video sequences.

4.2.1 3D convolutional neural networks

Convolutional Neural Networks (CNN) are variants of Multi-layer perceptrons which are inspired by biology. For 2D CNN, the weights $\mathcal{W}$ of a layer can be parametrized as a 4D tensor: source feature map index, source vertical position index, source horizontal position index and destination feature map index. Analogously, in the case of 3D CNN, $\mathcal{W}$ is parameterized as a 5D tensor with a source time index.

For action recognition, it is preferable to capture the motion information

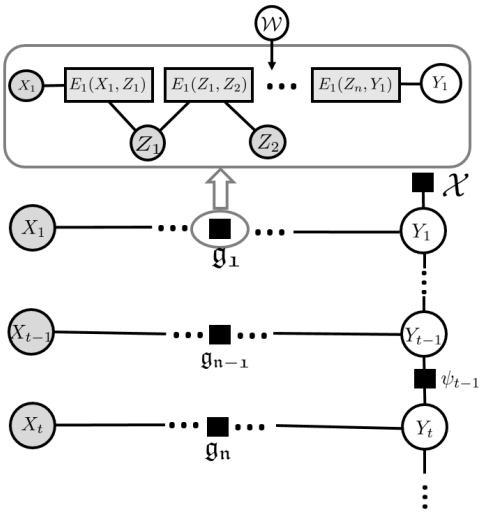


Figure 4.1: Factor graph representation for our 3DCNN Markov Field. Top plate model: the factor graph of Deep Convolutional Neural Networks is governed by the same set of model parameters $\mathcal{W}$, $Z_1, Z_2, \ldots, Z_n$ are latent variables with n layers in the deep net, X is the input contextual frame and Y denotes the output emission probabilities of the hidden states. Bottom: higher level factor graph with pairwise potential ψ and prior $\mathcal{X}$. If the graph is directed, the top part can be seen as a Hidden Markov Model; if undirected, the top part is the standard linear-chain Conditional Random Field. Note that we clamp the intermediate factors $\mathfrak{g}$ during parameter estimation to extract higher level features via the deep net structure.

encoded in multiple successive frames. Ji *et al.* [15] are the first to propose using 3D convolution in the place of 2D convolution for video analysis. However, in their model, the 3DCNN is treated as a holistic descriptor for action detection–the whole action sequence needs to be of a fixed length (7 frames in their surveillance detection), constrained by the convolutional structure. In our proposed model, however, the 3DCNN is used to extract an intermediate spatial-temporal descriptor, and the higher level temporal information will be encoded in a Markov field, rendering it possible for the model to accommodate various longer sequences.

The 3D convolution is attained by convolving a 3D kernel to the cuboid structured by stacking multiple successive frames together. We follow the nomenclature as in [16]. Formally, the value of a unit at position (x, y, z) (z here corresponds the time-axis) in the jth feature map in the ith layer, denoted

as v_{ij}^{xyz}, is given by:

$$v_{ij}^{xyz} = 1.7159 * tanh(\frac{2}{3}(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(t+r)})) \tag{4.1}$$

where $tanh(\cdot)$ is the hyperbolic tangent function, w_{ijm}^{pqr} is the value at the position (p, q, r) of the kernel, b_{ij} is the bias for this feature map, m indexes over the set of feature maps in the $(i-1)$th layer connected to the current feature map, and P_i, Q_i, R_i are the height, width and number of contextual frames of the kernel, respectively. Note that here we differ from [16] by using the scaled hyperbolic tangent function, saturating the node output to accelerate the learning process–a similar "trick" has been adopted for LeNet [99].

The Rectified Linear Units (*ReLUs*) were adopted in [13], shown in Fig 4.2 where trainings are almost an order faster faster than their equivalents *tanh* units. The activation neurons expressed in terms of ReLUs are:

$$v_{ij}^{xyz} = max(0, (b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(t+r)})) \tag{4.2}$$

Note that [13] claim the the nonlinearity function set $f(x) = |tanh(x)|$ works especially well with their type of contrast normalization followed by local average pooling. It is also imperative for faster learning since it has a tremendous impact on the performance of large models trained on large-scale dataset.

A typical 3D CNN for one of our experiments is demonstrated in Fig. 4.3.

4.2.2 3DCNN Markov Field

For sequential modeling, there are two broad categories: generative models and discriminative models. In this paper, we foster the marriage between 3DCNN and a continuous-observation Hidden Markov Model with discrete hidden states. And if the problem is discriminative in nature, a linear-chain Conditional Random Field is adopted to directly minimize the error rate. For each time step t, there is one corresponding contextual frame random obser-

Algorithm 3: Deep 3D Convolutional Dynamic Networks – training

Data:

$\mathbf{X} = \{\mathbf{x_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - raw input feature sequence in the form of $M_1 \times M_2 \times T$, where M_1, M_2 are the height and width of the input image and T is the number of contiguous frames of the spatio-temporal cuboid.

Note that the GPU library *cuda-convnet* [13] used requires square size images and T is a multiple of 4.

$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - frame based local label (achieved by semi-supervised forced-aligment),
where $\mathbf{y_i} \in \{C * S + \boldsymbol{1}\}$ with C is the number of class, S is the number of hidden states for each class, $\boldsymbol{1}$ as ergodic state.

1 Preprocessing the data $\mathbf{X}$ (normalizing, median filtering the depth data).
2 Feeding the above features to Eq.4.2.
3 Supervised fine-tuning the Deep 3D Convolutional Neural Networks using standard SGD Backpropagation.

Result:

3DCNN - a 3D Deep Convolutional Neural Networks to generate the emission probabilities for hidden markov model.

$\mathbf{p}(\mathbf{H_1})$ - prior probability for $\mathbf{Y}$.

$\mathbf{p}(\mathbf{H_t}|\mathbf{H_{t-1}})$ - transition probability for $\mathbf{Y}$, enforcing the beginning and ending of a sequence can only start from the first or the last state.

Algorithm 4: Deep 3D Convolutional Dynamic Networks – test

Data:

$\mathbf{X} = \{\mathbf{x_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - raw input feature sequence in the form of $M \times M \times T$.

3DCNN - the trained 3D Deep Convolutional Neural Networks to generate the emission probabilities for hidden markov model

$\mathbf{p}(\mathbf{H_1})$ - prior probability for $\mathbf{Y}$

$\mathbf{p}(\mathbf{H_t}|\mathbf{H_{t-1}})$ - transition probability for $\mathbf{Y}$

1 Preprocessing the data $\mathbf{X}$ (normalizing, median filtering the depth data).
2 Feedforwarding **3DCNN** to generate the emission probability $\mathbf{p}(\mathbf{X_t}|\mathbf{H_t})$ in Eq.3.25
3 Generating the score probability matrix $\mathbf{p}(\mathbf{H_{1:T}}, \mathbf{X_{1:T}})$ from Eq.3.25
4 Finding the best path $\mathbf{V_{t,\mathcal{H}}}$ by Viterbi decoding as in Eq.3.26

Result:

$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - frame based local label
where $\mathbf{y_i} \in \{C * S + \boldsymbol{1}\}$ with C is the number of class, S is the number of hidden states for each class, $\boldsymbol{1}$ as ergodic state

$\mathbf{C}$ - global label, the anchor point is chosen as the middle state frame

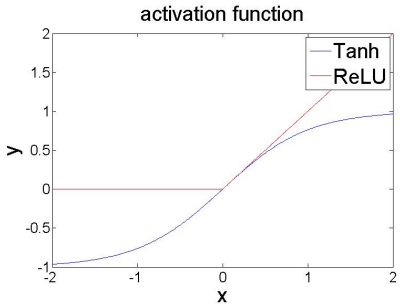


Figure 4.2: Tanh and ReLUs activation function.

vation variable X_t. Additionally, an unobserved hidden variable Y_t belongs to a finite set $\mathcal{Y} = (\bigcup_{a \in \mathcal{A}} \mathcal{Y}_a)$,where $\mathcal{Y}_a$ is a set of sub-states assigned by the force-alignment scheme for each action class a. The intuition motivating this construction is that the absolute duration of each action sequence could vary. This variance is encapsulated by enabling flexible forward transitions within the chain.

Assuming each of the conditional distributions is independent of all previous observations except the most recent, the full probability model can be defined as HMM as Eq 3.25, where $p(Y_1)$ is the prior on the first hidden state and in all our experiments, we have a uniform prior; $p(H_t|H_{t-1})$ is the transition dynamic model; and $p(X_t|Y_t)$ is the observation model and is estimated by the 3DCNN.

In the case of CRF, traditional practical models rely extensively on parameter tying, *e.g.* in the linear-chain case, and the same weights are often used for the factor $\psi_t(Y_t, Y_{t-1}, X_t)$ at each time step. In our 3DCNN-CRF framework, parameter tying is the natural approach. The CRF with cliques C_t with weights $\mathcal{W}$ can be written as:

$$p(Y|X) = \frac{1}{Z(x)} \prod_{\psi \in C_t} \psi_t(X_t, Y_t; \mathcal{W}), \tag{4.3}$$

with the global normalization factor as:

$$Z(x) = \sum_{Y} \prod_{\psi \in C_t} \psi(X_t, Y_t; \mathcal{W}), \tag{4.4}$$

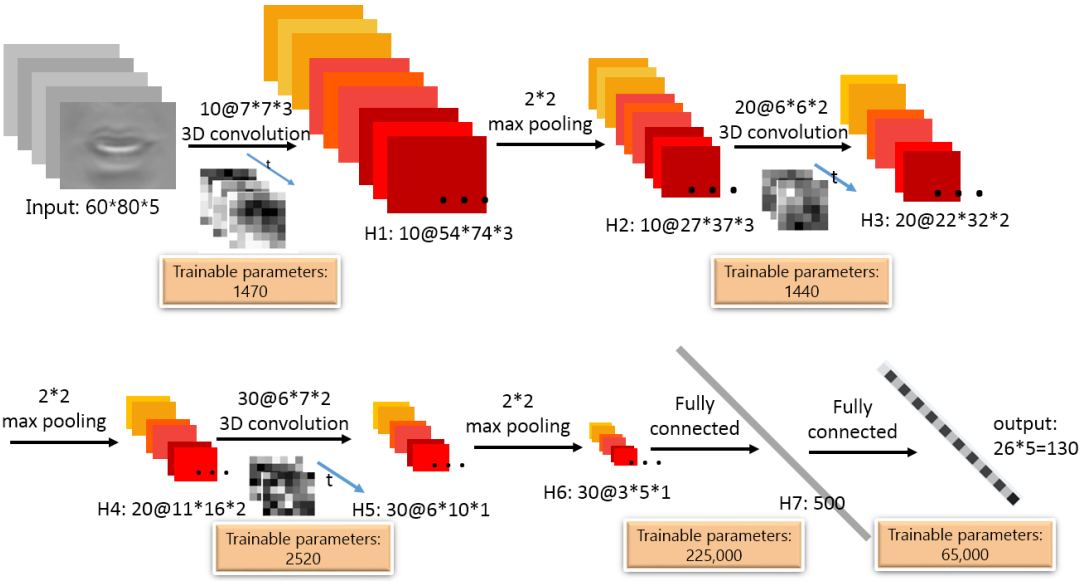


Figure 4.3: A typical example of 3D Convolutional Neural Networks for modeling the emission probability $\mathfrak{g}$ as in Fig. 4.1. The input is a *height* $\times$ *width* $\times$ *frames* contextual sequence and the output is of *number of classes* $\times$ *number of the hidden states* dimension under the parameter tying scheme. We illustrate the networks by a specific example, displaying the trainable parameters for each layer. It can be seen that most trainable parameters lie in the last two fully connected layers. By adopting the weights sharing scheme in the first several convolutional layers, as opposed to the Deep Belief Network architecture, the incorporation of prior knowledge of the intrinsic spatial-temporal property for a video sequence renders it possible to learn the filters without the explosion of model parameters.

Commonly, the potential function is chosen from the exponential family with energy E_t and $\psi_t(X_t, Y_t; \mathcal{W}) = exp\{-E_t(X_t, Y_t; \mathcal{W})\}$. To make learning tractable, a common setting is to use linear energy functions $E_t(X_t, Y_t; \mathcal{W}) = - < \mathcal{W}_t^{Y_t}, \psi(X) >$. This results in the most often used log-linear model [100]. As we have discussed, a linear energy function is the major bottleneck for CRF, limiting its capacity to extract higher level features. We propose 3DCNN to substitute this shallow structure of log-linear model.

In a chain-structure CRF, there are two types of cliques as in Fig. 4.1:

singleton factors $\psi_c(X_t, Y_t)$ at each time t, whose potential functions are obtained by the forward pass of 3DCNN. This corresponds to the emission probability of an individual contextual video frame to be assigned as one of the states for a motion sequence.

pairwise factors $\psi_c(Y_{t-1}, Y_t)$ between two consecutive frames $t-1$ and t that represent the interactions between adjacent pairs of the motion sequence. Intuitively, consider a task of predicting the trajectories of a gesture writing letter "Z" in the corpus of "A" to "Z", a top right to bottom left diagonal stroke is more likely to happen at the two corresponding end points than a curvy stroke of ↄ because we have never seen the latter case during our training stage. Hence, this enhances the belief of the model's former prediction.

While the 3DCNN-CRF framework is quite generic, for the sake of inference tractability, we focus on linear-chain CRF based on a first-order Markov chain. Note that, by incorporating other types of factors, *e.g.* triplet factors, similarity factors might further enhance the discriminative power of the model [75]. However, focusing on first-order Markov chain allows examining the actual effect of 3DCNN-CRF on video sequence labeling tasks. Moreover, exact inference is achieved for chain structure CRF, enabling efficient belief propagation by variants of the standard dynamic programming algorithms.

