LEARNING HIERARCHICAL REPRESENTATIONS FOR VIDEO ANALYSIS USING DEEP LEARNING

by

YANG YANG
B.S. Beijing University of Technology, 2008

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2013

Major Professor: Mubarak Shah

ABSTRACT

With the exponential growth of the digital data, video content analysis (e.g., action, event recognition) has been drawing increasing attention from computer vision researchers. Effective modeling of the objects, scenes, and motions is critical for visual understanding. Recently there has been a growing interest in the bio-inspired deep learning models, which has shown impressive results in speech and object recognition. The deep learning models are formed by the composition of multiple non-linear transformations of the data, with the goal of yielding more abstract and ultimately more useful representations. The advantages of the deep models are three fold: 1) They learn the features directly from the raw signal in contrast to the hand-designed features. 2) The learning can be unsupervised, which is suitable for large data where labeling all the data is expensive and unpractical. 3) They learn a hierarchy of features one level at a time and the layerwise stacking of feature extraction, this often yields better representations.

However, not many deep learning models have been proposed to solve the problems in video analysis, especially videos "in a wild". Most of them are either dealing with simple datasets, or limited to the low-level local spatial-temporal feature descriptors for action recognition. Moreover, as the learning algorithms are unsupervised, the learned features preserve generative properties rather than the discriminative ones which are more favorable in the classification tasks. In this context, the thesis makes two major contributions.

First, we propose several formulations and extensions of deep learning methods which learn hierarchical representations for three challenging video analysis tasks, including complex event recognition, object detection in videos and measuring action similarity. The proposed methods are extensively demonstrated for each work on the state-of-the-art challenging datasets. Besides learning the low-level local features, higher level representations are further designed to be learned in the context of applications. The data-driven concept representations and sparse representation of the events are learned for complex event recognition; the representations for object body parts

and structures are learned for object detection in videos; and the relational motion features and similarity metrics between video pairs are learned simultaneously for action verification.

Second, in order to learn discriminative and compact features, we propose a new feature learning method using a deep neural network based on auto encoders. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes both discriminative and generative properties of the features simultaneously, which gives our features a better discriminative ability. Second, our learned features are more compact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features. Extensive experiments with quantitative and qualitative results on the tasks of human detection and action verification demonstrate the superiority of our proposed models.

To my mom,

for her love and sacrifices.

ACKNOWLEDGMENTS

First and foremost I would like to thank my advisor, Dr. Mubarak Shah, for his precious guidance, encouragement and support. Without it, this work would not have been possible. I really appreciate his patience and efforts to help me make progress in academic research. The knowledge and skills that I have learned from him will continue to shape me in my future life.

I thank all the members of the Canon research group, who made my summer internship enjoyable last year. Especially, I am very grateful to Dr. Bradley Denny and Dr. Juwei Lu, for their scientific advice and many insightful discussions.

For this dissertation I would like to thank my committee members, Dr. Rahul Sukthankar, Dr. Gita Sukthankar, Dr. Kenneth O. Stanley and Dr. Niels Lobo, for their precious services in my committee and valuable comments on my research work.

My PhD life was made memorable in large part due to my colleagues and friends at UCF. I would like to thank Jingen Liu and Imran Saleemi for supervising me at the beginning of this PhD journey. I also thank Berkan Solmaz, Baoyuan Liu, Guang Shu, Gonzalo Vaca, Kishore Reddy, Enrique G. Ortiz, and many many others for traveling this journey alongside me. It is their practical advice and optimism that encouraged and dragged me to the finish line.

The most special thanks goes to my husband, Xiangxin Zhu, who is my best partner, friend, and my research inspiration. He has been giving me his unconditional love and unending support during my good and bad times. I married the best person out there for me.

Finally, I owe everything to my parents for bringing me up like the apple of their eyes. It is through their selfless dedication and sacrifices, that I have come this far.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

With the exponential growth of the digital data, computer vision is becoming an important research area with many applications, such as video surveillance, medical imaging, and autonomous vehicle navigation, etc. The ultimate goal of computer vision is to make computers perceive like humans or even better than humans. One of the important and practical tasks is to analyze and understand images and videos, so called visual recognition. For example, given a video, we expect the computers can tell or describe the activity in it, which humans can achieve effortlessly.

Because humans usually outperform the machine vision systems in all the tasks, building a system [75, 30] that emulates visual recognition in cortex has always been an attractive idea. Many researchers have been proposed different algorithms for visual recognition by analyzing and mimicking the process of human brain. Recently, with the new developed learning blocks [24, 66, 25, 83, 64, 97, 22, 79, 71, 23] based on artificial neural network, the bio-inspired methods have been explored extensively and have achieved significant success in many competitions in the area of speech [28, 61, 62] and object recognition [35, 8, 74].

One of the common properties among the successful deep learning models is the composition of multiple non-linear neural networks. These are in some cases [42, 92, 97, 23, 44] inspired by the hierarchical nature of primate visual cortex . Bioscientists have discovered that [13, 17], different cell types found in visual cortex represent the results of different stages of hierarchical processing, involving the building of increasingly complex response properties from one level to the next. In some "higher" cortical areas, for example, neurons may respond best to very specific stimuli such as colors, or even entire objects such as faces. Actually in the computer vision area, using the hierarchical ideas to solve the visual recognition tasks has become popular over the years. Most importantly, hierarchical approaches have been shown to consistently outperform the single-template (holistic) recognition systems on a variety of visual recognition tasks. Recognition

usually involves the computation of several features at one step and their combination in the next step.

Both the hierarchical representation and the bio-inspired deep learning models have, so far, mostly focused on understanding the single static images, very few has been explored in the video domain. The core technical challenge is the huge dimension of the video data. Since the input of the neural network is usually the flattened raw pixel values of an image, when dealing with videos, simply flattening will dramatically increase the input dimension, followed by the growing number of hidden variables. So then the existing models are not capable to handle this given the limited number of training examples and the current computational power.

However, a fact which should not be neglected is that humans learn through consecutive images most of the time throughout life. The changes between the successive frames capture the motion information which is a crucial cue, especially for action, event recognition and tracking. Moreover, as the videos usually contain multiple images of the same objects with different poses and viewpoint, it is easier to build a richer appearance model from videos rather than single image. Therefore, how to explore and utilize the information of videos wisely and efficiently becomes an important and interesting problem in the computer vision area.

In this dissertation, the goal is to explore and develop more suitable algorithms and systems for video content analysis using the bio-inspired deep learning methods. More specifically, the thesis focuses on learning different hierarchical representations for the tasks of complex event recognition, object detection in specific videos and measuring action similarity. The representations for each task are learned in bottom up and local to global fashion. Each level is designed to capture different types of information of the videos, such as local features, global structures, action or scene concepts. To learn the representations, unsupervised or semi-supervised learning methods are proposed with specific framework designs in the context of different applications.

In the next section, we will start with the overview of the feature representations and hierarchical representation learning. Then we will move to the motivation of the proposed methods

and finally describe the three tasks and our contributions.

1.1 Overview and motivation

Feature representations play an important role in many visual recognition tasks, such as object recognition, action recognition, and event recognition. Many perceptual information processing systems, both biological and non-biological, often consist of elaborate algorithm designs to extract certain features or representations from an input sensory array. The performance of the systems heavily relies on the quality of the designed features. Such features in computer vision area range from simple "on-off" units to "hand" or "face" detectors. The good feature representations should be able to extract the "useful" information from the massive input sources. Features reduces the extremely high dimensional raw data to an affordable size for the computers to process. However, the definition of "useful" information is always dependent upon the application context. Especially in the classification tasks, the features are usually encoded with discriminative property, which can find the regularity within the same class while distinguishing those from different classes.

Many works in the literature have been devoted to manually design the low-level features [52, 9, 85, 10]. One of the desired properties of feature representations is the transformation invariance to a certain degree. Because in many visual recognition tasks, the appearance of the same object changes due to different camera locations and viewpoint. Several works have been proposed to extract the good features from images and videos, such as SIFT, HOG and MBH. They are designed to be scale or rotation invariant by quantizing the edges into histogram bins. The successful hand-designed features have been widely applied in all kinds of vision tasks. Designing particular features for specific problems can definitely improve the performance of the overall system. However, it is time consuming and expensive considering the huge time and efforts researchers spend to decide what is the discriminative information and figure out how to manually encode it into the

feature representations.

Instead of manually designing the features, recently, the bio-inspired learning methods have been proposed to learn the features directly from the raw data. In the 2D image case, the learned features are similar to a bank of Gabor like 2D filters with different orientations and scales. To encode the invariance property of the learned features, researchers further impose the subspace or topographic constraints on top of the learned filters. Hence the filters in the same subspace or in the same topographic neighborhood are with similar orientations and scales, thus are invariant to a certain degree of transformations. In the past decade, the bio-inspired learned features have been showing success among many competitions from speech recognition to object recognition.

Most of the proposed feature learning or deep learning methods have been demonstrated on the 1D audio based speech recognition [20, 61, 45, 21] or the 2D static image based object recognition [74, 8, 35]. Not many works have been proposed to solve the problems based on 3D video. One reason is that the input of the learning algorithm is usually the flattened images (e.g. if the image is 20×20 pixels, the input dimension is 400.). When the image becomes larger (e.g. 40×40 pixels), the input dimension will increase with the square of the image size (input dimension will be 1600). So then, the number of the weights which need to be learned in the network will also increase. Under the condition of limited training examples and computational power, training the giant network would be impractical. Reducing the number of dimension in the input, using the tiny images [34], is one possible solution. However, as in videos, since there is one more time dimension, simply resizing the videos will still results in high dimensional input. Moreover, another bigger issue with video resizing is that there is no obvious way to rescale a video to a constant dimensionality since the variation in temporal duration is much greater than image variation in terms of aspect ratio, whereas the deep learning models require the input of the network to be of the same length.

To solve the input dimension problem, convolutional neural networks (CNN) [79, 42] have been proposed to use the small local patches as input, instead of the whole image. The weights

connecting input to hidden units are shared across the image. CNN learns the low level filters on small input patches, then uses the learned filters to convolve with a larger region of the input image to obtain a set of feature maps. The max or average pooling operation is then performed over a certain neighborhoods. One can therefore extract local patches from these locally-invariant multidimensional feature maps and feed them to higher levels. Besides CNN, researchers also proposed to use deep learning methods to learn local spatio-temporal features followed by bag of word framework to achieve the invariance in temporal duration [40]. So far, most of the methods are only dealing with simple actions or focusing on low-level local spatio-temporal features.

Beyond action recognition, in this dissertation, we explore more challenging tasks for video content analysis, including complex event recognition, object detection in videos and measuring action similarity. Instead of only focusing on the low-level feature learning, we explore the hierarchical representations under each application context. For the complex event recognition, the low-level features are the learned multi-modality features from audio, scene and motion. The mid-level representation is the discovered data-driven concepts learned using three layers of RBM (Restricted Boltzmann Machine). The high-level representation is the sparse representations of concepts learned based on the mid-level features. For the human detection in videos, the low-level local features are learned using convolutional neural network based on auto-encoders, which capture local color edges. The higher level representations, capturing body parts appearance or global body structure, are learned from the local features by increasing the receptive field size. For the action verification, the first level learns the relational filter pairs between two input videos and the second level learns the similarity metrics based on the learned relational filters.

In the next section, we will introduce the three proposed approaches in more detail, and describe the contributions of each work.

1.2 Proposed work and contributions

In this section, we will introduce three approaches that we have proposed to learn the good features and hierarchical representations in video based tasks. We will first describe what is complex event recognition and how we learn the hierarchical representations from the video using the unsupervised deep learning methods. Then we will introduce the proposed semi-supervised learning method which imposes the label information, and how we use it to learn the good features from video for object detection. Finally, instead of modeling one input, we will describe how we modify the semi-supervised model into a gated neural network to model the relationship between two inputs and measuring the similarity between two action videos.

The overall contributions of this thesis are summarized below:

We extensively explore the bio-inspired deep learning and feature learning methods for video content analysis; For complex event recognition, we propose to learn the good features directly from the raw signal of three modalities using Topography Independent Component Analysis (TICA). We further propose to discover the data-driven concepts from the video clips using Deep Belief Networks (DBNs) based on the TICA feature representations. For the object detections in videos, unlike the traditional feature learning methods, which are purely unsupervised, we propose a semi-supervised method which learns the features generatively and discriminatively. Moreover, we design a new framework to adapt the generic object detectors to specific videos without utilizing any tracking or annotation information. For measuring action similarity, instead of modeling the representation of one video, we model the relationship of two input videos to discover the similarity between them. The model learns the relational features and multiple similarity metrics simultaneously by optimizing both at the same time.

1.2.1 Complex event detections using data-driven concepts

Automatic event detection in a large collection of unconstrained videos like TRECVID Multimedia Event Detection (MED) is a challenging and important task due to several aspects. First, the high diversity of the data within the same class. For example, the grooming animal event can be a video of washing a cat or combing a dog. Second, there is no user-tagged data facilitating neither the training nor the testing processes. Third, similar scenes or actions appear among different events. For example, the crowd scene appears both in the event parade and flash mob. Fourth, there is a huge number of videos for training and testing.

The key issue is to describe long complex videos with high level semantic descriptors, which should find the regularity of events in the same category while be able to distinguish those from different categories. We propose a novel unsupervised approach to discover data-driven concepts from multi-modality signals (audio, scene and motion) to describe high level semantics of videos. Rather than using the hand designed features, in this task, we propose to learn the good features using Topographic Independent Component Analysis (TICA) in an unsupervised manner directly from the raw signals of three modalities: 1D audio, 2D image and 3D video. Further, instead of manually collecting and labeling the concepts, and training the discriminative concept detectors, we further propose to discover the data-driven concepts from three modalities based on the learned TICA features using deep belief nets (DBNs). So then each complex video can be represented as the similarity score of the clips in that video with the data-driven concepts. Finally, a compact and robust sparse representation is learned to jointly model the concepts from all three modalities.

Notice that, in this work, all the learning methods are unsupervised, which do not need any label information. The unsupervised approach is suitable for the big data, especially when the labels are not available or it is not practical to label all of the videos. As the traditional deep learning or feature learning methods, unsupervised learning is in contrast to previous supervised

learning methods, such as Back Proragation (BP). Instead of fitting the learned model from the raw signals directly to the labels, it is more meaningful to make the representations more generative in order to deal with larger data.

1.2.2 Semi-supervised learned features for object detection in videos

Object detection has been explored extensively and has achieved significant success in the past decade [9, 11, 15, 82, 84, 89]. Most of the state-of-the-art detectors are designed for a single static image and are trained from a large set of labeled examples. The performance of a detector will inevitably be degraded when it is applied to frames in a video taken under conditions which are very different from those of the training examples.

We propose an approach to improve the object detection results in a given video obtained from a generic trained object detector. Our method does not utilize the tracking results as the detection-by-tracking methods do, nor any annotation from the video as other supervised detector adaptation methods do. Instead, we only use the detection scores obtained from the generic detector and treat this as a semi-supervised classification problem. The key idea is that object appearing in different frames of video should share some similar properties and we want to extract and exploit these properties for classification.

Since selections of the right features plays an important role in object detection, we argue that, the classical hand-crafted features, such as HOG, SIFT, may not be universally suitable and discriminative enough for every type of video. In a particular video, the way objects appear would share some similar properties which could be leveraged to distinguish them from the non-objects. Hence, unlike other proposed methods, which are built using hand-designed features, we learn the good features directly from the raw pixels of the video itself.

In order to learn discriminative and compact features, we propose a new feature learning method using a deep neural network based on auto encoders. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes both discriminative and generative

properties of the features simultaneously, which gives our features better discriminative ability; second, our learned features are more compact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features. Moreover, we learn a discriminative feature hierarchy from local patches to global images.

1.2.3 Learning features and metrics jointly for measuring action similarity

Measuring the similarity of two human actions is an important task which has many applications. It is challenging since matching video pairs is intimately tied to the invariance modeling: two videos are the same if they are invariant under some classes of allowable transformations.

Two critical factors that affect the performance include low-level feature representations and similarity metrics. However, finding the right representations and metrics is hard. In this work, we describe a novel approach that learns the similarity metrics and the feature representations jointly. More specifically, we learn the spatial-temporal feature pairs and multiple metrics which can model the complex action transformations. In this way, the features and the metrics will cooperate to achieve an optimal solution.

Oftentimes two distinct actions share the same scene background (this happens a lot in sport videos). Existing generative feature learning approaches [40] tend to be distracted by the common scene instead of learning discriminative features to tell apart the actions. In order to improve the discriminative ability of the learned features, we propose a new learning method using both generative and discriminative objectives based on gated auto encoders [54].

Our method differs from existing representations or metric learning methods in two ways: 1) while other methods treat feature learning and metric learning as independent tasks, we argue that they should be learned jointly since features and metrics are tightly interdependent; 2) our method learns more discriminative features than its purely generative counterparts.

1.3 Organization of the thesis

The rest of this thesis is organized as follows. Chapter 2 contains the literature review on deep learning, feature learning, action and event recognition, and object detections in video. In Chapter 3, we present complex event detection using data-driven concepts. Chapter 4 presents our framework for object detention in videos using semi-supervised learned features. Chapters 5 describe the action verification task by learning the features and metrics jointly and discriminatively. Finally, Chapter 6 describes some future directions.

CHAPTER 2: LITERATURE REVIEW

In this chapter, we first focus on the review of recent works on feature learning and deep learning in Section 2.1 and describe several off-the-shelf feature learning algorithms using only single-layer network in Section 2.2. Since our methods have been applied to complex event recognition, object detection and measuring action similarity, we further discuss the literature of each task in Section 2.3, 2,4 and 2.5 respectively.

2.1 Deep learning and feature learning

Low-level hand designed features have been heavily employed with much success in scene, object and action recognition, such as MFCC [60], SIFT [52], STIP [38], Cuboid [10], and MBH [85]. Interestingly, among most of the successful features, one common property is involving the calculation of edge gradients, followed by histogram quantization or pooling operation. It is effective at capturing low-level local motion information. However, for understanding images and videos, the challenge nowadays is to find good representations for mid and high-level information, such as object parts.

Recently, several unsupervised learning methods have been proposed to learn features from unlabeled data. Many new schemes for stacking layers of features to build "deep" representations have been proposed. Most have focused on creating new training algorithms to build single-layer models that are composed to build deeper structures. Among the algorithms considered in the literature are sparse-coding [94], RBMs [24, 66], sparse RBMs [43], sparse autoencoders [18, 64], and mean-covariance RBMs [63], as well as many others [44, 96]. Thus, amongst the many components of feature learning architectures, the unsupervised learning module appears to be the most heavily scrutinized. Biologically-inspired sparse learning algorithms have long been studied by researchers in the filed of natural image statistics. There has been a growing interest in applying

these methods to learn visual features. The authors of [31] show that Independent Component Analysis (ICA) can be used as a self-taught learning method to generate saliency maps and features for robust recognition. A TICA based convolutional neural network is adopted in [39] for static images, which achieves state-of-the-art performance on several datasets. Authors in [97] propose a deep learning network based on sparse coding to implement the spatial pooling of object recognition problem. They achieve state-of-the-art performances on Caltech 101 and Caltech 256 dataset.

From image to video, extending the 2D features to 3D is the predominant methodology in action and event recognition. In [79], a GRBM based deep learning algorithm is proposed to learn the spatio-temporal features for object and action recognition. This method can be considered as an extension of convolutional RBMs [44] from 2D to 3D. It focuses on scaling up from low level feature to high level (global) feature using convolutional layers and average spatial, max temporal pooling strategy. The results show that the proposed method outperforms classical hand-designed features. In [41], a novel Independent Subspace Analysis(ISA) combined with two layers convolutional network is proposed for learning spatio-temporal features from videos for human action recognition. It adopts Wang's pipeline [86] using dense sampling and replaces the feature descriptors with learned features. Using their learned features, superior results are achieved on several benchmark datasets. Another advantage of [41] is the lower computational cost during feature learning. Due to the current trends, challenges and interests in action recognition, this list will probably grow rapidly.

2.2 Unsupervised learning methods

In this section, we describe several off-the-shelf feature learning algorithms using single-layer network.

The architecture of the single-layer network is shown in figure 2.1. To learn the filters, it

begins by extracting random sub-patches (or sub-cuboids from 3D videos) from unlabeled input images. Each patch is stretched into a vector in $\mathbb{R}^N$. The training data X is then constructed by concatenating C randomly sampled patches, $x^{(1)}, \cdots, x^{(C)}$, where $x^{(i)} \in \mathbb{R}^N$. The goal is to learn the weights W connecting x to simple units h. Given this $N \times C$ matrix X, we apply the following pre-processing and unsupervised learning steps.

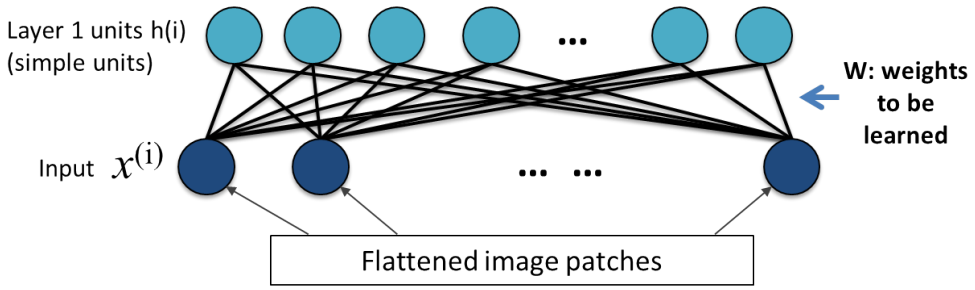


Figure 2.1: Single-layer network in the 2D image case. Each input $x^{(i)}$ is a flattened image patch. The network learns weight W in which each row vector is a filter same size with $x^{(i)}$.

It is common practice to perform several simple normalization steps before attempting to generate features from the data. In the learning process, we assume that every patch $x^{(i)}$ is subtracted from its mean (local brightness) and divided by its the standard deviation (local contrast).

After normalizing each input vector, the entire dataset X may optionally be PCA whitened, which means performing PCA with each component divided by its standard deviation and keep the first n principal components. While this process is commonly used in deep learning work it is less frequently employed in computer vision. We use X' to denote the PCA whitened $n \times C$ training data matrix, and $x'^{(i)}$ to denote one whitened image patch.

After pre-processing, an unsupervised learning algorithm is used to discover features from the unlabeled data. Each unsupervised learning algorithm tries to learn a matrix W that map the input vector $x^{(i)}$ or $x'^{(i)}$ to a new m dimensional feature vector. Each row of the matrix W, denoted

by $w^{(j)}$, can be treated as a linear filter, and the output is composed of m filter responses. Afterwards, nonlinear operation is optionally imposed to the linear filter responses for some models. In this section, we will introduce several different unsupervised learning methods: (1) PCA, (2) ICA, (3) ISA, (4) TICA, (5) AE and (6) RBM.

1. **Principal Component Analysis (PCA)**

PCA is commonly considered to be the most convenient dimension reduction tool. In natural image statistics, PCA is the most basic learning algorithm that learns the components which are uncorrelated with each other and have high variances. PCA is performed by optimizing the following problem:

$$\underset{W}{\operatorname{argmax}} \sum_{i=1}^{m} var(w^{(i)}X), \text{ subject to } WW^T = I. \tag{2.1}$$

The maximization of variance enforces that the principal components are uncorrelated. PCA is practically implemented by eigenvector decomposition of the covariance matrix of the data.