4.2.3 Inference

For inference, there are two steps. First, a feed forward 3DCNN is passed through to compute the emission probability $p(X_t|Y_t)$. In a second step, the dynamic programming is used to find the output $\hat{Y}$ with minimum energy. Specifically, for the hidden Markov model, as a directed tree, the inference can be solved exactly using the max-sum algorithm. We can infer the action presence in a new sequence by Viterbi decoding the same as Eq.3.26, where initial state $V_{1,\mathcal{Y}} = \log(P(Y_1|X_1))$. From the shortest path, we decode the probability of an action $a \in \mathcal{A}$ as $p(Y_t = a|X_{1:t}) = V_{T,\mathcal{Y}}$.

Similarly, inference in CRFs consists of finding $\hat{Y}$ that best matches input X

(*i.e.* with lowest energy):

$$\hat{Y} = \underbrace{argmin}_{Y} p(Y|X, \mathcal{W}) \tag{4.5}$$

$$= \underbrace{argmin}_{Y} \sum_{c \in C} E_c(X, Y_c, \mathcal{W}) \tag{4.6}$$

The linear-chain CRFs inference task can be performed efficiently and exactly. We construct the factor graph and adopt the belief propagation scheme using the libDAI [101] library.

4.2.3.1 Inference Library: libDAI

The inference library adopted is libDAI [101][1] which is a free and open source C++ library. Various implementations of exact and approximate inference methods for probabilistic graphical models with discrete-value variables are provided. The libDAI library supports directed graphical models (Bayesian networks) as well as undirected graphical models (Markov random fields and factor graphs).

Due to non-convexity of the objective function, initialization is a vital step for effective learning in deep neural networks. As it's mentioned in previous chapter, we could utilize the unlabeled data for initializing the deep neural networks (such as Deep Auto-Encoder [102]) and the limited labeled data are used for the final fine-tuning. It's also expected with good initialization from the generative pre-training, the global performance of 3DCNN Markov Field could be further improved since the bottom feature detector part plays a crucial part in extracting relevant high-level representations. It is a perspective of our work that we have not yet explored. For the deep architecture, we adopt the standard stochastic gradient descent for the first half epochs to find a good initialization, then use conjugate gradient (or other quasi-Newton method, *e.g.* BFGS) for the rest of the epochs. For the higher level graphical part, we clamp the parameters of 3DCNN, and only the transitional (pairwise) potential is learned.

[1] https://staff.fnwi.uva.nl/j.m.mooij/libDAI/

4.3 Related Works

Previous works on video-based motion recognition focused on adapting handcrafted features and low-level hand-designed features have been heavily employed with much success. These methods usually have two stages: an optional feature detection stage followed by a feature description stage. Well-known feature detection methods ("interest point detectors") are Harris3D [1], Cuboids [2] and Hessian3D [3]. For descriptors, popular methods are Cuboids [4], HOG/HOF [1], HOG3D [5] and Extended SURF [3]. In a recent work of Wang *et al.* [6], dense trajectories with improved motion-based descriptors epitomized the pinnacle of handcrafted features and achieved state-of-the-art results on a variety of "in the wild" datasets. Given the current trends, challenges and interests in action recognition, this list would probably continue to spread out extensively.

In the evaluation paper of Wang *et al.* [7], one interesting finding is that there is no universally best hand-engineered feature for all datasets, suggesting that learning features directly from the dataset itself may be more advantageous. Albeit the predominant techniques for visual recognition from images and video depends on hand-crafted features, there has been a burgeoning shift to the techniques that automatically extract low-level and mid-level features, either in supervised, semi-supervised or unsupervised context [8, 9, 10].

With the recent resurgence of neural networks invoked by Hinton and others [11, 99], deep neural architectures have been served as an effective solution for extracting high level features from data. Such models have been successfully applied to a plethora of different domains: the GPU-based cuda-convnet [13] classifies 1.2 million high-resolution images into 1000 different classes; multi-column Deep Neural Networks [14] achieve near-human performance on the handwritten digits and traffic signs recognition benchmarks; 3D Convolutional Neural Networks [15, 16] recognize human actions in surveillance videos; Deep Belief Networks combining with Hidden Markov Models [17] for acoustic modeling outperform the decade-dominating paradigm of GMM+HMM. In these fields, deep neural nets have shown great capability to detect and extract higher level relevant features.

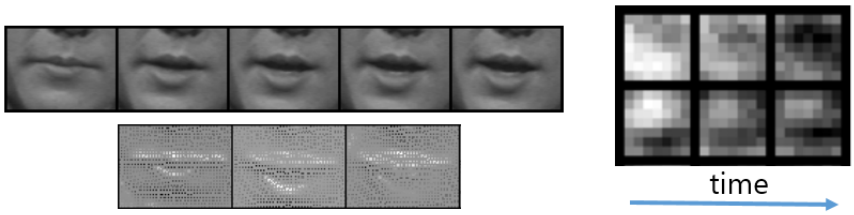


Figure 4.4: Top Left: example frames of uttering a letter "A"; bottom left: first filtered layers (normalized for visualization), note how the lip areas have been detected. Right: examples of learned first layer filters: we only show two maps of filters. Note that the motion has been detected from the time axis and the spatial pattern within a 7×7 filter bank.

Method	Accuracy
HOG3D [5]	0.488
Multiscale Spatial Analysis [103]	0.446
Local Binary Pattern [104]	0.589
Baseline Preprocessed Video [98]	0.462
RBM Video [98]	0.542
3DCNN+HMM (this work)	***0.596***
3DCNN+CRF (this work)	***0.604***

Table 4.1: Recognition accuracies on the AVLetters lipreading dataset compared to state-of-the-art approaches. Top parts correspond to hand-crafted approaches and in the bottom parts we compare with the learning-based methods.

4.4 Experimental Results

To demonstrate the proposed 3DCNN Markov Field's effectiveness in modeling spatio-temporal video data, we performed experiments on two video labeling problems that have distinct modalities to show the proposed architecture's ability to generalize.

4.4.1 Lipreading task

AVLetters [103]. This dataset 6.1.6 pconsists of 10 speakers saying the letters "A" to "Z", three times each. The dataset provides pre-extracted lip regions of 60×80 gray scale pixels. The evaluation is a visual-only lipreading task which is an intrinsically challenging task. We report results with the *third-test* settings

Method	correct count	accuracy
singleton	491/691 letters	71.06%
	11/100 words	11%
singleton+ pairwise	539/691 letters	78.00%
	27/100 words	27%

Table 4.2: An individual ("John") letter recognition rate and word recognition rate. It can be seen that incorporating pairwise information can help further differentiate the difficult lip motion pairs. The dataset comprises of 100 words, and each word is in the range of 3-9 letters.

used by [98, 103, 104] for comparison. Some example frames of the sequences are shown in 4.4.

We specify the 3DCNN architecture as follows: the input contextual frames are of size $60 \times 80 \times 5$, the first layer contains 10 maps of $7 \times 7 \times 3$ 3D kernel followed by max pooling; the second convolutional layers has 20 maps of $6 \times 6 \times 2$ 3D kernel followed by max pooling; the third convolution layer is composed of 20 maps of $6 \times 6 \times 2$ 3D kernel followed by max pooling; then we have one fully connected layer of size 500; the output layer is of size 5×26 (number of hidden states for each class$\times$ number of classes). Note that, for max pooling, we only pool over the spatial axis and never pool along the temporal axis. We run the first 50 epochs with standard SGD and another 50 epochs with the conjugate gradient method.

We include the results using only video information throughout training and testing in Table 4.1 for a fair comparison. Note that the results in [98] reach accuracies of 0.592 and 0.644 by a Bimodal Deep Autoencoder and Video-Only Deep Autoencoder respectively and both auto-encoders incorporate the audio information during the auto-encoder reconstruction scheme. Integrating audio information into our 3DCNN Markov Field framework will be investigated in our future work.

4.4.1.1 Markov Field at a Higher Level:

We demonstrate our model's effectiveness in extracting relevant features and its ability to serve as a convenient plug-in functionality by superimposing another Markov Field on top of the overall structure of Fig. 4.1.

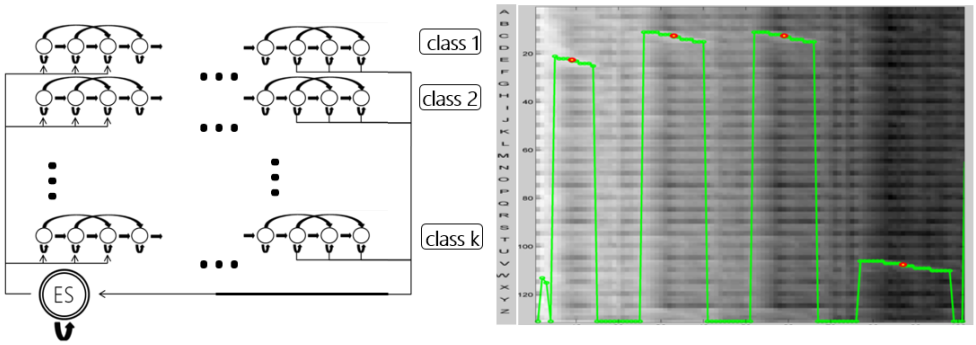


Figure 4.5: Left: an ergodic states (ES) is included between video sequences. Each node represents a single frame and each row represents a single dynamic model. The arrows indicate possible transitions between states. Right: an utterance of "ECCV" sequence. The fluorescent green is the Viterbi path from backtracking, a zoom in shows a path from states 20-25 (y-axis) indicating the lipreading class number 5 because we assign 5 states for each hidden Markov Model (left panel). Red circles are the predicted anchor points (middle frame of the Viterbi best path).

4.4.1.2 Transfer learning:

we synthesize an experiment that models the 3D sequences from the *AVLetters* dataset and use the word level letter pairwise potential from an optical character recognition (OCR) dataset compiled by Koller and Friedman [75]. On the higher level we superimpose another CRF, *i.e.*, the nodes of the CRF are the output probabilities given by a single 3DCNN+HMM. Instead of having optical characters as input, we take sequences of a single person's lip motion as input by incorporating the given pairwise potential of the compiled word corpus. We can further improve individual letter recognition rate as shown in Table 4.2 . For example, for the word *"torturing"*, without pairwise information, the singleton model predicts *"torturilg"* whereas the pairwise model correctly predicts *"torturing"*. Indeed, the lip motion of letter "L" and "N" are hard to differentiate (the only difference may hide in the motion of the unobserved tongue). Note that, during this experiment, only inference is required and no parameter learning is happening at this stage.

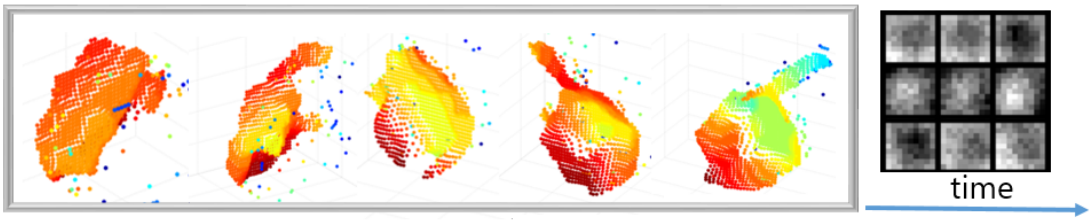


Figure 4.6: Left: point cloud projection from depth images of a gesture *"J"*. Right: 3 pairs of learned $9 \times 9 \times 3$ 3D filters in the first layer. It can be seen that some 3D kernels focus more on temporal modeling (top and bottom rows) whereas some focus more on spatial modeling (middle row). Because depth images are much noisier than the RGB images in the previous task, the learnt kernels correspondingly exhibit noisier patterns.

4.4.1.3 Anchor point discovery:

We also demonstrate that the framework could be used as a sub-module for simultaneously segmenting and recognizing video sequences by introducing an ergodic state. A toy example is illustrated in Fig. 4.5 as someone enunciates the letter sequence "ECCV". The anchor points have been successfully detected as the middle frame of the predicted video sequence. We believe that such a lipreading module could be used for CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof) for online identification.

4.4.2 Depth sequence gesture recognition

MSRGesture3D [105]: we demonstrate the generality of the proposed method on a different sensory input: a hand gesture dataset captured by a depth camera. To our knowledge, this is the first time that a pure learning approach without any hand-crafted features involved is adopted for the dynamic video sequence labeling task from depth images. This dataset contains a subset of gestures defined by American Sign Language (ASL). There are 12 gestures in the dataset and details can be found at 6.1.5. A point cloud projection sequence of one example gesture and the learnt 3D filter maps are shown in Fig. 4.6.

The 3DCNN architecture is almost the same as in the previous lipreading task: we resize the input contextual frames to the size of $80 \times 80 \times 5$; the first

Method	accuracy
SVM on Raw Features	0.44
3DHOG [47]	0.66
HON4D [106]	0.75
3DCNN+HMM (this work)	*0.69*
3DCNN+CRF (this work)	*0.72*

Table 4.3: Recognition accuracies on MSGesture3D, training on the 2-10 subjects and testing on the first subject with 36 testing sequences.

layer has 10 maps of $9 \times 9 \times 3$ 3D kernel followed by max pooling; the second convolutional layer contains 20 maps of $7 \times 7 \times 2$ 3D kernel followed by max pooling; the third convolution layer is composed of 20 maps of $6 \times 6 \times 2$ 3D kernel followed by max pooling; then there is one fully connected layer of size 500; finally, the output layer is of size 10×12. And we observe that there is a consistent slight performance gain for CRF over HMM at a higher level.