2. **Independent Component Analysis (ICA)**

Instead of using the second-order moments(variance and covariance) as the measure of the learned components, ICA aims to learn a set of components that are statistically independent. Maximum likelihood learning is adopted to maximize the joint probability of the learned components, and the log-probability is approximated by $\tanh$ function. The ICA optimization problem can be formulated as:

$$\underset{W}{\operatorname{argmin}} \sum_{i=1}^{C} \sum_{j=1}^{m} \tanh(w^{(j)}x'^{(i)}), \text{ subject to } WW^T = I. \tag{2.2}$$

The orthogonality of the learned component weights make sure that they are uncorrelated due to the fact that the input data is already whitened and uncorrelated.

3. **Independent Subspace Analysis (ISA)**

ISA is a nonlinear unsupervised learning method, which means the output of each component is not a simple inner product of the input vector and the weight vector. In ISA, a given number of subspaces are defined before learning the components. The components inside each subspace are not supposed to be independent, and the energy of the subspaces, which are computed by taking the square root of the sum of energy of all the components in one subspace, is maximized to achieve the independence of different subspaces. If there are s components in each subspace, then the number of subspaces is $p = m/s$. The optimization problem is shown as follows:

$$\underset{W}{\operatorname{argmin}} \sum_{i=1}^{C} \sum_{j=1}^{p} \sqrt{\sum_{k=1}^{s} (w^{(j)(k)} x'^{(i)})^2}, \text{ subject to } WW^T = I. \tag{2.3}$$

where $w^{(j)(k)}$ is the kth component in the jth subspace. The sum-square and square root operations introduce non-linearity into the algorithm and make it more flexible and capable of learning more complex structure. The dependence of components within each subspace leads to invariance inside each subspace and makes the learned filters more robust to small variation.

4. **Topographic Independent Component Analysis (TICA)**

TICA is an extension of ISA. TICA is inspired by the topographical organization in the cortex. While subspaces in ISA are independent from each other, the components in TICA are related in a 2D topographical grid based on their distance. A neighborhood matrix is defined to describe the statistical similarity between nearby components. The energy of each component is composed by the weighted sum of the square of the nearby components' response. We use A to denote the neighborhood matrix, and the following function is optimized:

$$\underset{W}{\operatorname{argmin}} \sum_{i=1}^{C} \sum_{j=1}^{m} \sqrt{\sum_{k=1}^{m} \left(A_{j,k} w^{(k)} x'^{(i)}\right)^2}, \text{ subject to } WW^T = I. \tag{2.4}$$

Such a topographic connection between different components brings more flexibility while keeps each component statistically different from each other. Such structure makes the most of the components and maximizes the performance with a fixed number of components.

5. **Auto-encoder (AE)**

Auto-encoder [83] attempts to reconstruct the data by minimizing the following cost function:

$$\underset{W}{\operatorname{argmin}} \sum_{i=1}^{C} \sum_{j=1}^{m} \|x'^{(i)} - w^{(j)T}(w^{(j)}x'^{(i)})\|^2. \tag{2.5}$$

Here we implicitly assumed linear activation, no biases and tied weights.

6. **Restricted Boltzmann Machine (RBM)**

RBM is a stochastic neural network that aims to learn regularity from the input visible data in an unsupervised manner. As a particular case of Boltzmann machine, there is no visible-visible and hidden-hidden connections in RBM. The links between the hidden layer and the visible layer are bidirectional and symmetric, and the hidden layer can be generated from the visible layer as the following equation:

$$h_j = \delta(b_j + w^{(j)}x^{(i)}) \tag{2.6}$$

where h_j is one hidden node, b_j is the bias, and $\delta(x) = (1 + exp(-x))^{-1}$ is a logistic function. In our experiment, we train single layer RBM and use the hidden layer as the descriptor of the input cuboid.

2.3 Computational Cost

For the unsupervised learning methods, feature learning consumes a relatively large amount of time. We compare the computational cost of different unsupervised learning methods on the HMDB51 [36] dataset using a Dell T3500 desktop with Quad-core 3.07GHz Intel Xeon CPU. The

Method	PCA	ICA	ISA	TICA	AE	RBM
Training Time	3 minutes	8 hours	18 hours	20 hours	16 hours	33 hours

Table 2.1: Comparison of training time for different unsupervised learning methods

cuboid size is $16 \times 16 \times 10$, and the training samples take about 16G memory. We perform 1000 iterations for ICA, ISA, AE and TICA and 30 epochs for RBM. Table.2.1 shows the training time for different learning methods. All the unsupervised methods except PCA takes several hours for training.

The testing time for each method is almost the same because the main computation is the inner product between the filters and the cuboids. ISA and TICA also involve nonlinear pooling, but these operations take much less time than the cuboid filtering.

Recall that the methods described above are the off-the-shelf algorithms for unsupervised single-layer feature learning. The learned first-level filters (features) are Gabor like edge detectors. In the later sections, we will describe our proposed learning methods which learn multi-level representations of videos for three tasks.

2.4 Complex event recognition

The rapid growth of digital videos has made an urgent need for having effective methods for video analysis. Among all, complex event recognition is one of the most challenging problems with increasing demand. By complex, we mean that each event video can contain multiple concepts, such as actions and scenes. The challenging nature of event recognition problem lies in the fact that: First simple actions and scenes are the building blocks of events while the action and scene recognition problem itself is still very challenging. Second, once solving the low level problems, how to fuse the information into high-level decision is a non-trivial problem.

Recently, the bag-of-words (BoW) approach has achieved impressive results in many recognition problems including action recognition. The straight forward simple way of solving complex event would be treating the event as action. Then event recognition could be solved by extracting spatio-temporal interesting features and represent the event using BoW histograms. However, this approach has innate limitations in representation and semantic description of the underlying data as it jumps directly from low level features to the very high level class labels. Therefore, the methods which are based on BoW approach cannot easily provide any semantic intermediate description of the data. For recognizing complex events, it is crucial to learn the low-level events along with their relationships to the event categories. For example, for Birthday party event, low-level events may include: person cheering, person singing, person blowing candles, person taking pictures, etc.

Thus, we argue that it is very logical to decompose the complex event recognition into mid-level concepts of different modalities: audio, images and video. The audio, scene and action recognition problems have been widely explored in the past. The concept detectors can provide semantic representation for videos with complicated content, which can be very useful for developing powerful retrieval or filtering systems for consumer media. Lots of effort [51, 50] have been devoted to building huge datasets for training concept detectors. However most of them are recorded in a well constrained conditions [48, 67], which are not suitable for detecting actions in complex events. [51] provides a benchmark dataset with 25 selected concepts over a set of 1,338 consumer videos. But its concept collections are based on static images only. Audio or motion concepts are not used. Due to the large diversity of the data and insufficient training samples, concept detectors perform far below expectation. In this thesis, we propose an unsupervised approach to discover concepts from three modalities using DBN, which has been proposed to solve digit recognition and achieved promising results [24]. Besides, it has been shown in [22] that DBN performs better than PCA and LLE (Locally Linear Embedding) in terms of dimension reduction.

Multiple data sources can be combined using either early fusion or late fusion strategies [77, 51, 91, 72]. Traditional fusion methods treat each source independently[91]. We argue that it

is desirable to exploit the relationships between multiple sources to achieve robust classification. In this thesis, we propose sparse coding [59] to perform late fusion and empirically show the benefits of such approach.

2.5 Object detections in videos

Several works have been proposed to detect objects in videos. They can be divided into two categories. One is detection by tracking [3, 16, 19], which use the trajectories information to help improve detection results and the improved detection can be used backward to improve tracking. Another is detection by detection [29, 98, 57, 70, 68, 47, 69, 90, 87, 88] and most methods in this category treat this as a semi-supervised problem and try to propagate the label to new examples correctly. Authors in [90] used HOG feature with tree coding in a non-parametric detector adaptation method. Javed et al. [29] proposed a co-training based approach using color and edges as the feature representation. Authors in [47] trained two disparate classifiers simultaneously by carefully choosing independent or complementary hand designed features.

Most of the successful methods above heavily depend on choosing the correct low-level features, such as SIFT, HOG and color histogram. Notice that the object appearance in video frames should share some regularities among each other, which could be used for discriminative classification. We intend to learn the good features directly from the raw pixels of a video.

Feature learning[46, 64, 97, 24], which finds concise, slightly higher-level representations of inputs, has been successfully applied to object recognition and scene recognition. Most of the methods [46, 40, 83, 64, 79, 97, 24] are unsupervised learning algorithms. The goal is to use unlabeled data to improve the supervised learning task, although the unlabeled data cannot be associated with the supervised task. However, in our case since we have the label information(the confidence scores from the detector), and since training and testing examples from video frames are highly correlated , we would like to use the label information to directly learn a more discrimi-

native feature set for better classification. Therefore, we need to learn the features generatively and discriminatively. Several methods [99, 65, 53] have shown significant improved results with discriminative features. The authors in [53] used sparse coding to learn multiple dictionaries for each category. [65] proposed to learn a semi-supervised method on top of bag-of-work representation for document recognition. The authors in [99] proposed a single level hybrid learning method for incremental feature learning. In this paper, we propose a new feature learning method using a deep neural network based on auto encoders with invariance design. We learn three levels of discriminative features from local to global by optimizing both discriminative and generative properties of the features simultaneously.

2.6 Action recognition and verification

To judge if two videos in a pair are the same action or not is an essential task with many applications in computer vision area. Most of the methods in the literature solve this problem from two perspectives: finding the invariant representation and determining the right distance metric. To make the feature invariant, many low-level descriptors [52, 9, 38, 85, 40] have been proposed and demonstrated on several action benchmarks. Each of the features are designed to deal with a limited type of invariance. Once the low level features are designed for a particular task, to solve the pair matching problem, most of the state-of-the-art methods directly use the existing pre-defined metrics, such as histogram intersection [48], χ^2 [85], to measure the similarity distances between videos based on the proposed features. However, since the features are lie in a high-dimensional space, roughly using a pre-defined metrics without learning the manifold will inevitably degrade the performance.

Besides carefully choosing the low-level features, many researchers also examine the verification problem as a pure pair matching problem given the fixed feature representations. Basically, they want to design or learn the transformations between two videos which allow invariants. Mod-

eling these transformations by hand is usually either impractical or extremely time-consuming and difficult. Hence, researchers have proposed to learn the transformations, so called metric learning methods, which have been used with varying levels of success [73, 80, 4, 7, 93, 58, 49]. Unfortunately, many of them are designed based on Mahalanobis distance, in which case they are not expressive enough to model the subtle variations, and many of them only learn one type of transformation(one metric) which is obviously not enough to model the complex transformations. Moreover, if the feature representation is not distinctive enough, learning the metric will be hard and impractical. Overall, as described above, neither the feature level nor the metric level methods is optimal enough to match action pairs. Since they are two inseparable factors, designing them independently will definitely downgrade the performance. In this paper, we propose to learn the good features and metrics jointly and directly from the raw pixels of videos.

Deep learning, which finds concise, higher-level representations of inputs, has been successfully applied to object recognition, scene recognition, as well as face verification. Most of the methods [46, 64, 55, 44, 97, 24, 79, 54, 78] are unsupervised learning algorithms. The idea is to use unlabeled data to facilitate in a supervised learning task, even if the unlabeled data does not have labels. However, in our case, since we are more interested in the actions appearing in the videos and we have the label of videos in a pair, we would like to use the label information to directly learn a more discriminative model for better classification. Therefore, we propose to learn the model in a semi-supervised manner. Several methods [99, 65, 53] have shown significantly improved results by considering discriminative property. The authors in [53] used sparse coding to learn multiple dictionaries for each category. [65] proposed to using a semi-supervised method on top of bag-of-word representation for document recognition. The authors in [99] proposed a single level hybrid learning method for incremental feature learning. In this paper, we propose to address the action verification in videos problem using a discriminative and generative method based on gated auto encoders.

2.7 Summary

In the following chapters, we will introduce several formulations and extensions of deep learning methods which learn hierarchical representations for video analysis, including complex event recognition, object detection in videos and measuring action similarity. For complex event recognition, we propose a novel unsupervised approach to discover data-driven concepts from multi-modality signals (audio, image and video) to describe high level semantics of videos. Our methods consists of two main components: we first learn the low-level features separately from three modalities. Then we discover the data-driven concepts based on the statistics of learned features mapped to a low dimensional space using deep belief nets (DBNs). For improving generic object detector in videos, we present a new model that learns the hierarchical object representations in a semi-supervised manner. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes for both discriminative and generative properties of the features simultaneously, which gives our features better discriminative ability; second, our learned features are more com- pact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features. For measuring action similarity, we describe a novel approach that learns the features and metrics directly from the data. We propose a generative plus discriminative learning method based on gated auto encoders to simultaneously learn the features and their associated metrics. Extensive experiments with quantitative and qualitative results on the three tasks demonstrate the superiority of our proposed models.

CHAPTER 3: COMPLEX EVENTS DETECTION USING DATA-DRIVEN CONCEPTS

3.1 Introduction

User uploaded videos on the internet have been growing explosively in recent years. Automatic event detection in videos is an interesting and important task with great potential for many applications, such as on-line video search and indexing, consumer content management, etc. However, it is a very challenging task to deal with large corpora of unconstrained videos with huge content variations and uncontrolled capturing conditions (as illustrated in Fig.3.1).

Common approaches in event recognition rely on hand-crafted low level features such as SIFT [52], STIP [37], MFCC [60], and human-defined high level concepts [51]. The use of high level semantic concepts have been proven effective in representing complex events [91]. However, how to discover a powerful set of semantic concepts is still unclear and has not been investigated in previous works. The drawbacks of human defined concepts include: (1) it's hard to extend these concepts to a larger scale, (2) they can not handle multiple modalities, and (3) the concepts don't generalize well to new datasets.

In this chapter, we propose a novel unsupervised approach to discover event concepts directly from training data in three modalities (audio, image frames and video).

We first learn low level features for each modality using Topography Independent Component Analysis (TICA), which has shown superior performance over popular hand-designed features in [39]. Then we map our low level features to more compact representations using deep belief networks (DBN) [24]. After that, our data-driven concepts are learned by clustering the training data in a low-dimensional space using vector quantization (VQ). This dimension reduction step is crucial to produce reasonable clustering results. Finally, we merge the concepts from three different modalities by learning compact sparse representations. The framework is shown in Fig.3.2.

Figure 3.1: Randomly selected example videos from our dataset. Each row shows frames of four videos from one categories.

We argue that unsupervised learning of concepts is appropriate due to two reasons. First, the disconnection between limited linguistic words and complexity of real world events makes human definition of visual concepts very hard if not impossible. We will later show that large number of learned concepts help to improve recognition accuracy significantly. Second, most of the time, insufficiency of annotated data prevents us from learning concepts in supervised manner. We present extensive evaluation of our method. The results show that our proposed approach significantly outperforms popular baselines.

The rest of the chapter is organized as follows. The proposed method is presented in section 3.2-3.4 in the following order: Low-level Feature Learning, Data-driven Concept Discovery, and Event Representation Learning. Extensive experiment results, comparisons and analysis are reported in section 3.5. Finally, we summarize in section 3.6.

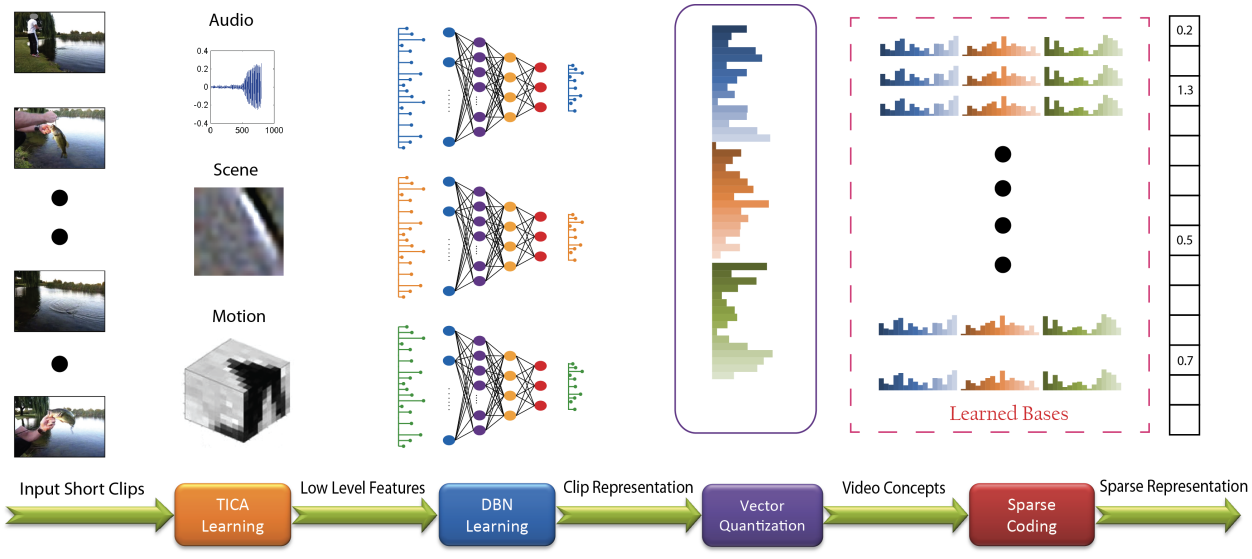


Figure 3.2: Framework of the proposed method. Each video is divided into short clips. We first learn low level features for each modality using Topography Independent Component Analysis (TICA). Then we map our low level features to more compact representations using deep belief networks (DBN). After that, our data-driven concepts are learned by clustering the training data in a low-dimensional space using vector quantization (VQ). Finally, we merge the concepts from three different modalities by learning compact sparse representations.

3.2 Learning low-level features

We use the TICA feature learning networks [39] to learn the invariant audio, image and video features from 1D audio signal, 2D image patches and 3D video cuboids respectively. To make the paper self-contained we briefly descbribe TICA in the context of event recognition. For more details, please refer to [39] and [27]

We write $x^{(p)} \in \mathbb{R}^n$ as the p^{th} whitened local raw signal extracted from one modality of the video clips. For 2D image patches and 3D video cuboids, we flatten them into 1D vectors. Learning features can be viewed as learning a set of filters that map the raw signal into feature space by calculating the filter responses. TICA is a two-layered network. The first layer learns m filters $S \in \mathbb{R}^{m \times n}$ from input $x^{(p)}$ by minimizing Eqn.3.3. The filter responses are the activations of the first hidden layer units H. The second layer's filters V are manually fixed to pool over a small neighborhood of adjacent first layer units H, representing the subspace structure of the neurons in the first layer. More specifically, in a 2D topography, the h_k units lie on a 2D grid, with each activation of the second layer r_i pooling over a connected 3×3 block of H units through V.

In more detail, the activation of units k in the first layer is:

$$h_k\big(x^{(p)}; S\big) = S_k \cdot x^{(p)}, \tag{3.1}$$

where S_k is the k^{th} row of S.

The activation of unit i in the second layer is:

$$r_i\big(h_k; V\big) = \sqrt{\sum_{k=1}^{m} V_{ik} h_k^2}, \tag{3.2}$$

where $V \in \mathbb{R}^{m \times m}$ is a fixed matrix that encodes the topography of the hidden units H. m is the number of hidden units in the first layer.

In the filter learning process, the optimal S is learned by minimizing function:

$$S^* = \arg\min_S \sum_{p=1}^{T} \sum_{i=1}^{m} r_i\big(x^{(p)}; S; V\big). \tag{3.3}$$
$$\text{s.t.} \quad SS^T = I$$

where T is the total number of training samples. The orthogonality constraint $SS^T = I$ provides competitiveness and ensures that the learned features are diverse. In the feature extraction process, given S^* and the new whitened local raw signal x, the activation in the second layer R will be served as the feature of x.

Considering the data we have is quite diverse and huge amount, we argue that learning good features directly from the data is very efficient. We choose TICA as our low-level building block because of its two advantages: feature robustness and less computational complexity. The pooling architecture of TICA ensures the learned features are invariant to slight location and orientation shifts, and selective to frequency, rotation and motion velocity. The filter learning is much faster than other methods such as GRBM [79] since the gradient of the objective function Eqn.3.3 is tractable. The feature extraction is also fast compared with sparse coding as the feature is simply computed through the matrix vector products.

3.3 Data-driven concept discovery

Previous works [48, 67] use human defined concepts for action recognition. However, this is not suitable to event recognition due to two reasons: first, defining concepts that describe the huge diversity of human actions using limited linguistic words is not practical. Second, current event datasets [2] do not have detailed annotation of action concepts for each video clip, which make it hard to train concept detectors. These two problems originally motivate us to propose data-driven concept discovery.

We assume there is only one type of concept from each of the three modalities appearing in a single shot clip. The idea is that we want to find a representation $Y \in \mathbb{R}^d$, which can map each clip of different modalities from raw signal space to a semantic space, where clips with similar concepts are near to each other. Considering the high diversity of our data, instead of pooling the low-level TICA features spatial-temporally, we use bag of word (BoW) histogram $Q \in \mathbb{R}^D$ by adopting vector quantization (VQ) technique using K-means soft assignment [81].

One problem is, the BoW histogram is usually long (corresponding to large cookbook) in order to capture variations of data. And k-means is well-known to be sensitive to noise in a high dimensional space especially when we apply Euclidean distance as similarity measurement. To address this, we propose to use deep belief nets (DBN) [22] to learn a lower dimensional representation for the clips from each modality. A DBN is a two-layered network, which is a stack of restricted Boltzmann machines (RBMs). The activations of the lower RBM serve as the input of the upper RBM. In each RBM, the hidden layer captures strong correlations of the units' activities in the layer below. For our highly complex event data, stacking several RBMs is an efficient way to progressively expose low-dimensional, non-linear structure. We begin by describing RBM in the case of real valued input following the description in paper [22] and [43], and then we show how we use the learned clip representation to discover data-driven concepts from each modality.

DBN learning: We start with the visible units Q in the bottom layer, which are essentially the BoW representation of each clip. A set of hidden units l are built through symmetric connection weights represented by weight matrix W. We can view the RBM as an undirected graphical model and the energy of any state in it is given by the following function:

$$\begin{aligned} E(q,l) &= -\log P(q,l) \\ &= \frac{1}{w\sigma^2}\sum_i q_i^2 - \frac{1}{\sigma^2}(\sum_i c_i q_i + \sum_j b_j l_j + \sum_{i,j} w_{ij} q_i l_j). \end{aligned} \tag{3.4}$$

Here, σ is the standard deviation of the Gaussian density, l_j are hidden unit variables, q_i are visible unit variables, w_{ij} is the weight connected with q_i and l_j, c_i and b_j are the bias term of visible and hidden units respectively. The learning process is to estimate w_{ij}, c_i and b_j through minimizing the energy of states drawn from the data q distribution, and raise the energy that are improbable given the data. We follow [43] to use contrastive divergence learning which gives an efficient approximation to the gradient of the energy function. Further, in each iteration, we apply contrastive divergence update rule, followed by one step of gradient descent using the gradient of the regularization term.

Once training a layer of the network is finished, we feed the output values of this layer as inputs of the next higher layer. Finally, after finishing training all the layers, we obtain the clip representation as the outputs of the last layer, denoted as $Y \in \mathbb{R}^d$. By doing so, We map the original features to much lower dimensional space since $D \ll d$.