In Table 4.3, we compare the results with the state-of-the-art [106] where the first subject is used for testing and subjects 2-10 are for training. Competitive results have been achieved. Note that, in their Histogram of Oriented 4D Normals (HON4D) model, a very task-specific holistic descriptor is used for depth images refined by a discriminative density measure. In contrast, our proposed model is trained from scratch and more adaptive to various sensory inputs. Notwithstanding, we explicitly model time variance which we consider a merit for the tasks of simultaneously video segmentation and recognition. Furthermore, the number of parameters in our model is drastically larger than the number of training instances (less than 15,000 frames in this dataset). Therefore, overfitting is unavoidable in the proposed framework. Consequently, we reckon that, with more training instances, it is safe to conclude that the margin of performance improvement will be more limited for the ad-hoc features in [106] whereas there would still be a large improvement for our model.

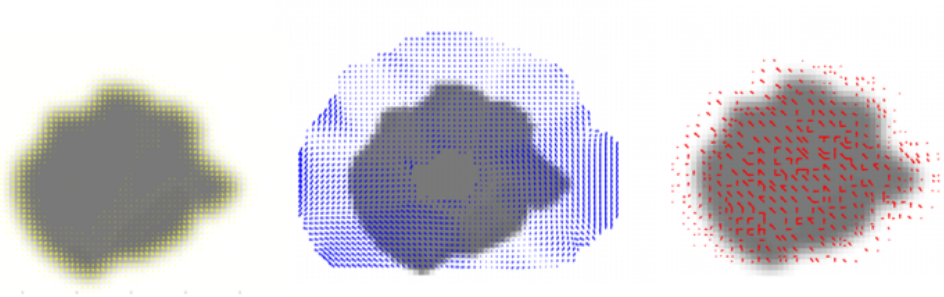


Figure 4.7: Left: spatial gradient (yellow) around an image of a fist; middle: optical flow (blue) of two frames; right: the filtered image convolved with the learned 3D kernels (red). Note that the learned filtered image exhibits both spatial and temporal activation. Even inside the seemly more uniform palm area, there are still some strong activation signals passing to the next convolutional layer.

4.4.3 Looking into the filtered maps

We show the quiver plot of filtered images convolved with learned 3D kernels in Fig. 4.7 (red). In the work of Ji *et al.* [16], the first layers are hand-crafted (*e.g.*, spatial gradient, optical flow, *etc.*). Their approach incorporates human prior knowledge of what matters for action classification and helps the networks better initialize. We argue that with a proper learning paradigm, the relevant information should be automatically extracted as well. Another interesting conclusion in [16] is that by juxtaposing an auxiliary hand-tuned features (dense BoW SIFT and motion edge history images) at the penultimate layer would serve as a regularization scheme. Such combination of well studied and problem specific features might further boost the performance of our proposed architecture and worth further investigating.

4.4.4 Computational complexity

Though learning the 3D convolutional nets using stochastic gradient descent is tediously lengthy, once the model finishes training, with a low inference cost, our framework can perform real-time video sequence labeling. Specifically, a single feed forward convolutional neural network incurs trivial computational time ($\mathcal{O}(T)$) and is fast because it requires only matrix products and convolu-

tion operations. The complexity of Viterbi algorithm is $\mathcal{O}(T * |S|^2)$ with number of frames T and state number S. Similarly, the inference for linear-chain CRF requires $\mathcal{O}(T * |S|^2)$ in the worst case. At prediction time, the method is as fast as other hand-engineered features such as 3DHOG.

4.5 Discussion

This chapter presented a framework that utilises 3D Convolutional Neural Networks for learning contextual frame-level representations and estimates the emission probabilities for Markov fields. These models extract relevant features from both temporal and spatial dimensions by performing 3D convolutions for the task of video sequence labelling. The results show that our single model, using the same architecture across two sensory inputs, *i.e.* RGB and depth, is consistently as good as or better than a wide variety of handcrafted and other learning based methods. It also suggests that learning features directly from data is a very important research direction. With more and more data and flops-free computational power, the learning-based methods are not only more generalisable to many domains, but also are powerful in combining with other well-studied probabilistic graphical models for modelling and reasoning dynamic sequences.

Chapter 5

Multimodal Deep Dynamic Networks

5.1 Introduction

This chapter presents a novel multi-modal dynamic network for time series prediction. Multimodal input is a real-world situation in gesture recognition applications such as sign language recognition. In the first part, two Deep Belief Dynamic Networks are deployed to extract high level audio and skeletal joints representations. Instead of traditional late fusion, another layer of perceptron for cross modality learning taking the input from each individual net's penultimate layer is deployed as the top of Fig 5.1. In the second part, two heterogeneous Deep Neural Networks: a Deep Belief Dynamic Networks for skeletal module and a Deep 3D Convolutional Dynamic Network for depth image module are deployed as in the bottom of Fig 5.1. In particular, we demonstrate that multimodal feature learning will extract semantically meaningful shared representations, outperforming individual modalities, and the early fusion scheme's efficacy against the traditional method of late fusion.

Multimodal learning involves relating information from multiple sources. For example, images and depth scans are correlated at first-order as depth discontinuities often manifest as strong edges in images. Conversely, audio and visual data for gesture recognition have correlations at a "mid-level", as

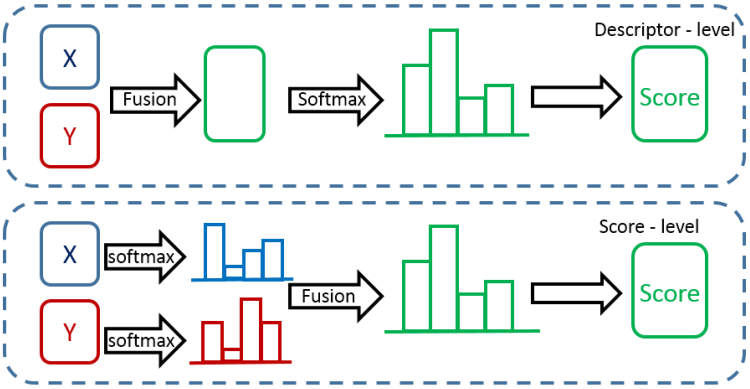


Figure 5.1: Different Pipelines for descriptor fusion.

phonemes and joints motions; it can be difficult to relate joint spatio-temporal information to audio waveforms or spectrograms. Learning from multimodal inputs is technically challenging because different modalities have different statistics and different kinds of representations. For instance, text is discrete and often represented by very large and sparse vectors, while images are represented by dense tensors that exhibit strong local correlations. Traditional multi-agent systems tend to adopt the late fusion scheme by normalizing the confident values from an individual modality for final prediction, ignoring the subtle intrinsic properties within different modalities. Fortunately, deep learning has the promise to learn adaptive representations from the input, potentially bridging the gap between these different modalities.

In this chapter, a novel framework of bimodal/multimodal dynamic networks is proposed for continuous gesture recognition given 3D joint positions, depth image sequence and the audio utterance of the gesture tokens. We focus on data driven analysis of acyclic skeleton-audio, skeleton-depth sequence labeling problems. The model has been designed with bimodal gesture recognition in mind, but should extend itself well to other multimodal high-dimensional time series data.

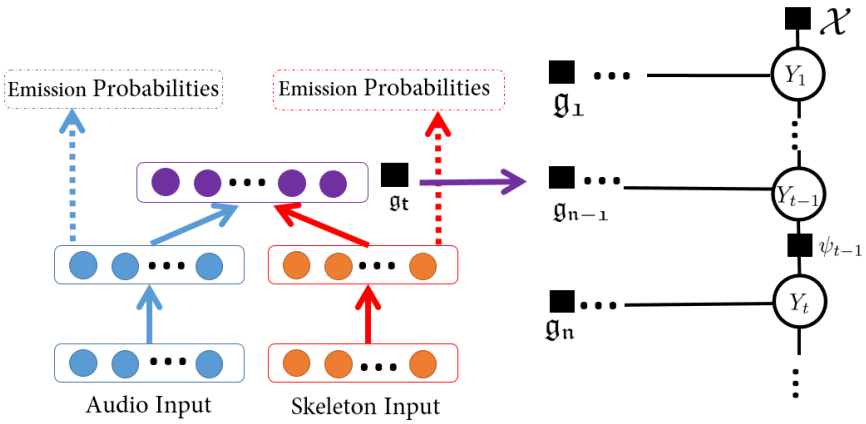


Figure 5.2: Architecture of the multimodal dynamic networks: each modality (audio or skeleton input) is first pre-trained by a Deep Belief Network, and their penultimate layers are fused together to generate a shared representation for dual modalities. The outputs are the emission probabilities $\mathfrak{g}_t$ for temporal dynamic modeling. In our experiments, we assume each of the conditional distributions per frame is independent of all previous observations except the most recent, hence the higher level is specified as a Hidden Markov Model.

5.2 Architecture

Problem formulation: Given a multimodal input sequence $\mathbf{X^m} = \{\mathbf{x_1^m}, \mathbf{x_2^m}, \ldots, \mathbf{x_t^m}\}$, where $\mathbf{m}$ is the modal index (in our experiment $\mathbf{m} = \mathbf{2}$ because we only use the audio-skeleton pair or the skeleton-depth pair), instead of finding the global label $\mathbf{Y}$ directly, we dissect the problem into finding the individual $\{\hat{\mathbf{y}}_\mathbf{1}, \hat{\mathbf{y}}_\mathbf{2}, \ldots, \hat{\mathbf{y}}_\mathbf{t}\}$ and reasoning with a higher level Markov field to obtain the most likely label $\hat{\mathbf{Y}}$.

The earlier fusion architecture is shown in Figure 5.2. The individual emission probability estimators are based on the state-of-the-art architectures as in [17, 18]. Specifically, a Deep Belief Network is deployed for each modality to estimate the output emission probability. Since both feature modalities $\mathbf{X^m}$ are continuous instead of binomial features, the first visible layer is a Gaussian Restricted Boltzmann Machine to model the energy term.

The outputs of the neural net are the hidden states learned by force alignment during the supervised training process. Once each individual modality is trained, the penultimate layer is extracted and fused for the shared repre-

sentation. Then, the standard backpropagation can be adopted for adjusting the weight W^m for each modality m:

$$W^m = W^m - \alpha \frac{\partial}{\partial W^m} J(W) \tag{5.1}$$

where α is the annealed learning rate and $J(W)$ is the cost function (cross-entropy) of the last layer perceptron by feed forwarding the fused penultimate layers from mutli-modalities. The output of the network is $p(X_t|Y_t)$ denoting the fused emission probability and is denoted by $\mathfrak{g}_t$ in Figure 5.2.

Assuming each of the conditional distributions is independent of all previous observations except for the most recent, the full probability model is now established as an HMM the same as in Eq 3.25. We can infer the action presence in a new sequence by Viterbi decoding the same as Eq 3.26.

5.3 Related Works

Gesture recognition has been a popular research field in recent years due to its promising application prospects in human-computer interaction. In the early days of gesture recognition research, most approaches were controller-based, in which users had to wear or hold certain hardware for motion data capturing. In vision-based approaches, users motion data are captured by cameras and numerous computer vision methods have been successfully adopted into this area for further data analysis and understanding. Over the last few years, with the immense popularity of the Kinect, there has been renewed interest in developing methods for human gesture and action recognition from both 3D skeletal data and audio data captured synchronously by the device.

Deep learning is an emerging field of machine learning focusing on learning representations of data and has recently found success in a variety of domains, from computer vision to speech recognition, natural language processing, web search ranking, and even online advertising. The ability of deep learning methods to capture the semantics of data is, however, limited by both the complexity of the models and the intrinsic richness of the input to the system. In particular, current methods only consider a single modality leading to

an impoverished model of the world. Sensory data are inherently multimodal instead: images are often associated with text; videos contain both visual and audio signals; text is often related to social content from public media; etc. It is expected that the cross-modality structure may yield a big leap forward in machine understanding of the world. The proposed framework is mostly related to the works of [98, 107] in that, instead of the traditional late fusion, all resort to an early fusion scheme. In [98], the input sequences are treated as a holistic entity, hence, the method is not adaptable to various time length input. A multimodal Deep Boltzmann Machine is introduced in [107] to learn a generative model of the joint dimension of text and image inputs for information retrieval.

In order to model the time series data, the unimodal of our architecture is built upon the framework of [17, 18] which deploy Deep Belief Networks in the place of Gaussian Mixture Models to model the emission probabilities for Hidden Markov Models. However, our proposed framework is the first work to learn the shared representation for modeling multimodal dynamic time series inputs.

5.4 Experiments

5.4.1 Skeleton-Audio pair module

ChaLearn Italian Gesture Recognition-Kaggle track: this dataset is on "multiple instance, user independent learning" [20] of gestures. The details of the dataset can be found at 6.1.1.1. We focus on the skeletal modality and audio modality. Note that a large number of frames is advantageous in our model settings over other nonparametric models for estimating skeletal human poses.