Building Concepts: The low dimensional representations Y from similar clips are then grouped into concepts with a semantic meaning. In our framework, concepts are obtained from three modalities separately and each event video is represented as the occurrence frequency of each concepts from three modalities, denoted as Z.

3.4 Event representation learning

It is common that concepts of different modalities are highly correlated with each other. For example, in a birthday party event, action concept 'people dancing' almost always co-occurs with concept 'happy music' or scene concept 'crowd people', instead of 'horrible music' nor 'traffic scene'. By modeling the interaction context and inter-modality occurrence of concepts, we can remove noisy concepts and further improve the event representation. The idea is that we want to learn a set of bases which capture the co-occurrence information of concepts and that way the event can be represented as a linear combination of the bases. Further by imposing the sparsity on the

coefficients, the noisy or irrelevant concepts will be removed.

More precisely, given N events represented in terms of concatenated concepts from three modalities, $\{Z^{(1)}, \cdots, Z^{(i)}, \cdots, Z^{(N)}\}$. We learn the basis by modeling this problem as a sparse coding problem [59]:

$$\phi^* = \arg\min_{a,\phi} \sum_i \left\| Z^{(i)} - \sum_j a_j^{(i)} \phi_j \right\|_2^2 + \beta \|a^{(i)}\|_1$$
$$\text{s.t.} \quad \|\phi_j\|_2 \leq 1, \ \ \forall j \in \{1, 2, \cdots, s\}. \tag{3.5}$$

where ϕ_j is the basis vector, $a_j^{(i)}$ is the coefficient of i^{th} event associated with j^{th} basis. The first term in Eqn.3.5 is reconstruction error, while the second term enforces sparsity of coefficients. β is the relative weight to balance the two terms. We use sparse coding algorithm in [59] to solve this minimization problem.

After learning a set of bases ϕ, we can encode an input event $Z^{(t)}$ as sparse linear combination of the bases. The combination coefficients $a^{(t)}$ will serve as the final representation of this event, and can be obtain by solving Eqn.3.6.

$$\arg\min_{a^{(t)}} \left\| Z^{(t)} - \sum_j a_j^{(t)} \phi_j \right\|_2^2 + \beta \|a^{(t)}\|_1. \tag{3.6}$$

SVM with χ^2 kernel is used for classification [6].

3.5 Experiments

In this section, we will describe the dataset sand discuss several interesting observations that we had.

3.5.1 Datasets and experimental settings

We tested our approach on TRECVID 2011 event collection [2], which has 15 categories: "Boarding trick", "Flash mob", "Feeding animal", "Landing fish", "Wedding", "Woodworking project", "Birthday party", "Changing tire", "Vehicle unstuck", "Grooming animal", "Making sandwich", "Parade", "Parkour", "Repairing appliance", "Sewing project". As shown in Fig.3.1, it is a new set of videos characterized by a high degree of diversity in content, style, production qualities, collection devices, language, etc. The frame rate ranges from 12 to 30 fps, resolution ranges from 320×640 to 1280×2000, the time duration ranges from 30 seconds to 5 minutes.

We manually defined and annotated an action concept data set based on TRECVID event collection, which has 62 action concepts (e.g. open box, person cheering, animal approaching, wheel rotating, etc.) for approximately 9,000 videos. To the best of our knowledge, this is the largest action concepts dataset in related literatures.

In the experiments, we first resize all the videos to 480×640, and then divide each video into 4 and 10 seconds clips with 2 seconds overlap, based on our observation that the motion concepts duration varies from 2 to 10 seconds. There are approximately 300,000 clips in total. Performance was evaluated in terms of Mean Average Precision (MAP) on 15 events.

Also, we compare our low-level features with other hand-designed features on UCF YouTube action dataset, which has 11 action categories. 25-fold cross-validation is used. It is important to note that although the YouTube dataset is one of the most extensive realistic action datasets in the vision community, it is still less noisy and much simpler then the TRECVID data in terms of inner-class diversity.

3.5.2 Low-level feature extraction

We use TICA to learn three modalities of feature representation: audio, image and video. For each modality, approximately 200,000 sampled signals/patches/video-blocks are used to train

the filters and 600 filters from each modality are finally chosen as the bases for feature construction. The audio signal is extracted with sampling rate of 16 KHz. The inputs of the visual layer are 800, 20×20, $20 \times 20 \times 10$, respectively, in the three modalities. Fig. 3.3, 3.4, 3.5 shows randomly selected learned filters for 1D, 2D and 3D training examples respectively.

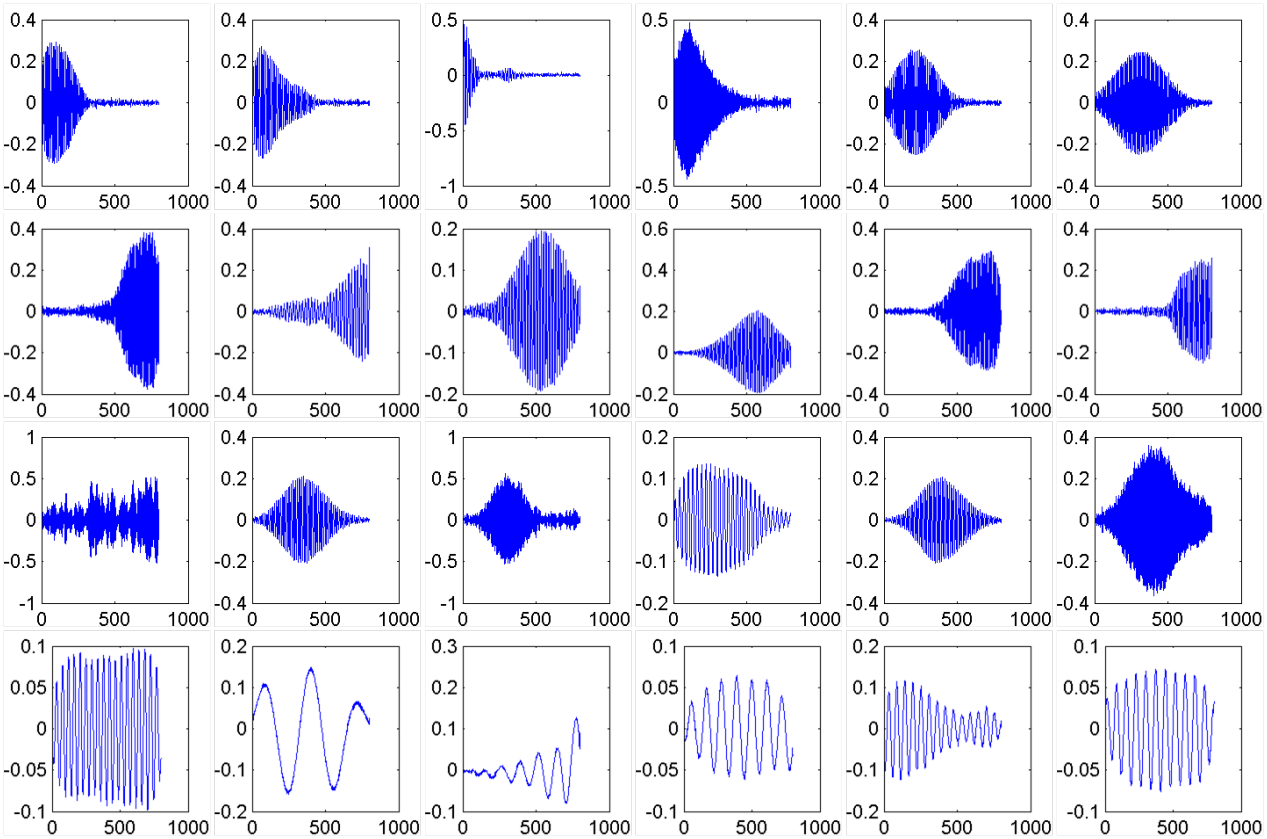


Figure 3.3: 24 out of 600 audio filters learned from TRECVID event collection.

In order to demonstrate that our learned features outperform other classical hand designed features. We use the same bag of word framework as [86] where the code book is generated using K-means and the histogram is classified using SVM with χ^2 kernel. The code book size is set to 4,000. We compare our results on manually annotated 62 action concepts, EC and UCF YouTube dataset, using MFCC [60], MBH [85], SIFT [52] and STIP [37].

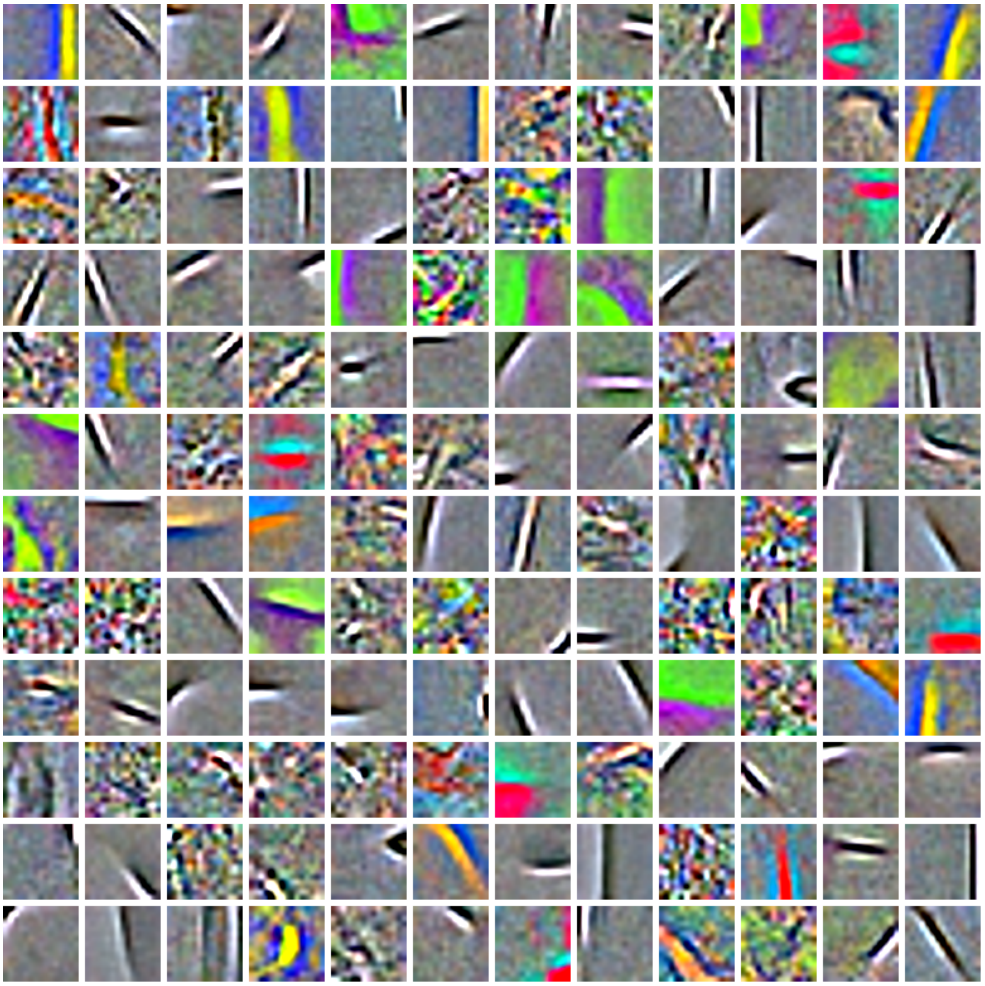


Figure 3.4: 144 out of 600 image (2D) filters learned from TRECVID event collection. Since the training patches have 3 channels (RGB), the learned filters are also with 3 channels. The color information of the filters mainly captures the scene concepts, such as indoor, outdoor.

Table 3.1 summarizes the results. Our learned features work 10% better on average in terms of recognition accuracy than all the other hand designed features, on EC and 62 concepts dataset. The performance of our 3D TICA feature is 20% higher than STIP (30.9%) motion feature on 62 action concepts dataset. The results also show that combining features from different modalities improves the overall accuracy. This suggests that features from different modalities capture complementary information and using features from different sources is necessary.

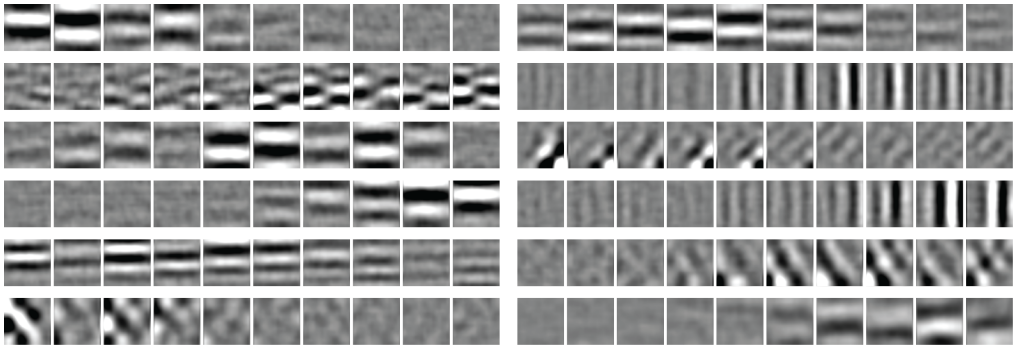


Figure 3.5: 12 out of 600 spatiotemporal (3D) filters learned from TRECVID event collection. Filter size is $20 \times 20 \times 10$. Each row shows two 3D filters.

Table 3.1: A comparison of performance using different features and modality combinations. Our learned features outperform all the other hand designed features on the difficult TRECVID dataset; Combining features from different modalities improves the overall accuracy.

	UCF 11	TRECVID 62 concepts	TRECVID 15 Events
MFCC [60]	×	31.1	34.8
SIFT [52]	58.1	40.3	30.1
STIP [37]	57.5	30.9	41.0
MFCC+SIFT+Dollar	×	×	**51.1**
MBH [85]	**83.9**	36.0	44.0
ISA[41]	75.8	51.3	53.5
TICA 1D	×	31.7	39.7
TICA 2D	56.4	43.3	45.2
TICA 3D	74.3	**53.5**	**55.2**
TICA 2+3D	79.1	57.9	59.5
TICA 1+2+3D	×	58.1	**63.2**

Interestingly, we notice that the learned features are not better than Motion Boundary Histogram (MBH) on UCF YouTube dataset. The reason is presumably that MBH tends to overfit itself to relatively easy dataset, such as UCF YouTube, which contains only well-defined action with relatively simple and clean background. However, its performance plunges by a half to 44% on

difficult dataset TRECVID, where our method yields the highest accuracy of 63.2%. This demonstrates the robustness of learned local features and suggests that feature discovery is important and necessary especially under uncontrolled in-the-wild condition.

3.5.3 Data-driven concept discovery

We trained a five-layer deep belief net using RBM at each layer. The RBMs were initialized with small random weights and zero biases, and trained for 60 epochs using mini-batches size of 100. For the linear-binary RBM we used a learning rate of 0.001. We reduced the learning rate at the beginning of learning when the gradient can be large, and also at the end of learning in order to minimize fluctuations in the final weights. We also used a momentum of 0.8 to speed up the learning.

Fig.3.6 shows the detection results, which evaluates the performance of the clip representation of each layer: after being trained in each layer, clips are grouped into concepts based on the new representation. Then, SVM is used for classification. We first attempt to use K-means directly on the initial clip representation without any dimension reduction on the data. The event detection MAP is 39% based on concept representation. Then we adopt RBM recursively to reduce the data dimension from 4,000 to 100. Fig.3.6 shows that when the dimension of the clip representation reduces from 4,000 to 1000, the event detection MAP increases from 39% to 66% and it reaches the highest point at 1,000 dimension. This supports our assumption that the initial representation of the clip lies in a high dimensional space where Euclidean distance can not measure the true similarity and DBN learns the regularity between the clips correctly. If we keep decreasing the dimension of the clip representation, the accuracy goes down. It means that the high dimensional data is compressed into a too concise space, some useful information maybe lost there. Further, we repeat the same experiments using other manifold learning methods such as PCA, EighenMap and LLE. Figure 5 shows the detection results. It is clear that DBN performs significantly better.

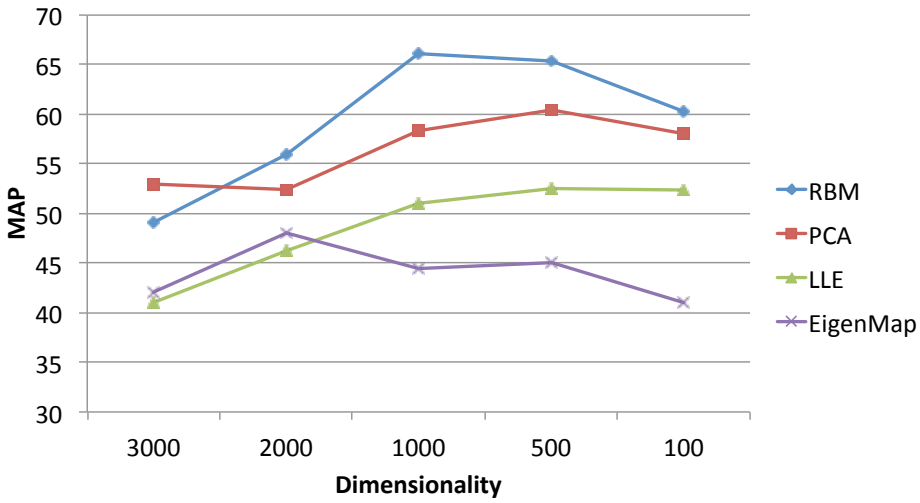


Figure 3.6: A comparison event detection performance as a function of dimensionality of RBM, PCA, LLE and EigenMaps. The accuracy of directly using K-means clustering on 4000 dimension without any dimension reduction is 39%. We compare different dimension reduction techniques. RBMs achieve the best performance of 66% when the number of hidden units is 1,000. Note that the accuracy generally increases, for all techniques, while we reduce the dimension of clip representation, which suggests the necessity of dimension reduction in concept learning.

Fig.3.7 shows the classification results based on a different number of concepts using three modalities and their combination. The results show that motion concepts play an important role in the event detection problem. And a large number of concepts helps the recognition mainly because that larger pool of concepts captures finer level variations of actions, e.g., running action from different viewpoints. However, when the number of concepts increases further, the accuracy drops presumably due to the insufficiency of training video samples, and SVM runs into overfitting.

We observe that, the curve of audio signal (TICA 1D) peaks at 500 concepts, while that of image (TICA 2D) and video (TICA 3D) reach their highest performance much later. This implies that the underlying variation of audio signal is less than that of motion and 2D scene signals, which is consistent with common sense. Combined concepts (TICA 1D+2D+3D) achieve the best results since audio, scene and motion concepts capture complementary information in the videos. We

also use STIP features to run the same experiments. The performance is significantly worse than using TICA feature. This shows that low-level features are important for discovering meaningful data-driven concepts.

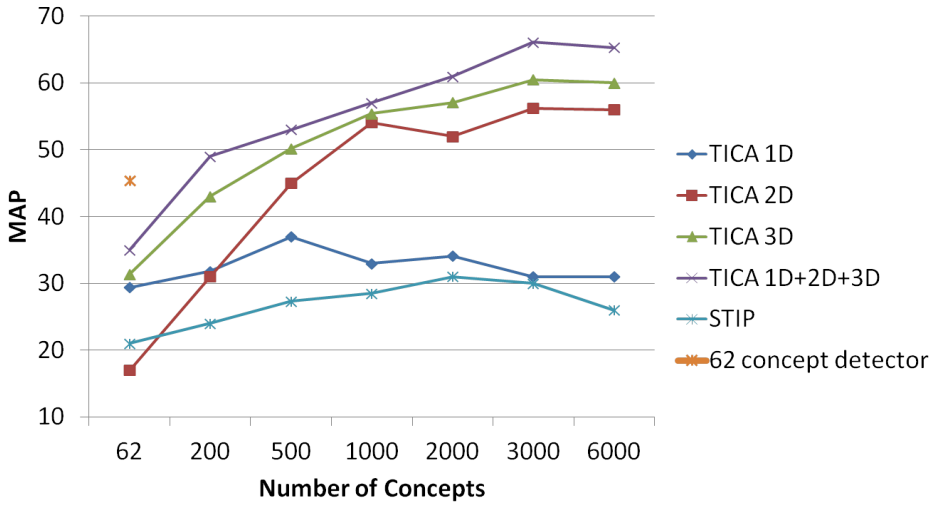
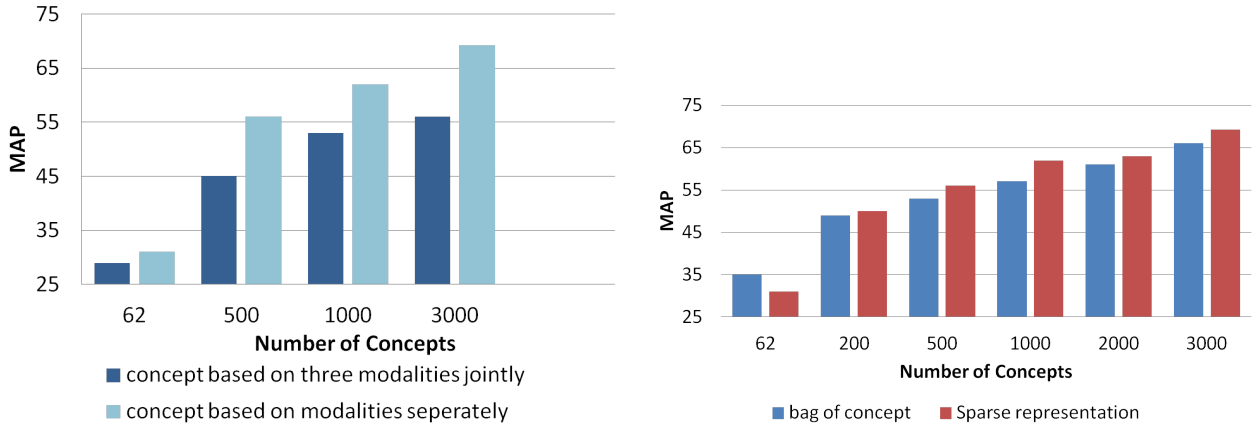


Figure 3.7: A comparison of event detection performance using proposed approach employing audio, image and video features individually and jointly as a function of number of discovered concepts. We also show performance using standard STIP features. Finally, we show the importance of discovered concepts compared to manually annotated 62 action concepts. The results show that a larger number of data-driven concepts improves the detection rate and is better than human annotated concepts.

Interestingly, the model using 62 human-defined concepts trained in supervised manner, outperforms its counterpart using the same number of concepts but discovered in unsupervised way. This suggests more supervision helps when only a small number of concepts are used. However, when the number of concepts increase, data-driven concepts perform better. This shows that a large number of concepts improves the event detection.

We also compare the performance of discovered concepts using early feature fusion of three modalities to the concepts learned from three modalities separately. Fig.3.8a shows that discovering concepts separately is always better.



(a) (b)

Figure 3.8: (a)A comparison of event detection using different number of concepts discovered using three modalities jointly and separately. The former fuse the low level features from three modalities first as the initial clip representation, then learns concepts jointly. The later discovers the audio, scene and motion concepts separately. The results show that using concepts based on the modalities separately outperforms the early fusion one. (b)A comparison of event detection using sparse representation and bag of concept representation. Sparse representation works consistently better than bag of concept representation.