5.4.1.1 Audio Features

The speech was dissected by a 25-ms Hamming window with a 10-ms fixed frame rate. 12th-order Mel frequency cepstral coefficients (MFCCs) and energy, along with their first and second temporal derivatives are the used as the

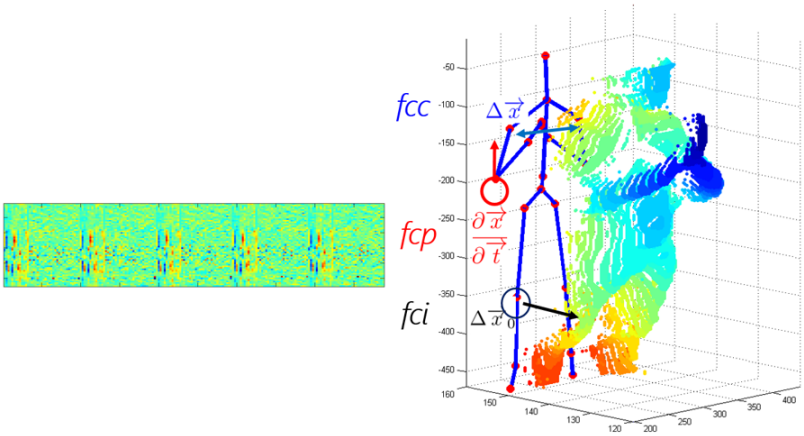


Figure 5.3: Input modules: left–audio input in the form of MFCC, and in order to conform to the 20 fps, 5 frames are concatenated together (10-ms fixed frame rate); right–skeleton 3D positional features.

standard preprocessing step. In order to conform to the 20 fps, 5 frames are concatenated together (10-ms fixed frame rate). Hence one audio frame will be of dimensionality $39 \times 5 = 195$. Before feeding into the *DBN*, the input were normalized so that each element of the inputs will have zero mean and unit variance averaged over all the training examples.

5.4.1.2 Skeleton Features

Only upper body joints are relevant to our discriminative gesture recognition tasks. Therefore, we consider only the 9 upper body joints for our task (full body joints have been compared, but as expected, led to inferior results compared to upper body 9 joints). The 9 upper body joints used are *"ShoulderCenter, ShoulderLeft, ElbowLeft, WristLeft, HandLeft, ShoulderRight, ElbowRight, WristRight, HandRight"*.

The 3D coordinates of N joints of frame c are given as: $X_c = \{x_1^c, x_2^c, \ldots, x_N^c\}$. We deploy 3D positional pairwise differences of joints [87] for observation domain $\mathcal{X}$. They capture posture features, motion features and offset features by direction concatenation: $\mathcal{X} = [f_{cc}, f_{cp}, f_{ci}]$ as demonstrated in the same as Eq 5.2. This results in a raw dimension of $N_{\mathcal{X}} = N_{joints} \times (N_{joints} - 1)/2 + (N_{joints}^2 + N_{joints}^2) \times 3$ where N_{joints} is the number of joints used. Hence, in our

Modality and Method	Classification rate
Audio Only, DBN+HMM [17]	0.554
Skeleton Only, DBN+HMM [18]	0.586
Audio + Skeleton, Model averaging	0.668
Multimodal DBN+HMM	***0.701***

Table 5.1: Recognition accuracy compared to the individual modal method and the multimodal confident score averaging scheme (late fusion) on ChaLearn Italian multimodal dataset.

experiment, $N_{joints} = 9, N_{\mathcal{X}} = 594$. Note that before extracting any features, all the 3D joint coordinates are transformed from the world coordinate system to a person centric coordinate system by placing the HipCenter (or ShoulderCenter if applied) at the origin. By including temporal differences f_{cp}, f_{ci} partly moderates the strong conditional independence preconditional of HMMs, *i.e.* continuous frames are independent to the previous frames given the current hidden states.

Admittedly, we do not completely neglect human prior knowledge about information extraction for relevant static postures, velocity and offset overall dynamics of motion data. Nevertheless, the aforementioned three attributes are all very crude pairwise features without any "tweak" into the dataset or handpicking the most relevant pairwise, triple wise, *etc.* , designed features [77, 78, 79, 80]. A similar data driven approach has been adopted in [21] where random forest classifiers were adapted to the problem of recognizing gestures using a bundle of 35 frames. These sets of feature extraction processes resemble the *Mel Frequency Cepstral Coefficients (MFCCs)* for the speech recognition community [17].

5.4.1.3 Dynamic Networks Setup

All *DBNs* were pre-trained with a fixed learning rate using *SGD* with a mini-batch size of 100 training examples. For Gaussian-bernoulli RBMs, a smaller learning rate is desirable, hence 250 epochs is run with a fixed learning rate of 0.002 while for binary-binary RBMs we used 75 epochs with a learning rate of 0.02 and with a mini-batch size of 100 training cases. For fine-tuning, the

learning rate starts at 0.1 with 0.998 scaling after each epoch.

For high level feature extraction, we fix the network architecture as $[N_{\mathcal{X}}, N_2, 1000, 1000, 1000, 1000, N_{\mathcal{TC}}]$ where $N_{\mathcal{X}}$ is the observation domain dimension and N_2 is the number of hidden nodes in *GRBM*. For audio modality, $N_{\mathcal{X}} = 195, N_2 = 1000$ and for skeletal modality, $N_{\mathcal{X}} = 594, N_2 = 2000$; $N_{\mathcal{TC}}$ is the output target class. And, in all our experiments, the number of states associated with an individual action $N_{\mathcal{H}_a}$ is chosen as 10 for modeling the states of an action class. Once each individual modality's Deep Belief Network finishes fine tuning, we combine the multi-DBNs and extract their penultimate layers and further run 200 epochs to slightly adjust the weights for each individual modal DBN.

5.4.1.4 Results & Computational Complexity Analysis

We compare our model with the state-of-the-art methods using individual input modals and the baseline multimodal method by averaging the individual modal output confident scores in Table 5.1. It can be seen that both multimodal recognition rates are considerably higher than a single modal input. And the proposed framework of early fusion outperforms the confident score averaging scheme. We further plot the classification rate for each gesture class among different modalities in Fig. 5.4. The bar plot shows the complementary information between two modalities: even when one modality achieves low recognition rate, the multimodal fusion achieves on par with another modality, *e.g.*, gesture 6; when both modalities generate noisy output, the shared multimodal scheme could learn the complementary representation and achieve a superior result, *e.g.*, gesture 15.

With a low inference cost, our framework can perform real-time gesture recognition. Specifically, a single feed forward neural network incurs trivial computation time, linearly in $\mathcal{O}(mT)$ and the complexity of Viterbi algorithm is $\mathcal{O}(T * |S|^2)$ with the number of modalities m, the number of frames T and the state number S.

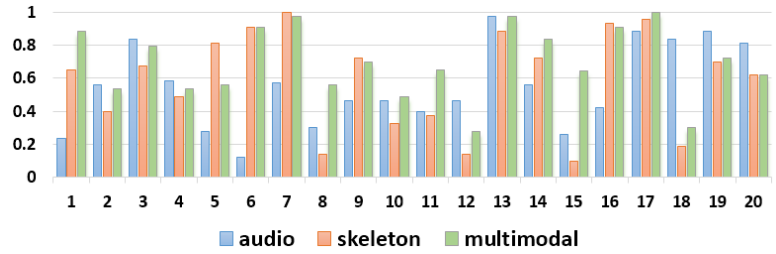


Figure 5.4: Comparison of individual gesture class classification rates among different modalities.

5.4.2 Skeleton-Depth pair module

ChaLearn Italian Gesture Recognition-ChaLearn LAP track: this dataset has the same vocabulary as the 5.4.1. The details of the dataset can be found at 6.1.1.1. Only the skeletal modality and the audio modality are considered. Note that a large number of frames is advantageous for deep neural networks settings over other nonparametric models for estimating skeletal human poses.

The set contains 940 labeled sequences. First 650 sample sequences are used for training and the next 50 sample sequences for validation with the rest 240 for testing where each sequence contains around 20 gestures with some noisy non-meaningful vocabulary tokens.

5.4.2.1 Deep Learning Library: Theano & cuda-convnet

The Deep Belief Network library used in this section is ***Theano*** [108] [1] which is a Python library with efficient handling of multi-dimensional arrays, expressive mathematical manoeuvrer and user-friendly optimization process.

The GPU enabled blazing fast Convolutional Neural Network library used in this section is ***cuda-convnet*** [13] [2] which is a fast C++/CUDA implementation of convolutional neural networks. The very flexible framework enables various connectivity configurations and neuron activation functions. The very efficient juggling between CPU and GPU code structure and GPU programming renders itself one of the fastest CNNs library.

[1] http://deeplearning.net/software/theano/
[2] https://code.google.com/p/cuda-convnet/

5.4.2.2 Skeleton Module

Only upper body joints are relevant to the discriminative gesture recognition tasks. Therefore, only the 11 upper body joints are considered as feature input. The 11 upper body joints used are *"ElbowLeft, WristLeft, ShoulderLeft, HandLeft, ElbowRight, WristRight, ShoulderRight, HandRight, Head, Spine, HipCenter"*.

The 3D coordinates of N joints of frame c are given as: $X_c = \{x_1^c, x_2^c, \ldots, x_N^c\}$. 3D positional pairwise differences of joints [18] are deployed for observation domain $\mathcal{X}$. They capture posture features, motion features by direction concatenation: $\mathcal{X} = [f_{cc}, f_{cp}]$ as demonstrated in the same as Eq 5.2. Note that offset features f_{ci} used in [18] depend on the first frame, if the initialization fails which is a very common scenario, the feature descriptor will be generally very noisy. Hence, the offset features f_{ci} are discarded and only two robust features are $[f_{cc}, f_{cp}]$ kept.

$$f_{cc} = \{x_i^c - x_j^c | i, j = 1, 2, \ldots, N; i \neq j\} \tag{5.2}$$

$$f_{cp} = \{x_i^c - x_j^p | x_i^c \in X_c; x_j^p \in X_p\} \tag{5.3}$$

This results in a raw dimension of $N_{\mathcal{X}} = N_{joints} \times (N_{joints} - 1)/2 + N_{joints}^2) \times 3$ where N_{joints} is the number of joints used. Therefore, in the experiment with $N_{joints} = 11, N_{\mathcal{X}} = 528$.

Hidden states: Force alignment is used to extract the hidden states, *i.e.*, if a gesture token is 100 frames, the first 10 frames are assigned as hidden state ***1*** and the 10-20 frames are assigned as hidden state ***2*** and so on and so forth.

Ergodic states: neutral frames are extracted as 5 frames before or after a gesture tokens labelled by ground truth.

Caveat:

- When extracting any features, the 3D joint coordinates have not been transformed from the world coordinate system to a person centric coordinate system by placing the *"HipCenter"* at the origin.
- Note also that the normalization scheme by scaling the skeleton position using length of *"HipCenter"* and *"Spine"* didn't work well in the implementation.

- The third point worth noting is that some actors performed gestures using their left hand as a dominant hand whereas some using their right hand which will be worth investigating this effect in future research. However, those tokens are treated indiscriminately. Hence, the feature fed into *GRBM* are almost raw, un-preprocessed.

In the training set, there are in total 400, 117 frames. During the training of *DBN*, 90% is used for training, 8% for validation (for the purpose of early stopping) 2% is used for test evaluation.

For high level skeleton feature extraction, two network architectures, *i.e.* a smaller one and a larger one were experimented: $[N_{\mathcal{X}}, 1000, 1000, 500, N_{\mathcal{TC}}]$ and $[N_{\mathcal{X}}, 2000, 2000, 1000, N_{\mathcal{TC}}]$, where $N_{\mathcal{X}} = 528$ is the observation domain dimension; $N_{\mathcal{TC}} = 201$ is the output target class. In all our experiments the number of states associated to an individual action $N_{\mathcal{H}_a}$ is chosen as 10 for modeling the states of an action class. The feed forward networks are pre-trained with a fixed learning rate using *SGD* with a mini-batch size of 200 training examples. We have run 100 epochs for unsupervised pre-training. For Gaussian-binary RBMs, learning rate is fixed at 0.001 while for binary-binary RBMs as 0.01 (note in general, training *GRBM* requires smaller learning rate). For fine-tuning, the learning rate starts at 1 with 0.99999 mini-batch scaling. Maximum fine-tuning epochs is 500 with early stopping strategy and in the experiments, early stopping occurs around 440 epoch. Optimization complete with best validation score (the frame based prediction error rate) of 38.19%, with test performance 38.11%.

We believe further carefully choosing network architecture would lead to more competitive results. However, in order not to "creeping overfitting", we would like to treat the model as the aforementioned more generic approach. Since a utterly natal approach will generally have a tough time challenging against established, delicately fine-tuned approaches at the beginning. More essentially, new born methodologies should have a chance to mature and develop so as not being forced to battle against the top performance.

Post-Processing:

The predicted token less than 20 frames are discarded as noisy tokens. Note that there are many noisy gesture tokens predicted by viterbi decoding. One

way to sift through the noisy tokens is to discard the token path log probability smaller than certain threshold. However, because the metric of this challenge: *Jaccard index* strongly penalizes false negatives, experiments show that it's better to have more false positives than to miss true positives. Effective ways to detect false positives should be an interesting aspect of future works.

The performance of the skeleton module is shown in Tab 5.2. It can be seen that larger net (Net2) will generally perform better than smaller net (Net1), averaging multi-column nets almost will certainly further improve the performance [14]. Hence, in the following experiments, only the multi-column averaging results are reported.