We show the discovered data-driven concepts from 2D scene in figure 3.9. The images in each row are the top candidates chooses from the cluster center, which belongs to one data-driven concepts. We can see that the data-driven concepts in 2D scene can discover the outdoor grass scene, black colored vehicle tires, crowd scene and outdoor snow scene. But some data-driven concepts are really hard to tell the semantic meaning.

3.5.4 Sparse video representation

After concepts are discovered, each long event video can be represented in terms of concepts. Fig.3.8b shows the comparison of the sparse representation and the bag of concept representation, in terms of detection rate. In addition, based on the best results, the detection rate of each category compared with baseline (SIFT + MFCC + STIP) is shown in Fig.3.10. The MAP of our method over 15 events is **68.2%**. In comparison, the MAP of SIFT+STIP+MFCC is 51.1%.

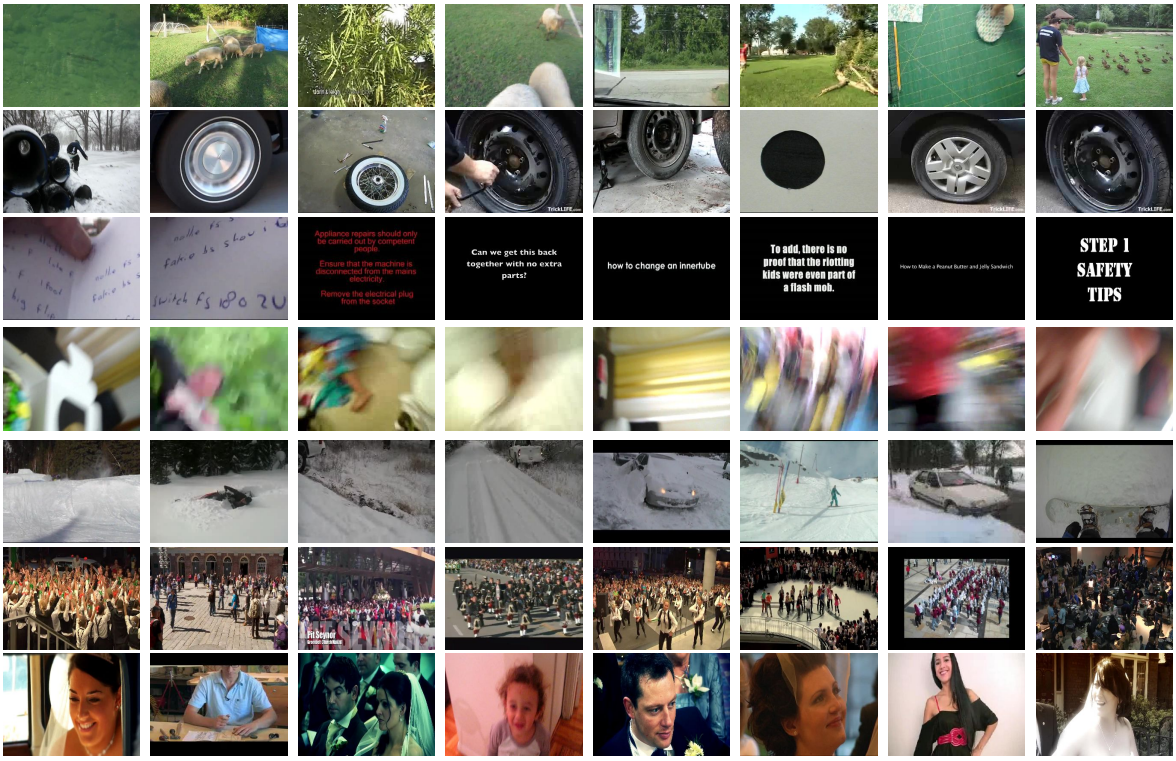


Figure 3.9: The discovered data-driven concepts from 2D scene. The images in each row are the top candidates chooses from the cluster center, which belongs to one data-driven concepts. We can see that the data-driven concepts in 2D scene can discover the grass scene, snow scene and the vehicle tires.

3.6 Summary

In this chapter, we present a three-step approach which learns the sparse video representation based on data-driven concepts from three modalities (audio, image and video) in an unsupervised manner. We use TICA to directly learn the good features from the raw signals, and use RBM to discover the data-driven concepts in an unsupervised way. Through learning the low-level features and clip representation, high-level semantic concepts are discovered. Extensive experiments show that our method significantly outperforms the baselines using human designed features on complex in-the-wild event recognition dataset.

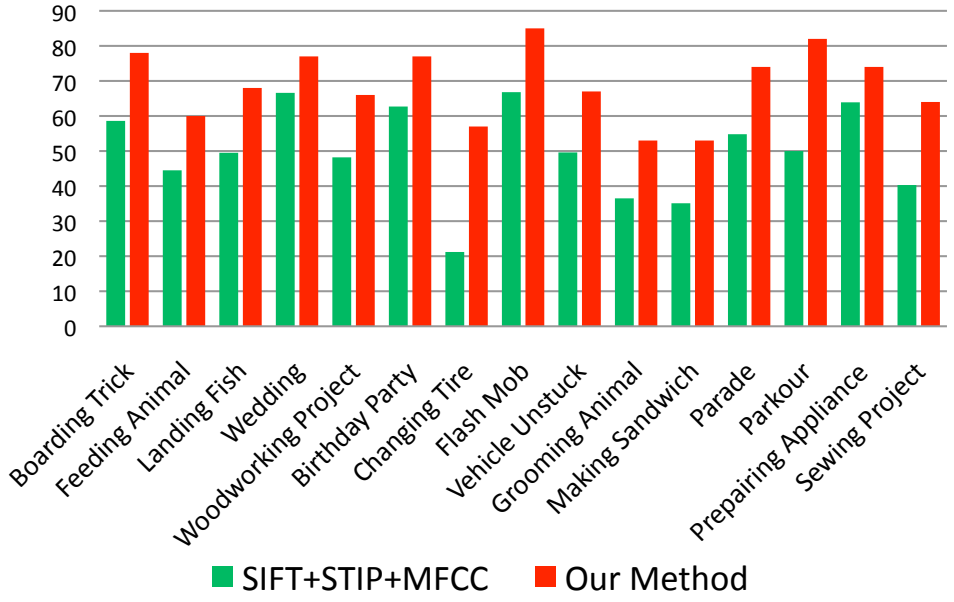


Figure 3.10: Comparison of our proposed method with the combination of MFCC, SIFT and STIP features in terms of detection accuracy on each event category. Our mean average precision is 68.2%. The MAP of combination method is 51.1%.

Theoretically, the low-level feature learning and data-driven concepts can be discovered using the same neural networks. In this work, the reason we choose TICA for low-level feature learning is the invariance property and the faster computational time. As described in chapter 1, the low-level local descriptor is desired to be invariant to certain degree of transformation. In TICA, which is a two layers network, the second layer units pool the neighborhood units of the first layer. By imposing the sparse constraint on the summation of the second units, the neighborhood learned filters are similar but with certain transition or rotation. In another words, the filters in the same neighborhood can detect the edges (in a 2D case) with slight transformations and output the same response by the pooling units. Thus, the feature representation of TICA can achieve the invariance property. The data-driven concepts are discovered starting with the bag-of-word representation of video clips using TICA feature. For the BOV representation, since each bin of the histogram has a particular meaning, we do not need the invariance property in this case. Instead, as the histogram

quantization is always noisy, we aims to denoise the data while achieving dimension reduction. While RBM is a generative stochastic neural network that can learn a probability distribution over its set of inputs and has been shown good performance on image denoising.

The unsupervised learning method is suited for big data, especially when the labels are not available or annotating the huge data is not practical. The learned filters need to be over-complete in order to achieve satisfied performance. However, it will make the feature set huge and the computation expensive. Notice that, in many tasks, few label information is actually available and it can be treated as a cue for finding useful information. For example, if we want to classify human and dog, the discriminative features would be the shape information; if to classify golden retriever and husky, then the color information will be useful. In the next chapter, we will describe our proposed method which utilize the label information to learn a more compact and discriminative feature set for object representation.

CHAPTER 4: SEMI-SUPERVISED LEARNING OF FEATURE HIERARCHIES FOR OBJECT DETECTION IN A VIDEO

In the previous chapter, we have described how to learn multi-modalities low-level features using TICA, and how to discover the data-driven concepts for event recognition using DBN. Both learning are in an unsupervised manner and they are suited for big data where the label is not available or annotating the huge data is not practical. However, the unsupervised learned features only contain the generative property of the input data and lack discriminative property which many features are designed to have. Moreover, the pure generative features always results in a large over-complete set which makes the feature computation process expensive. Notice that, in many vision tasks, few label information is actually available. In this chapter, we introduce our proposed method which imposes the label information into the feature learning model to learn a discriminative and compact set of features. We apply the new model to object detection in videos by learning the hierarchical discriminative representations of objects.

4.1 Introduction

Object detection has been explored extensively and has achieved significant success in the past decade [9, 11, 15, 82, 84, 89]. Most of the state-of-the-art detectors are designed for a single static image and are trained from a large set of labeled examples. The performance of a detector will be inevitably degraded when it is applied to frames in a video taken under conditions which are very different from those of the training examples. Because of the large variation across different environments, a generic classifier trained on extensive datasets may perform sub-optimally in a particular test environment. In general, the construction of appearance-based object detector is time-consuming and difficult because a large number of training examples must be collected and manually labeled in order to capture different variations in object appearance. Therefore, how

to adapt a learned generic detector to the images in a specific video taken under different visual condition becomes a very important problem to be explored.

A large amount of work [3, 98, 16, 19, 29, 57, 70, 69, 47, 70] has been reported on improving object detection in video frames. Several authors [3, 16, 19] propose to improving detection and tracking simultaneously through detection by tracking and vice versa. The detection results serve as a cue to build the tracking results, and the detection component maybe further improved by the result of trackers through online learning. But the improvement will be heavily downgraded if we directly use the noisy detections as initialization of the trackers. Besides detection-by-tracking, researchers have also devoted their efforts on developing online learning/adaptation algorithms for detectors [29, 98, 57, 70, 68, 47, 69]. However, online retraining of the detector is usually hard due to the less training samples and expensive due to the model complexity.

In this chapter, we propose to improving the detection results of a generic detector on a video by refining the detection scores in an offline fashion, without requiring any tracking (trajectories) information or annotation from the video. To achieve this, the original detector with a low detection threshold setting is firstly applied to the frames in the target video. All detected visual examples are collected to form the candidate detection pools using both positive and negative examples pertaining to the target video.

Since selections of the right features plays an important role in object detection, we argue that, the classical hand-crafted features, such as HOG, SIFT, may not be universally suitable and discriminative enough to every type of video. In a particular video, the way objects appear would share some similar properties which could be leveraged to distinguish them from the non-objects. Hence, unlike other proposed methods, which are built on using hand-designed features, we learn the good features directly from the raw pixel of the video itself.

In order to learn discriminative and compact features, we propose a new feature learning method using a deep neural network based on auto encoders. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes both discriminative and generative

properties of the features simultaneously, which gives our features better discriminative ability; second, our learned features are more compact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features. Moreover, we learn a discriminative feature hierarchy from local patches to global images. Extensive experiments with qualitative and quantitative results demonstrate the efficacy of our approach.

The rest of the chapter is organized as follows. The proposed method is presented in section 4.2 in this order: Preliminary, generative feature learning, discriminative feature learning and learning higher levels. Extensive experiment results, comparisons and analysis are reported in section 4.3. Finally, we present a summary in section 4.4.

4.2 The Model

We formulate our problem as a semi-supervised classification problem. First we apply an original detector on a video to get a substantial amount of candidate detections for rescoring. Those detections are initially labeled as confident-positive, confident-negative or hard examples by their confidences. Later we use the confident-positive and confident-negative examples to learn video-specific features. Then, we re-score the hard examples by training a classifier using the learned features. After the rescoring, a small number of hard examples with high confidence are moved into the confident-positive or confident negative sets for next iteration of feature learning. We repeat the above steps until no hard samples become confident ones. In our experiments, it usually converges in 4-7 iterations. The flow chart of the framework is illustrated in figure 4.1.