5.4.2.3 Depth 3D Module

Preprocessing & Normalizing: shifting, scaling and resizing
Working directly with raw input Kinect recorded data frames, which are 480×640 pixel images, can be computationally demanding. Deepmind technology [109] presents the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. Similarly, the following basic preprocessing steps are adopted aimed at reducing the input dimensionality from the original 480×640 pixel to 90×90 pixels. The square-sized of the final image is required because the used GPU implementation from [13] expects square inputs and the input channel should be in the set of $[1, 3, 4x]$. Finally, a cuboid of 4 frames, hence, size $90 \times 90 \times 4$, is extracted as a spatio-temporal unit. There are two normalization schemes implemented as Algo. 5 and Algo. 6. Note that the Algo. 5 depends heavily on the provided maximum depth from the recording scene and Algo. 6 depends on the accurate detection of skeleton joins, and both scheme require the performer remains a roughly static position (though the max pooling scheme in 3DCNN to some extend overcome the problem of position shifting). Generally, Algo. 6 is more robust than Algo. 5 because the provided maximum depth can sometimes be very noisy, *e.g.*, Sample0671, Sample0692, Sample0699, *etc*.

Algorithm 5: Normalization scheme 1: template matching

Data:
$\mathbf{T}$ - exemplary template with original scale of size 320×320, (Sample0003 is chosen as the exemplary template, shown in 5.5a).
$\mathbf{R_{depth}}$ - reference depth, fixed to ***1941*** (acquired from the above exemplary template $\mathbf{T}$).
$\hat{\mathbf{T}}$ - test image, as shown in 5.5b.
$\mathbf{M}$ - user foreground segmented mask.

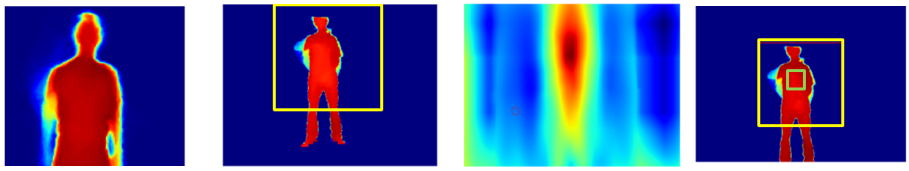
1 Apply a 5×5 aperture median filter to test depth frame $\hat{\mathbf{T}}$ as in [110] to reduce the salt and pepper noise.
2 Multiply test depth frame $\hat{\mathbf{T}}$ with the user segmented mask $\mathbf{M}$: $\hat{\mathbf{T}} = \hat{\mathbf{T}} \times \mathbf{M}$.
3 Template matching test image $\hat{\mathbf{T}}$ with $\mathbf{T}$ using normalized cross-correlation [111], the response score $\mathbf{R}$ is shown in 5.5c.
4 Shift the image according to the maximum response $\mathbf{R}$ to its centre applying affine transformation [112].
5 Scale the image according to reference depth $\mathbf{R_{depth}}$ and the median depth of a bounding box in the centre of the image with 25×25 size as shown as the green boundingp box in 5.5d.
6 Resize the image from 320×320 to 90×90.

Result:
$\tilde{\mathbf{T}}$ - Resize-normalized image shown in the yellow bounding box of 5.5d.

Overall Architecture & Details of Learning

The 3DCNN architecture is specified as Fig. 5.6: the input contextual frames are of size $90 \times 90 \times 4$ substracting the mean activations over all training set from each pixel, the first layer contains 16 maps of $7 \times 7 \times 4$ 3D kernel followed by local response normalization layer [13] and stride 2 max pooling; the second convolutional layers has 32 maps of 5×5 kernel followed by local response normalization layer and stride 2 max pooling; the third convolution layer is composed of 32 maps of 6×6 kernel followed by max pooling; then we have one fully connected layer of size 1000; the output layer is of size $201 = 10 \times 20 + 1$ (number of hidden states for each class$\times$ number of classes plus one ergodic state).

The training set is roughly of 400,000 frames and is divided into 33 mini-batches with first 30 batches for training and the rest 3 batches for validation.



(a) template image (b) test image (c) template response (d) shift-resize image

Figure 5.5: Illustration of normalization scheme 1: template matching.

Algorithm 6: Normalization scheme 2: skeleton normalization

Data:
$\mathbf{S_{spine}}$ - Skeleton Spine joints pixel coordinates.
$\mathbf{S_{shoulder}}$ - Skeleton Shoulder joints pixel coordinates.
$\hat{\mathbf{T}}$ - test image.
$\mathbf{M}$ - user foreground segmented mask.
$\mathbf{R_{length}}$ - reference length of shoulder to spine, fixed to ***100*** (1 meter).

1 Apply a 5×5 aperture median filter to test depth frame $\hat{\mathbf{T}}$.
2 Multiply test depth frame $\hat{\mathbf{T}}$ with the user segmented mask $\mathbf{M}$.
3 Shift the image according to the centroid of Spine joint $\mathbf{S_{spine}}$.
4 Scale the image according to the $\mathbf{R_{length}}/(\mathbf{S_{spine}} - \mathbf{S_{shoulder}})$.

Result:
$\tilde{\mathbf{T}}$ - Resize the shifted-scaledp image to 90×90 .

Standard *SGD* is run for the first 100 epochs with learning rate of 0.1 and the weight learning rate as 0.001 and weight bias learning rate 0.002 both momentum are fixed as 0.9, weight decay is fixed to 0.0005, the next 100 epochs with $0.1\times$learning rate. Another network trained by randomly cropping 82×82 pixels on the flight as [13] is also implemented to enhance the model's robustness. During the test time, the centre part and other 4 corner parts are averaged to obtain the final score, *c.f.* Fig 5.9c. Due to the time constraint, only 150 epochs are trained with the learning rate reduced to one tenth at the 92nd epoch. The training frame based classification error for the aforementioned two networks are shown in 5.9a and 5.9b. One interesting observation is that for the network with uncropped input, reducing the learning rate at 100 epoch, the frame-based classification rate reduces drastically whereas for the network with cropped input, reducing the learning rate results in a spike increase of

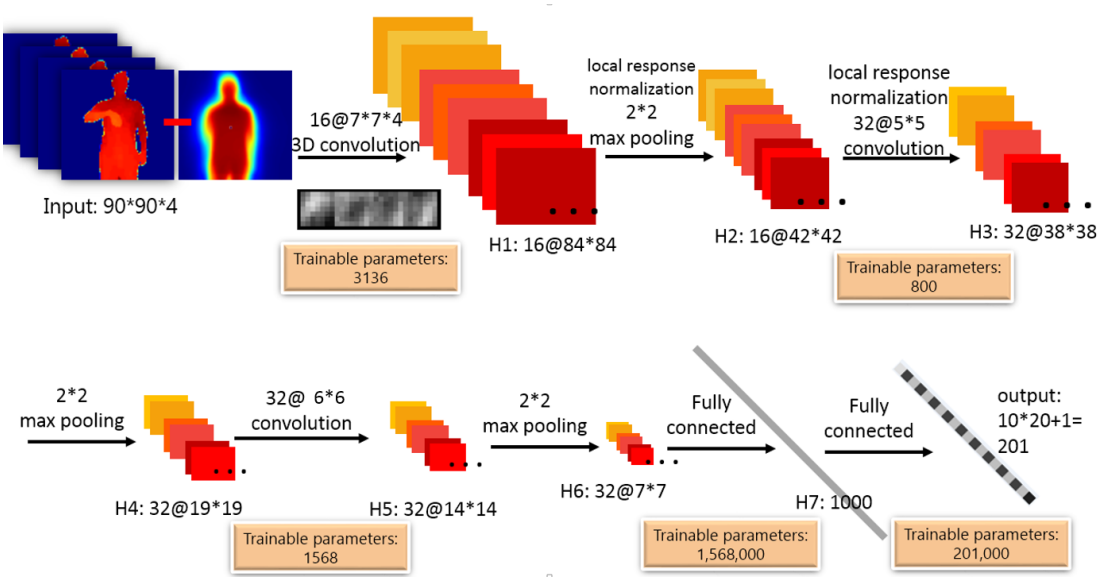


Figure 5.6: An illustration of the architecture of the 3DCNN architecture.

frame-based classification error rate. The reason for this discrepancy is worth further investigation.

Looking into the networks-visualization of filter banks

The weight filters of the first *conv1* layer are illustrated in Fig 5.7 and it can be seen that both shape pattern filters and motion filters are learnt effectively. Interestingly, the 3DCNN is able to learn the most informative motion part of the body effectively (highest response parts are the arms/hands areas), albeit no signal was explicitly given during training instructing which body parts the gesture recognition tasks should focus on.

5.4.2.4 Score Fusion

To fuse the dual model prediction, the bottom strategy of Fig.5.1 is adopted. The complementary properties of both modules can be seen in Fig.5.10. Note that the skeleton module generally performs better than the depth module, one reason could be that the skeleton joints are learnt from [19] and one key component of their success lies in the huge and highly varied training data: both from realistic and synthetic depth images, a total number of 1 million images were used to train the deep randomized decision forest classifier in

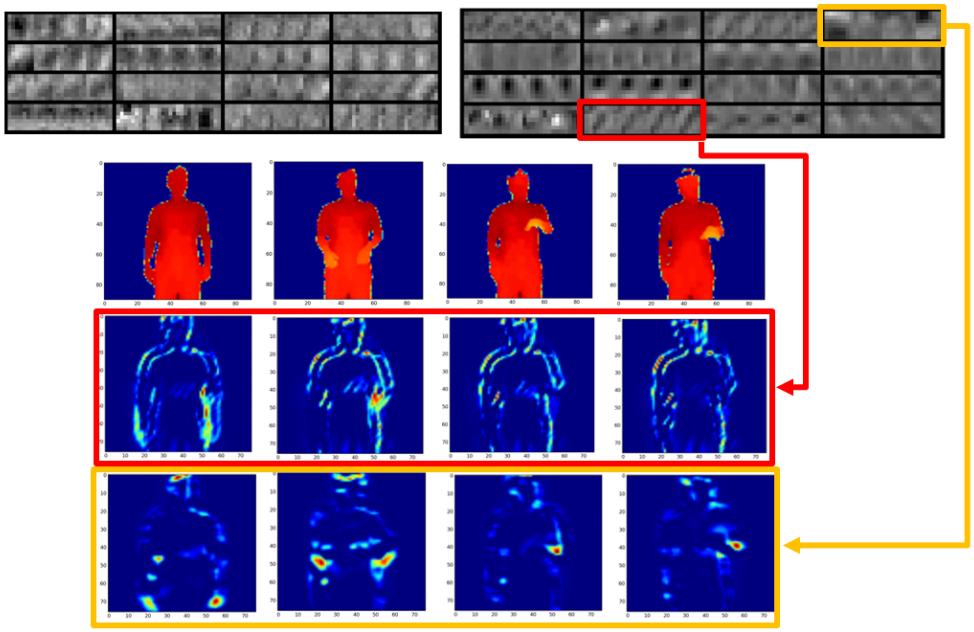


Figure 5.7: Top left: the *conv1* weights of the 3DCNN learnt with uncropped input; top right: the *conv1* weights of the 3DCNN learnt with cropped input. It can be seen that filters/weights of the cropped input trained networks are smoother. Bottom: visualization of sample frames after *conv1* layer (Sample0654, 264-296 frames, sampled every 8 frames). It can be seen that the filters of the first convolutional layer are able to learn both shape pattern(red bounding box) and motion(yellow bounding box). Also note that the high response maps correspond to the most informative part of the body, even though during the training process, all local patches are learned indiscriminately regardless of its location.

order to avoid overfitting. Hence skeleton data are more robust.

5.5 Discussion

Hand-engineered, task-specific features are often less adaptive and time-consuming to design. This difficulty is more pronounced with multimodal data as the features have to relate multiple data sources. In this chapter, a framework that utilizes Deep Neural Networks for modeling emission probabilities at frame-level was proposed, and two schemes for shared representation learning from multimodal sensory inputs were experimented. The framework can be used to extract a unified representation that fuses various modalities together for model-

Algorithm 7: Multimodal Deep Dynamic Networks – training

Data:

$\mathbf{X^1} = \{\mathbf{x_i^1}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - raw input(skeletal) feature sequence.

$\mathbf{X^2} = \{\mathbf{x_i^2}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - raw input(depth) feature sequence in the form of $M_1 \times M_2 \times T$, where M_1, M_2 are the height and width of the input image and T is the number of contiguous frames of the spatio-temporal cuboid.

Note that the GPU library *cuda-convnet* [13] used requires square size images and T is a multiple of 4.

$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}...\mathbf{t}]}$ - frame based local label (achieved by semi-supervised forced-aligment),
where $\mathbf{y_i} \in \{C * S + \boldsymbol{1}\}$ with C is the number of class, S is the number of hidden states for each class, $\boldsymbol{1}$ as ergodic state.

1 **for** $m \leftarrow 1$ *to* 2 **do**
2 **if** m *is* 1 **then**
3 Preprocessing the data $\mathbf{X^1}$ as in Eq.5.2.
4 Normalizing(zero mean, unit variance per dimension) the above features and feed to to Eq.3.17.
5 Pre-training the networks using *Contrastive Divergence* 3.16.
6 Supervised fine-tuning the Deep Belief Networks using $\mathbf{Y}$ by standard mini-batch *SGD* backpropagation 3.21.
7 **else**
8 Preprocessing the data $\mathbf{X^2}$ (normalizing, median filtering the depth data) Algo.5 or Algo.6.
9 Feeding the above features to Eq.4.2.
10 Supervised fine-tuning the Deep 3D Convolutional Neural Networks using $\mathbf{Y}$ by standard mini-batch *SGD* Backpropagation.

Result:

GDBN - a gaussian bernoulli visible layer Deep Belief Network to generate the emission probabilities for hidden markov model.

3DCNN - a 3D Deep Convolutional Neural Networks to generate the emission probabilities for hidden markov model.

$\mathbf{p(H_1)}$ - prior probability for $\mathbf{Y}$.

$\mathbf{p(H_t|H_{t-1})}$ - transition probability for $\mathbf{Y}$, enforcing the beginning and ending of a sequence can only start from the first or the last state.

Algorithm 8: Multimodal Deep Dynamic Networks – test

Data:
$\mathbf{X^1} = \{\mathbf{x_i^1}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input(skeletal) feature sequence.
$\mathbf{X^2} = \{\mathbf{x_i^2}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - raw input(depth) feature sequence in the form of $M \times M \times T$.
GDBN - a gaussian bernoulli visible layer Deep Belief Network to generate the emission probabilities for hidden markov model.
3DCNN - the trained 3D Deep Convolutional Neural Networks to generate the emission probabilities for hidden markov model.
$\mathbf{p}(\mathbf{H_1})$ - prior probability for $\mathbf{Y}$.
$\mathbf{p}(\mathbf{H_t}|\mathbf{H_{t-1}})$ - transition probability for $\mathbf{Y}$.

for $m \leftarrow 1$ *to* 2 **do**
  if m *is* 1 **then**
    Preprocessing and normalizing the data $\mathbf{X^1}$ as in Eq.5.2.
    Feedforwarding network **GDBN** to generate the emission probability $\mathbf{p}(\mathbf{X_t}|\mathbf{H_t})$ in Eq.3.25.
    Generating the score probability matrix $\mathbf{S^1} = \mathbf{p}(\mathbf{H_{1:T}}, \mathbf{X_{1:T}})$.
  else
    Preprocessing the data $\mathbf{X^2}$ (normalizing, median filtering the depth data) Algo.5 or Algo.6.
    Feedforwarding **3DCNN** to generate the emission probability $\mathbf{S^2} = \mathbf{p}(\mathbf{X_t}|\mathbf{H_t})$ in Eq.3.25.
    Generating the score probability matrix $\mathbf{S^2} = \mathbf{p}(\mathbf{H_{1:T}}, \mathbf{X_{1:T}})$.
Fusing the score matrix $\mathbf{S} = \mathbf{S^1} + \mathbf{S^2}$.
Finding the best path $\mathbf{V_{t,\mathcal{H}}}$ using $\mathbf{S}$ by Viterbi decoding as in Eq.3.26.

Result:
$\mathbf{Y} = \{\mathbf{y_i}\}_{\mathbf{i}\in[\mathbf{1}\ldots\mathbf{t}]}$ - frame based local label
where $\mathbf{y_i} \in \{C * S + \boldsymbol{1}\}$ with C is the number of class, S is the number of hidden states for each class, $\boldsymbol{1}$ as ergodic state.
C - global label, the anchor point is chosen as the middle state frame.

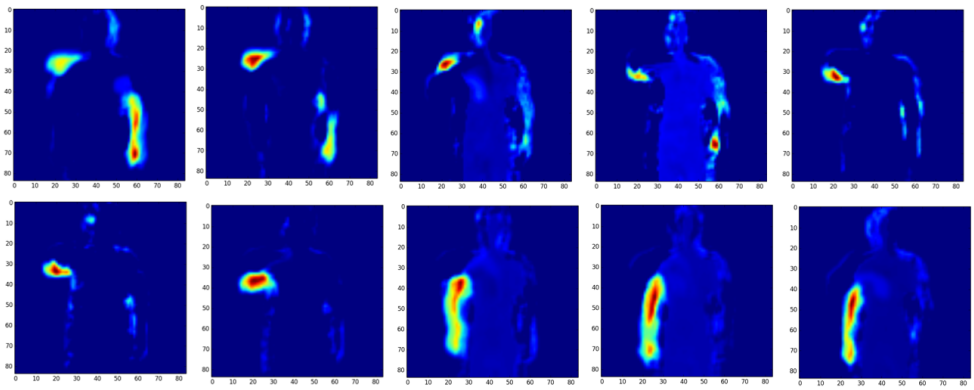
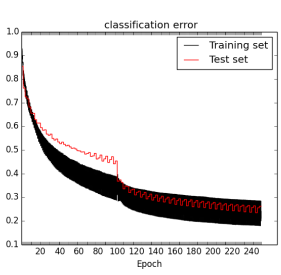
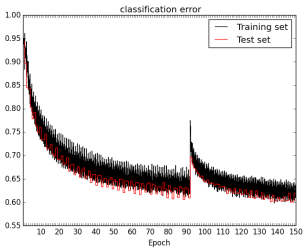


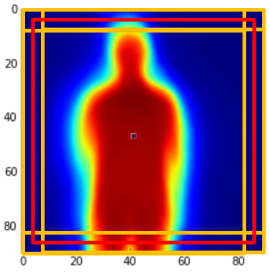
Figure 5.8: More illustrations of the middle level features from the activation images after first convolutional layer. High response arms and hands areas are learnt automatically without explicit learning signal in term of location information.



(a) Frame based classification error with uncropped input.



(b) Frame based classification error with cropped input.



(c) Cropped images to enhance model's robustness.

Figure 5.9: Visualization of the first filters and training statistics for 3DCNN.

ing time series data. The experimental results on bi-modal time series data, *i.e.*, audio and skeletal joints data, show that the multimodal DBN+HMM framework can learn a good model of the joint space of multiple sensory inputs, and is consistently as good as/better than the unimodal input. The proposed model also outperforms the traditional late fusion scheme, opening the door for exploring the complementary representation among multimodal inputs. It also suggests that learning features directly from data is a very important research direction and the learning-based methods are not only more generalizable to many domains, but also are powerful in combining with other well-studied probabilistic graphical models for modeling and reasoning dynamic

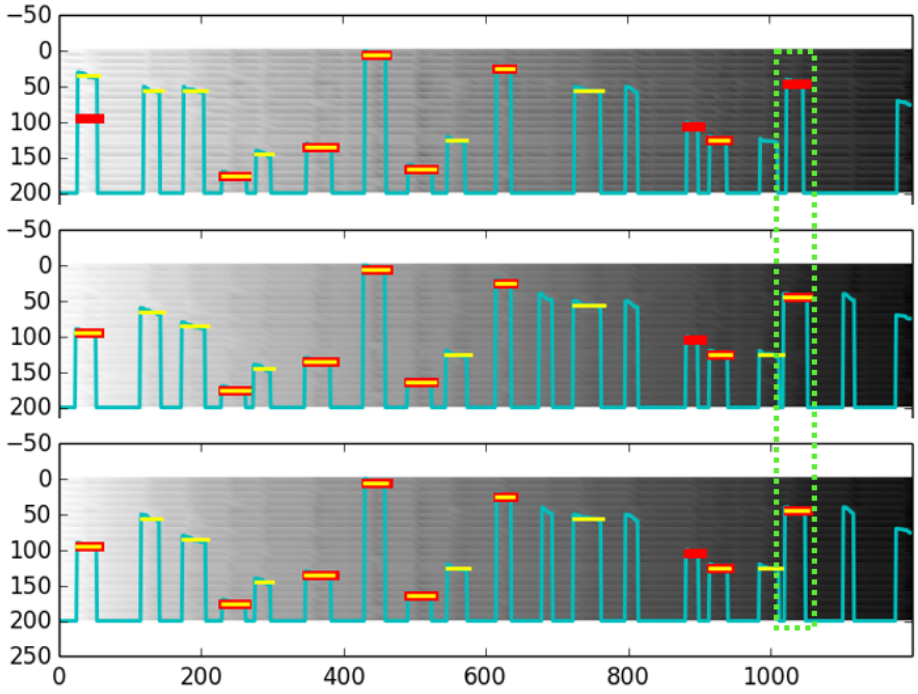


Figure 5.10: Viterbi decoding of two modules and their fusion result of sample sequence 704. Top to bottom: skeleton, depth, score fusion with x-axis representing the time and y-axis representing the hidden states of all the classes with the ergodic state at the bottom. Red lines are the ground truth label, cyan lines are the viterbi shortest path and yellow lines are the predicted label. There are some complementary information of the two modules and generally skeletal module outperforms the depth module. The fusion of the two could exploit the uncertainty, *e.g.* light green dashed box indicates that depth module makes the correct prediction whereas the skeletal module fails, the combined module is still making the correct prediction.

sequences. The heterogeneous inputs from skeleton and depth images require different feature learning methods and the late fusion scheme is adopted at the score level. Future works include learning the share representation at the penultimate layer and backpropagation the gradient in the share space.

Module \ Evaluation Set	Validation	Test
Skeleton–DBDN Net1	0.7468	-
Skeleton–DBDN Net2	0.8017	-
Skeleton–DBDN MultiNet	0.8236	0.7873
Depth–3DCNN Norm1 5	0.6378	-
Depth–3DCNN Norm2 6	0.6924	0.6371
Score Fusion	0.8045	0.8162

Table 5.2: Comparison of results in terms of Jaccard index between different network structures and various modules. DBDN Net1 corresponds to network structure of $[528, 1000, 1000, 500, 201]$ and DBDN Net2 $[528, 2000, 2000, 1000, 201]$, DBDN MultiNet is the average of 3 Nets (2 Net1 and 1 Net2 with different initializations). It can be seen that larger net has better performance and multi-column net will further improve the classification rate. Norm1 corresponds to the normalization Algo.5 and Norm2 corresponds to the Algo.6.

Team	Modalities	Features	Fusion	Classifier	Score
LIRIS	S,D,RGB	RAW, skeleton joints	Early	Deep neural network	0.8500
CraSPN	S,D,RGB	HOG, skeleton (BoW)	Early	Adaboost	0.8339
JY	S,RGB	HOG, skeleton (PCA)	Late	HMM	0.8268
Proposed method	S,D	Raw, skeleton	Late	DNN-HMM	***0.8162***
CUHK-SWJTU	RGB	Improved dense trajectories	-	Fisher Vector,VLAD	0.7919
lpigou	D,RGB	RAW	Early	CNN	0.7888
ismar	S	-	-	RF	0.7466
Quads	S	Fisher Vector	-	SVM	0.7454
Telepoints	S,D,RGB	STIP, Skeleton	Late	SVM	0.6888
TUM-fortiss	S,D,RGB	STIP, skeleton	Late	SVM, RF	0.6490
CSU-SCM	S,D,RGB	HOG, skeleton	Late	SVM, HMM	0.5972
iva.mm	S,D,RGB	HOG, skeleton (BoW)	Late	SVM, HMM	0.5563
Terrier	S	-	-	RF	0.5390
Team Netherlands	S,D,RGB	MHI, LPP	Early	SVM, Regression Trees	0.4307

Table 5.3: Performance comparison with various methods using various modules. S stands for skeletal modality, D stands for depth modality and RGB stands for RGB modality. The proposed method performs competitively. The first ranked method uses early fusion for multi-modal data.

Chapter 6

Conclusion and Future Directions

In this thesis, an introduction to the field of human action recognition using various input modules is presented.

The first part of the thesis presented various hand-crafted features starting from manifold learning and dimensionality reduction of the silhouettes, to bag-of-correlated-poses encoding temporal correlation between poses with soft-assignment scheme. The unique property of depth images were also discussed under the one-shot-learning scenario. A multi-view spectral embedding technique was adopted to embed different modules, *i.e.* RGB and depth images, into a smooth manifold.

In the second parts of this thesis, *i.e.* chapter 3 to chapter 5, which comprise the major contributions of the thesis, the data-driven approach is pursuit. Chapter 3 introduced basic notion of RBMs and DBNs which constitute the building blocks for estimating emission probabilities for HMM. This weakly-supervised scheme outperforms the traditional GMM+HMM paradigm. And by introducing an ergodic state during training, the proposed framework is able to segment and recognize action sequence simultaneously. Chapter 4 extended the aforementioned framework to image domain. 3DCNN was proposed in the place of DBN for estimating the emission probability of Markov Field. It also demonstrated the flexible framework for structure prediction under the CRF paradigm. Chapter 5 unified multimodal dynamic neural networks with various sensory inputs. Both early fusion and later fusion schemes were conducted and both shown their effectiveness in fusing multimodal in-

puts. The 3DCNN was also studied in more details, exhibiting some interesting shape and motion pattern learnt automatically by the networks which helps better interpreting the internal systems of the networks.

Open Questions and Future Directions

This thesis starts with the introduction of hand crafted features for video representation, then provides ground work using Deep Neural Networks as a feature extraction technique for action recognition. There are several potential, unexplored new frontiers, applications and extensions of the thoughts presented in this thesis, particularly related to various graphical models to represent high-dimensional time series models.

Bridging the gap between hand-crafted features and feature learning. For the past decade, researchers spent tremendous efforts in designing hand crafted features for video representation and until very recent, using deep learning technique for video representation is gaining momentum. The very recent challenge for gesture recognition in Tab.5.3 shows the learning-based approach has modest gain over traditional task-specific, hand-tuned features. However, could all those efforts spent in designing hand-crafted features help better initialize the network and better understand the learnt networks? What's the the quantitative fine-line of a big enough dataset that the notion using hand-crafted feature should be discarded in the case that learning-based approach is omnipotent in learning better features? Ji *et al.* [16] used multiple sets of hand crafted features for CNN initialization, the effectiveness of which, is worth further investigation.

Unsupervised learning and transfer feature learning. Even though most recent visual recognition challenges winners are all supervised based discriminative approaches, the vast quantities of real-world digital data are unlabelled and have been kept untapped. The field of unsupervised learning and transfer learning where the middle level features are reused are gaining more attention. Convolutional Auto-Encoder (CAE) [102] is one approach for unsupervised learning, however, most works were

done within a relatively small scale. In the recent work of [113], CNN was utilized to classify 1 million YouTube videos belonging to 487 classes, and the more generic features on the bottom of the network (such as edges, local shapes) are fixed and only the top 3 layers (dataset-specific) are re-trained for the UCF-101 dataset, resulting better recognition than the networks trained from scratch. One could utilize this powerful network trained on such a large scale dataset and benefit smaller datasets with limited labels.

Deep Reinforcement Learning. Reinforcement learning (RL) offers a powerful set of tools for sequential decision making uncertainty. It lies in the intersection of multidisciplinary studies of machine learning, optimal control, operational research, etc. The mapping function is required to learn to make sequences of decision and is evaluated by the long-term quality of its choices. RL stands out from other machine learning paradigms in that: there is no supervisor, only a reward signal; feedback is delayed, not instantaneous; input data are not *i.i.d.*, *i.e.* time really matters and agent's actions affect the subsequent data it receives. RL has been successfully applied to control helicopters, robotics, playing Atari games [109]. The convergence of RL and DL could assist new applications in terms of human action recognition in the relevance feedback system. The matrimony of both could assist new applications or new insights from both theoretical analysis and empirical studies perspectives.

Learning the high level temporal correlation as a unified framework. It is worth noting that the high level temporal encoding in this thesis is fixed during the training stage, *i.e.* transitional probability and prior probability are kept constant as the statistics collecting from the training data. One could combine the entire system into a unified Deep Recurrent Neural Network(RNNs) (Fig. 6.1) and backpropagating the error gradient from the highest hierarchy. RNNs are compelling because they have a high-dimensional nonlinear dynamics hidden states that permit the networks to encode and operate on past information. Long Short-Term Memory Recurrent Networks (LSTM)[114] has been introduced to ease the "van-

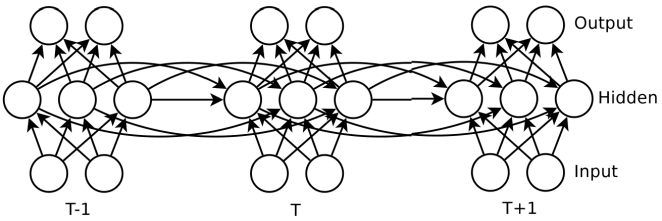


Figure 6.1: A Recurrent Neural Network[94] is a very deep feedforward neural network whose weights are shared across time.

ishing or exploding gradient problems". There is still provision for powerful enough hardware and fast enough training system because Recurrent Neural Networks are notoriously difficult to train[115]. However, unifying the system could potentially make the Deep Dynamic Networks more flexible in encoding many time steps.

Videos and other high-dimensional time series data are challenging areas where learning based techniques are gaining more and more momentums. Still there are many more broad and open questions. Several potential research directions have been outlined. Despite the burgeoning successes using deep neural networks for video sequence analysis, many interesting and important problems in unsupervised learning, transfer learning, unified learning remain. It is believe that further efforts to study these problems will make another step toward true Artificial Intelligent Systems.

Bibliography

[1] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, 2005. 1, 72

[2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, oct. 2005, pp. 65 – 72. 1, 6, 9, 10, 11, 18, 26, 72

[3] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *European Conference on Computer Vision*. Springer, 2008. 1, 72

[4] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *ACM International Conference on Multimedia*, 2007. 1, 72

[5] A. Klaser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients," in *British Machine Vision Conference*, 2008. 1, 72, 73

[6] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, 2013. 1, 72

[7] H. Wang, M. Ullah, A. Klaser, I. Laptev, C. Schmid *et al.*, "Evaluation of local spatio-temporal features for action recognition," 2009. 1, 6, 18, 23, 26, 72

[8] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *European Conference on Computer Vision*. Springer, 2010. 1, 72

[9] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1, 2, 63, 72

[10] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Spatio-temporal convolutional sparse auto-encoder for sequence classification," in *British Machine Vision Conference*, 2012. 1, 2, 63, 72

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, 2006. 2, 44, 72

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *arXiv preprint arXiv:1404.7828*, 2014. 2

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012. 2, 62, 63, 66, 67, 72, 88, 91, 92, 93, 96

[14] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 2, 63, 72, 91

[15] M. Y. Shuiwang Ji, Wei Xu and K. Yu, "3d convolutional neural networks for human action recognition," in *International Conference on Machine Learning*. IEEE, 2010. 2, 63, 65, 72

[16] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 2, 65, 66, 72, 78, 102

[17] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, 2012. 2, 44, 46, 47, 54, 56, 72, 82, 84, 86

[18] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2, 82, 84, 86, 89

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011. 3, 21, 27, 34, 94

[20] S. Escalera, J. Gonzlez, X. Bar, M. Reyes, O. Lops, I. Guyon, V. Athitsos, and H. J. Escalante, "Multi-modal gesture recognition challenge 2013: Dataset and results." in *ACM ChaLearn Multi-Modal Gesture Recognition Grand Challenge and Workshop*, 2013. [Online]. Available: http://gesture.chalearn.org/ 3, 47, 57, 84, 120

[21] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin, "Instructing people for training gestural interactive systems," in *ACM CHI*, 2012. 3, 47, 52, 56, 59, 60, 86, 122

[22] I. Guyon, V. Athitsos, P. Jangyodsuk, B. Hamner, and H. J. Escalante, "Chalearn gesture challenge: Design and first results," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012. 3, 47

[23] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 3, 47, 53, 58, 59, 122

[24] I. Laptev and T. Lindeberg, "Space-time interest points," in *IN ICCV*, 2003, pp. 432–439. 6, 9, 10, 11, 18, 26

[25] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," *Computer Vision–ECCV 2008*, pp. 650–663, 2008. 6, 18

[26] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, 2005, pp. 524–531. 6

[27] T. Goodhart, P. Yan, and M. Shah, "Action recognition using spatio-temporal regularity based features," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 745–748. 6

[28] A. Klaser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*. Citeseer, 2008. 6, 9, 18, 26

[29] M. Chen and A. Hauptmann, "Mosift: Recognizing human actions in surveillance videos," *Transform*, pp. 1–16, 2009. 6

[30] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *British Machine Vision Conference*, 2009. 6

[31] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2929–2936. 6, 18, 23

[32] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8. 6

[33] J. Davis and A. Bobick, "The representation and recognition of human movement using temporal templates," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 928–934. 7, 14

[34] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1395–1402. 7, 9, 15, 16, 124

[35] L. Wang and C. Leckie, "Encoding actions via the quantized vocabulary of averaged silhouettes," in *Proceedings of International Conference on Pattern Recognition*, 2010, pp. 3657–3660. 7

[36] L. Shao and X. Chen, "Histogram of body poses and spectral regression discriminant analysis for human action categorization," in *Proceedings of the British Machine Vision Conference (BMVC). Aberystwyth, UK*, 2010. 7, 19, 20

[37] H. Qu, L. Wang, and C. Leckie, "Action recognition using space-time shape difference images," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 3661–3664. 7

[38] X. Sun, M. Chen, and A. Hauptmann, "Action recognition via local descriptors and holistic features," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*. IEEE, 2009, pp. 58–65. 7, 8, 14, 15, 20, 23

[39] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 17–32. 8

[40] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 7, pp. 1271–1283, 2010. 8, 11, 26

[41] Y. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2559–2566. 8

[42] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, and T. Huang, "Novel gaussianized vector representation for improved natural scene categorization," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 702–708, 2010. 8

[43] L. Shao, D. Wu, and X. Chen, "Action recognition using correlogram of body poses and spectral regression," in *International Conference on Image Processing*, 2011. 12

[44] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 762–768. 12

[45] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36. 15, 21

[46] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006. 15, 16, 124

[47] D. Weinland, M. Özuysal, and P. Fua, "Making action recognition robust to occlusions and viewpoint changes," *Computer Vision–ECCV 2010*, pp. 635–648, 2010. 19, 23, 77

[48] M. Varma and B. Babu, "More generality in efficient multiple kernel learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1065–1072. 19

[49] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 489–496. 19

[50] I. Junejo, E. Dexter, I. Laptev, P. Pérez *et al.*, "Cross-view action recognition from temporal self-similarities," 2008. 19

[51] S. Hadfield and R. Bowden, "Kinecting the dots: Particle based scene flow from depth sensors," in *In Proceedings, International Conference on Computer Vision*, 2011. 21

[52] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Human activity detection from rgbd images," in *AAAI workshop on Pattern, Activity and Intent Recognition (PAIR)*, 2011. 21

[53] "Chalearn gesture dataset," *CGD2011, ChaLearn, California*, 2011. 21, 22, 24, 30, 32

[54] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali, "The american sign language lexicon video dataset," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, 2008. 21

[55] S. Park, S. Yu, J. Kim, S. Kim, and S. Lee, "3d hand tracking using kalman filter in depth space," *EURASIP Journal on Advances in Signal Processing*, 2012. 21, 24

[56] M. Ryoo and J. Aggarwal, "Stochastic representation and recognition of high-level group activities," *International journal of computer vision*, 2011. 22

[57] D. Zhang, F. Wang, Z. Shi, and C. Zhang, "Interactive localized content based image retrieval with multiple-instance active learning," *Pattern Recognition*, 2010. 22

[58] D. Weinland, R. Ronfard, and E. Boyer, "Automatic discovery of action taxonomies from multiple views," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006. 22

[59] T. Kadir, R. Bowden, E. Ong, and A. Zisserman, "Minimal training, large lexicon, unconstrained sign language recognition," in *Proc. BMVC*, 2004. 22

[60] J. Zieren and K. Kraiss, "Robust person-independent visual sign language recognition," *Pattern Recognition and Image Analysis*, 2005. 22

[61] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008. 23

[62] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 2007. 23

[63] D. Tran and A. Sorokin, "Human activity recognition with metric learning," *Computer Vision–ECCV 2008*, 2008. 23

[64] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 2010. 23, 30, 31, 32

[65] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, 1975. 24

[66] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005. 25

[67] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005. 25

[68] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2011. 26

[69] L. Breiman, "Random forests," *Machine learning*, 2001. 27

[70] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2001. 27

[71] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, 2007. 30, 32

[72] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, 2006. 41, 48

[73] R. M. Neal and G. E. Hinton, "A view of the em algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*. Springer, 1998. 41

[74] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007. 41

[75] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 43, 70, 75

[76] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," in *Advances in neural information processing systems*, 2006. 44

[77] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3d discriminative skeletal features for human action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013. 45, 49, 53, 56, 86

[78] M. Müller and T. Röder, "Motion templates for automatic classification and retrieval of motion capture data," in *SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006. 45, 49, 53, 56, 86

[79] S. Nowozin and J. Shotton, "Action points: A representation for low-latency online human action recognition," Tech. Rep., 2012. 45, 48, 49, 51, 53, 54, 56, 60, 86

[80] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, 2013. 45, 49, 53, 56, 59, 86

[81] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006. 46

[82] K. P. Murphy, *Machine learning: a probabilistic perspective*. The MIT Press, 2012. 46, 57, 118

[83] A. Lehrmann, P. Gehler, and S. Nowozin, "A non-parametric bayesian network prior of human pose," in *International Conference on Computer Vision*, 2013. 49

[84] A. Yao, J. Gall, and L. Van Gool, "Coupled action recognition and pose estimation from multiple views," *International journal of computer vision*, 2012. 53, 56

[85] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *European Conference on Computer Vision*. Springer, 2006. 53, 56

[86] T. Do, T. Arti *et al.*, "Neural conditional random fields," in *International Conference on Artificial Intelligence and Statistics*, 2010. 54

[87] X. Yang and Y. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2012. 55, 57, 58, 85

[88] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012. 56

[89] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 56

[90] J. Wu, J. Cheng, C. Zhao, and H. Lu, "Fusing multi-modal features for gesture recognition," in *ACM International Conference on Multimodal Interaction*, 2013. 57

[91] D. N. Lawrence, "Gaussian process models for visualisation of high dimensional data," in *Advances in Neural Information Processing Systems*, 2004. 58

[92] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, 2008. 58

[93] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 58

[94] J. Martens and I. Sutskever, "Learning recurrent neural networks with hessian-free optimization," in *International Conference on Machine Learning*, 2011. 59, 104

[95] M. Müller, A. Baak, and H.-P. Seidel, "Efficient and robust annotation of motion capture data," in *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 59

[96] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola Jr, and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition," *International Journal of Computer Vision*, 2013. 59

[97] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online human gesture recognition from motion data streams," in *ACM International conference on Multimedia*, 2013. 59, 60

[98] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *International Conference on Machine Learning*, 2011. 64, 73, 74, 84

[99] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," p. 22782324, 1998. 66, 72

[100] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International Conference on Machine Learning*, 2001. 69

[101] J. M. Mooij, "libDAI: A free and open source C++ library for discrete approximate inference in graphical models," *Journal of Machine Learning Research*, vol. 11, pp. 2169–2173, Aug. 2010. [Online]. Available: http://www.jmlr.org/papers/volume11/mooij10a/mooij10a.pdf 71

[102] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Artificial Neural Networks and Machine Learning*. Springer, 2011. 71, 102

[103] I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey, "Extraction of visual features for lipreading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 73, 74, 124

[104] G. Zhao, M. Barnard, and M. Pietikainen, "Lipreading with local spatiotemporal descriptors," *IEEE Transactions on Multimedia*, 2009. 73, 74

[105] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," in *European Conference on Computer Vision*. Springer, 2012. 76, 123

[106] O. Oreifej, Z. Liu, and W. Redmond, "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 77

[107] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines." in *Neural Information Processing Systems*, 2012. 84

[108] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, Jun. 2010, oral Presentation. 88

[109] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013. 91, 103

[110] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from rgbd images," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 7–12. 92

[111] J. Lewis, "Fast normalized cross-correlation," in *Vision interface*, vol. 10, no. 1, 1995, pp. 120–123. 92

[112] G. Bradski, *Dr. Dobb's Journal of Software Tools*. 92

[113] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 103

[114] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. 103

[115] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," in *International Conference on Machine Learning*, 2014. 104

Notation

We follow the notation system used in [82] to have a single, consistent notation to cover the wide variety of data, models and algorithms in a unified notation.

General math notation

Symbol	Meaning
f	function
$\mathscr{F}$	function set
∇	vector of first derivatives
$\mathscr{L}$	log-likehood
l	loss function
$\mathcal{F}$	free energy
Z	partition function
E	expectation

Table 6.1: Notation

List of commonly used abbreviations

Symbol	Meaning
iid	Independently and Identically Distributed
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
RF	Random Forest
HOG	Histogram of Oriented Gradient
SIFT	Scale-invariant Feature Transform
MHI	Motion History Image
GEI	Gait Energy Information
INV	Inversed Recording
STIP	Spatio-Temporal Interest Points
LPP	Locality Preserving Projection
HMM	Hidden Markov Model
CRF	Conditional Random Field
RBM	Restricted Boltzmann Machine
GRBM	Gaussian Bernoulli Restricted Boltzmann Machine
DBN	Deep Belief Networks
CNN	Convolutional Neural Networks

Table 6.2: Notation

Appendix

6.1 Details of the Dataset

6.1.1 ChaLearn Italian Gesture Recognition

6.1.1.1 Kaggle track

This dataset[1] is on "multiple instance, user independent learning" [20] of gestures. There are 20 Italian cultural/anthropological signs, *i.e.,vattene, vieniqui, perfetto, furbo, cheduepalle, chevuoi, daccordo, seipazzo, combinato, freganiente , ok, cosatifarei, basta, prendere, noncenepiu, fame, tantotempo, buonissimo, messidaccordo, sonostufo.* In this track, there are four modules, *i.e.*, audio, skeleton, RGB, depth, are provided. However, only the skeletal modality and audio modality are considered in the experiments. We use the subset where the label data are provided during our evaluation process. The set contains 393 labeled sequences with a total of 7754 gestures. We used 350 sequences for training and the rest 43 sequences for testing, where each sequence contains 20 unique gestures. In the training set, there are in total 339,700 frames (20 fps). An illustration of the RGB, depth (with user segmentation) and skeletal modalities is shown in Fig 6.2.

6.1.1.2 ChaLearn Looking At People (LAP) track 3

In this dataset[2] more than 14,000 gestures are drawn from a vocabulary of 20 Italian sign gesture categories as the same in 6.1.1.1 and Fig 6.2. This track is

[1] https://www.kaggle.com/c/multi-modal-gesture-recognition
[2] http://gesture.chalearn.org/homewebsourcereferrals

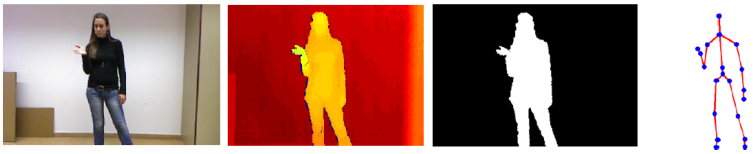


Figure 6.2: ChaLearn Italian Gesture Recognition.

an expansion of the Kaggle track with 650 sample sequences for training and validation and 240 sample sequences for testing. The focus of this track is on multi-modal automatic learning performed by different users, with the goal of achieving user independent continuous gesture segmentation and recognition. The evaluation criterion for this track is the *Jaccard* index (overlap) on a frame-to-frame basis.

$$J(A,B) = \frac{A \bigcap B}{A \bigcup B} \tag{6.1}$$

In this track, only skeletal modality and the depth modality (with user segmentation) are considered.

6.1.2 ChaLearn Gesture One-shot-learning Recognition

This dataset[1] selects "lexicons from nine categories corresponding to various settings or application domains; they include (1) body language gestures (like scratching your head, crossing your arms), (2) gesticulations performed to accompany speech, (3) illustrators (like Italian gestures), (4) emblems (like Indian Mudras), (5) signs (from sign languages for the deaf), (6) signals (like referee signals, diving signals, or mashalling signals to guide machinery or vehicle), (7) actions (like drinking or writing), (8) pantomimes (gestures made to mimic actions), and (9) dance postures." In this track, there are two modules, *i.e.*, RGB, depth, are provided (not that even though recorded by Kinect, no skeleton data are provided because the recording distance is not always within effective distance required by Kinect skeleton detection). Initially 20 development batches are provided and each batch "includes 100 recorded gestures grouped in sequences of 1 to 5 gestures performed by the same user. The gestures are drawn from a small vocabulary of 8 to 15 unique gestures, which we call a lexicon".

[1] http://gesture.chalearn.org/dissemination/cvpr2012

Figure 6.3: ChaLearn Gesture One-shot-learning Recognition.

In total, there are 50,000 gestures captured with Kinect with RGB and depth images of size 240×320 pixels at 10 frames per second. A visual illustration of a single frame is shown as Fig. 6.3.

The evaluation criterion for this track is the "Levenshtein distance" (or "edit distance"): a string metric for measuring the difference between two sequences with minimum number of insertions, deletions or substitutions.

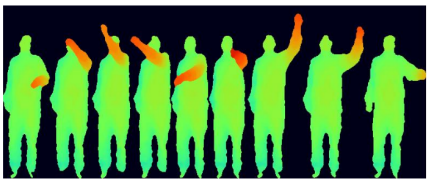
6.1.3 MSR Action3D

MSR Action3D dataset [23][1] is an action dataset of depth sequences captured by Kinect. This dataset contains twenty actions:" *high arm wave, horizontal arm wave,hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up & throw.*" (*c.f.* Fig.6.4a). Each action was performed by ten subjects for three times. Only skeleton module is used throughout the experiments. There are around 10,000 frames in MSR Action3D dataset which is a comparatively smaller size dataset.

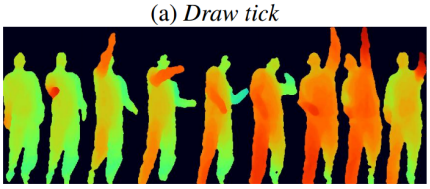
6.1.4 MSRC12

The MSRC12 dataset [21] [2] is originally proposed to "investigate the question of what is the most appropriate semiotic modality of instructions for conveying to human subjects the movements the system developer needs them to perform". Two categories of gesticulation, *i.e.* , Iconic - those imbue a correspondence between the gesture and the reference and Metaphoric - those that

[1] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/
[2] http://research.microsoft.com/en-us/um/cambridge/projects/msrc12/



(a) *Draw tick*



(b) *Tennis serve*

(a) MSR Action3D dataset, only skeleton module is used throughout the experiments.

(b) MSRC12 dataset, a windup example, only skeleton module is used throughout the experiments.

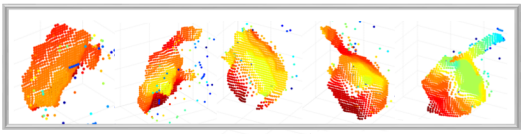
Figure 6.4

represent an abstract content, were investigated. Specifically they are: "*lift outstretched arms, Duck, Push right, Goggles, Wind it up, Shoot, Bow, Throw, Had enough, Change weapon, Beat both, Kick.*" The dataset includes 594 sequences and 719,359 frames, and in total approximately six hours and 40 minutes collected from 30 people performing 12 gestures with 6,244 gesture instances. The skeletal joints car captured by Kinect with an accuracy about 10 centimeters in joint positions. A visual demonstration for the gesture instruction is shown in Fig.6.4b.

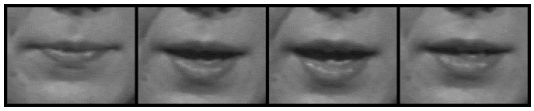
6.1.5 MSRGesture3D

This dataset [105][1] contains a subset of gestures defined by American Sign Language (ASL). There are 12 gestures in the dataset: "*bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j, z*". All of the gestures used in this experiment are dynamic gestures, where both the shape and the movement of the hands are important for the semantics of the gestures. There are ten subjects, each performing two or three times for one gesture class. In total, the dataset contains 336 depth sequences. The self occlusion is more common in

[1] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/



(a) MSRGesture 3D dataset, an instance of point cloud projection from depth images of a gesture *"J"*.



(b) AVLetter Lipreading dataset, an example uttering a letter "A".

Figure 6.5

the gesture dataset. Moreover, missing frames and noises from depth images make the problem more challenging. A point cloud projection sequence of one example gesture is shown in Fig. 6.5a.

6.1.6 AVLetter Lipreading

This dataset [103] [1](*c.f.* Fig.6.5b) consists of 10 speakers saying the letters "A" to "Z", three times each. The dataset provides pre-extracted lip regions of 60×80 gray scale pixels. Audio information is also provided in the format of Mel frequency cepstral coefficients (MFCCs).

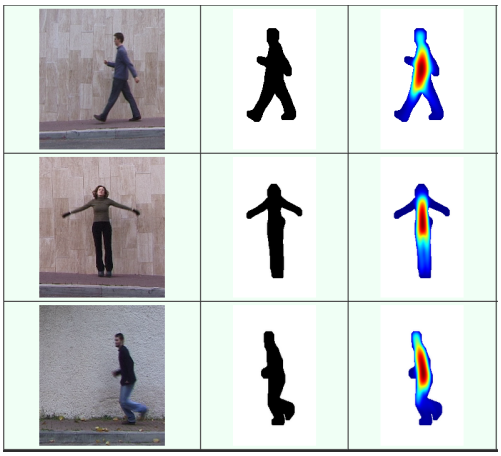
6.1.7 Weizmann

Weizmann[34][2](*c.f.* Fig.6.6a) has 90 low-resolution (180 x 144, deinterlaced 50 fps) video sequences showing nine different people, each performing 10 natural actions such as *"run, walk, skip, jumping-jack (or shortly jack), jump-forward-on-two-legs (or jump), jump-in-place-on-two-legs (or pjump), gallopsideways (or side), wave-two-hands (or wave2), waveone-hand (or wave1), or bend."*

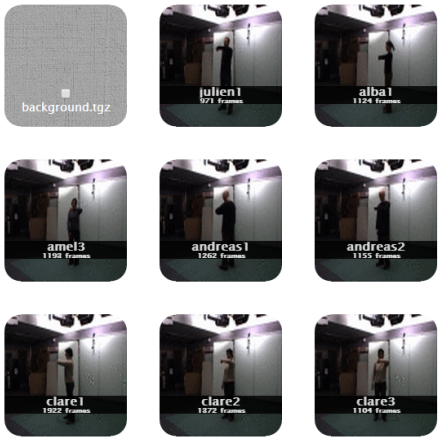
6.1.8 Inria Xmas Motion Acquisition Sequences (IXMAS)

In this dataset[46][3](*c.f.* Fig.6.6b), each action is performed three times by 10 different subjects and sequences are recorded from different viewpoints with

[1] http://www.ee.surrey.ac.uk/Projects/LILiR/datasets/avletters1/index.html
[2] http://www.wisdom.weizmann.ac.il/ vision/SpaceTimeActions.html
[3] http://4drepository.inrialpes.fr/public/viewgroup/6



(a) Weizmann dataset.



(b) IMXAS dataset.

Figure 6.6

multiple cameras. These actions include *"checking watch, crossing arms, scratching head, sitting down, getting up, turning around, walking, waving, punching, kicking, and picking up"*. One of the challenge of this dataset is that performers can freely rotate their orientation to the recording camera, moreover, there are also drastic appearance variations, self-occlusions.

6.2 Details of the Code

6.2.1 Deep Belief Dynamic Networks

The python project for "Leveraging Hierarchical Parametric Network for Skeletal Joints Action Segmentation and Recognition" can be found at:
`https://github.com/stevenwudi/CVPR_2014_code`

6.2.2 Deep 3D Convolutional Dynamic Networks

The python project, C++/CUDA backend for Deep 3D Convolutional Dynamic Network can be found at:
`https://github.com/stevenwudi/3DCNN_HMM`

6.2.3 CBP and Extended-MHI

The Matlab code for generating Correlated Body Poses and Extended Motion History Images for section 2.2 can be found at:
`https://github.com/stevenwudi/CBP-and-Extended-MHI`

6.2.4 One-Shot-Learning from RGBD Images

The Matlab code for generating "One Shot Learning Gesture Recognition from RGBD Images" for section 2.3 can be found at:
`https://github.com/stevenwudi/Kaggle_one_shot_learning`

6.2.5 Matlab Deep Learning Toolbox

The Matlab Deep Learning Toolbox with pedagogic purposes including Gaussian Bernoulli Deep Belief Network, Maxpooling Convolutional Neural Networks and Multimodal Deep Belief Networks can be found at:
`https://github.com/stevenwudi/DeepLearningTutorials`