To learn a set of representative features, we propose to use both a supervised and an unsupervised objective based auto-encoders [83]. We require the representation to be generative, which can produce good reconstructions of the input images, at the same time, to be discriminative, which can give good predictions of the image class labels.

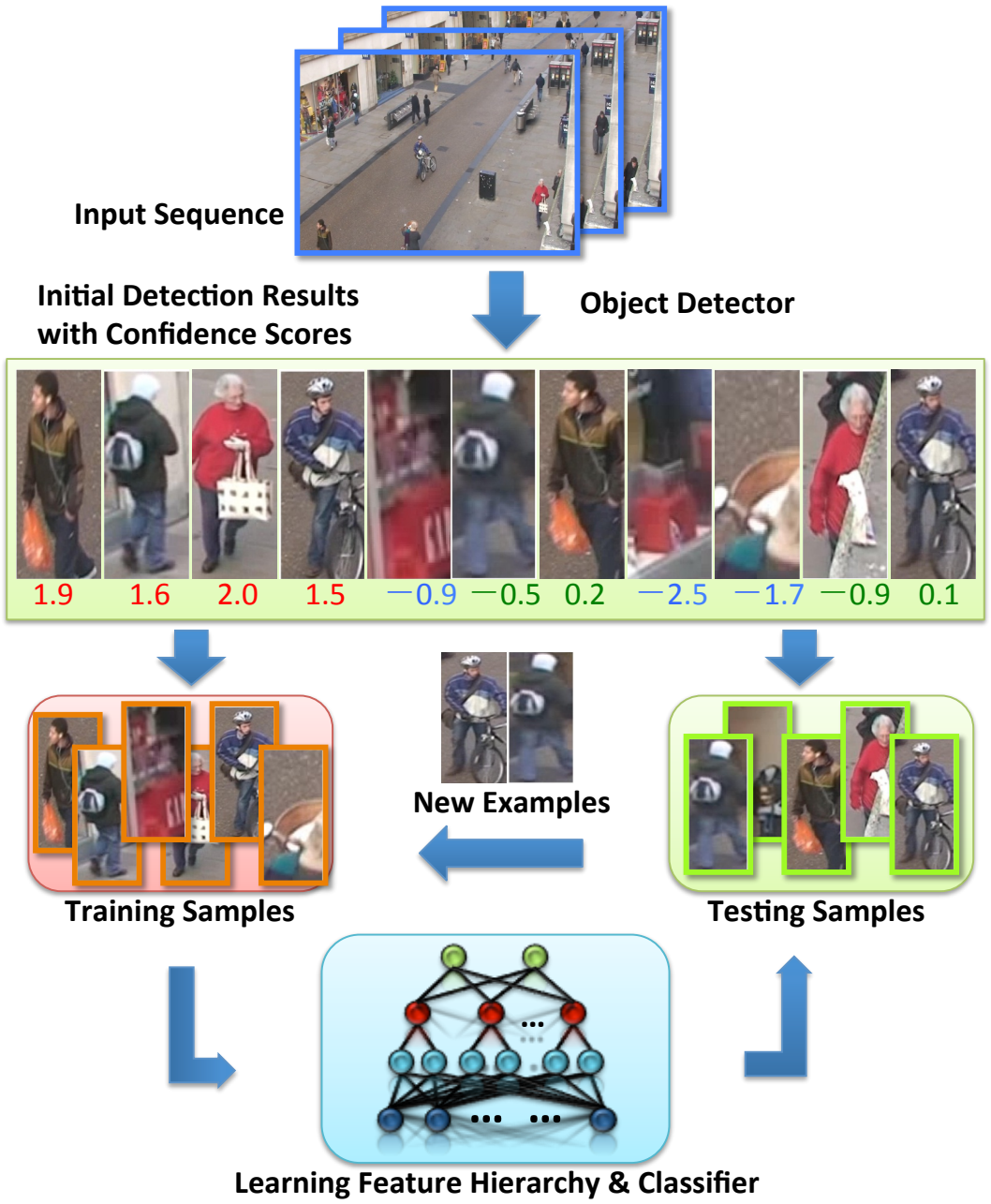


Figure 4.1: The flow chart of our proposed method for video object detection. The training and testing samples are first collected based on the confidence scores given by the object detector. Then the feature hierarchies and classifiers are learned from the training samples and used for re-scoring the testing samples. The testing samples with high confidence scores are included into the training samples iteratively until no testing sample is left.

Further, we learn the feature hierarchies from local to global by increasing the receptive field size (the 2D patch size). Our aim is to capture the local features such as edges with different orientations or color, as well as the global characteristic like the structure and shape. To do so, we stack the the auto-encoders to form a deep network.

In the following part of this section, we will start with introducing the preliminaries about auto-encoders, then move to unsupervised generative feature learning using auto-encoders and discriminative feature learning. Finally, we will describe how to learn higher level features.

4.2.1 Preliminaries: auto-encoders

We start by describing the algorithm for our basic learning module, based on the auto-encoders [83], an unsupervised learning architecture used to pre-train deep networks. Assume we have N randomly sampled local patches $x^{(i)} \in \mathbb{R}^D$ from the training set (the dark blue nodes in figure 4.2), to learn features from them, the conventional auto-encoders attempts to reconstruct the data by minimizing the following loss function:

$$E_{AE} = \sum_{i=1}^{N} \|x^{(i)} - W_2 s(W_1 x^{(i)} + b_1) + b_2\|^2 + Z(W_1 x^{(i)} + b_1), \tag{4.1}$$

where $W_1 \in \mathbb{R}^{N_1 \times D}$ is a weight matrix which maps the visible nodes to hidden nodes, $b_1 \in \mathbb{R}^{N_1}$ is a hidden bias vector, and $s(x) = \frac{1}{1+exp(-x)}$ is a non-linear sigmoid function. $W_2 \in \mathbb{R}^{D \times N_1}$ is a weight matrix which reconstructs the visible node from the hidden node, $b_2 \in \mathbb{R}^D$ is an input bias vector. Z is a regularization function. To simplify the formulation, we use linear activation, no biases and tied weights ($W = W_1 = W_2^T$). Hence, the cost function of auto-encoders can be simplified as:

$$E_{AE} = \sum_{i=1}^{N} \|x^{(i)} - W^T h(i)\|^2 + Z(h^{(i)}). \tag{4.2}$$

where, we let $h^{(i)} = Wx^{(i)}$ as the light blue nodes in the neural network shown in figure 4.2.

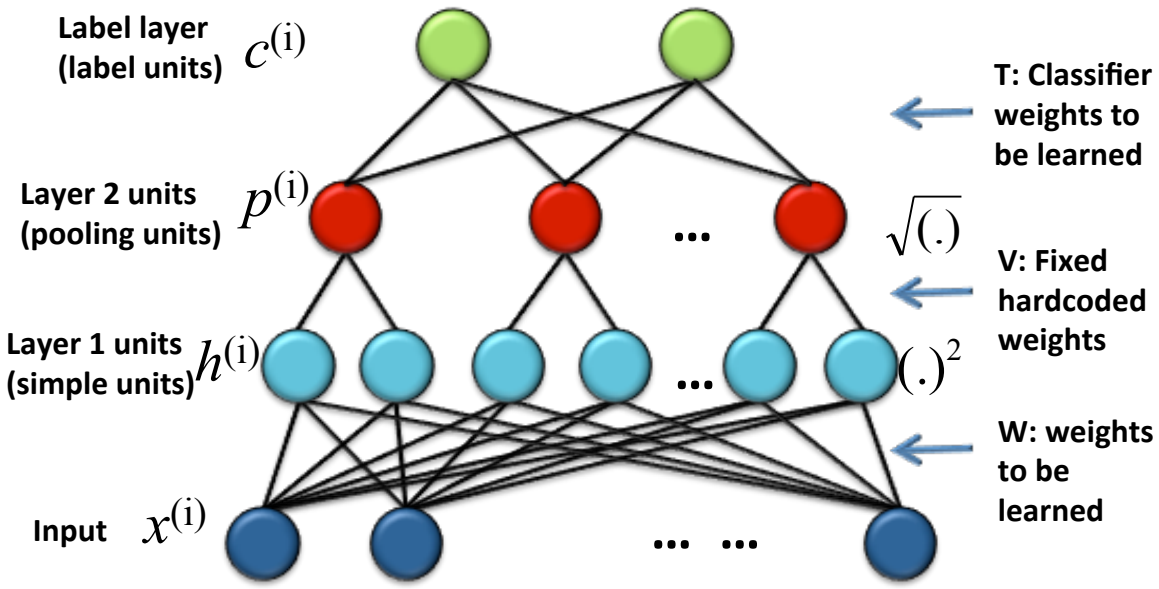


Figure 4.2: The neural network architecture for learning features at one level. Each dark blue node is an input pixel. Each light blue node is one feature response of the corresponding feature(filter) w. The red nodes are the pooling units pooling a non-overlap pair of feature responses(subspace is 2). The green node is the classification label which are used for discriminative feature learning.

4.2.2 Generative feature learning

A generative objective function measures an average reconstruction error between the input x and the reconstruction $x' = W^T Wx$. The underlying idea is that if the model achieves a good reconstruction property, then the representation has preserved most of the information from x. To make the learned features invariant to local transformation, we further impose a second layer on

the top of the auto-encoders by hard coded weights V which pools from several adjacent neurons h, as shown in figure 4.2 the red nodes. We enforce the activation of the second layer to be sparse. The loss function with the second layer pooling unit for unsupervised generative feature learning is as below:

$$E_{gen} = \sum_{i=1}^{N} \|x^{(i)} - W^T h^{(i)}\|_2^2 + \lambda \sum_{i=1}^{N} \|\sqrt{V(h^{(i)})^2}\|_1. \tag{4.3}$$

If we let the second layers activation $p^{(i)} = \sqrt{V(h^{(i)})^2}$, then equation 4.3 can be written as:

$$E_{gen} = \sum_{i=1}^{N} \|x^{(i)} - W^T h^{(i)}\|_2^2 + \lambda \sum_{i=1}^{N} \|p^{(i)}\|_1. \tag{4.4}$$

In this equation, the index i denotes data samples. Square and square-root operations are element-wise here. V is a subspace-pooling matrix with groups of size of two as illustrated in the figure 4.2 (we use four in the experiments). More specifically, each row of V picks and sums two neighboring feature dimensions in a non-overlapping fashion. The last term regularizes for sparsity in the pooling units. This design of pooling units is very similar with Independent Subspace Analysis (ISA) [26] and has the advantage of being able to learn overcomplete hidden representations. As V is hardcoded, we can efficiently optimize the loss function respect to the filter W via stochastic gradient descent.

4.2.3 Discriminative feature learning

The generative feature learning methods intend to learn the features or filters W by minimizing the reconstruction error and learn a set of redundant overcomplete features. However, a

good generative property does not necessarily implies a good discriminative ability. In the experiments, we found out that by randomly picking out some filters learned in a generative way, the classification performance does not drop. It means that not all the features are useful in terms of classification. Moreover, a redundant set of features will increase the computational complexity and we want to avoid it. Notice that we have the collected training set with labeled images. In order to incorporate this information, we add another objective to learn the features. The filters W are now not only learned from reconstructing the input x, but also a classifier predicting the label c from the representation p. A discriminative objective function computes an average classification loss between the actual label $c \in [0,1]^K$ and the predicted label $c' \in [0,1]^K$. More precisely, the loss function is used as a performance measure and we pose an optimization problem as follows:

$$E_{dis} = \sum_{i=1}^{N} \|\text{softmax}(Tp^{(i)}) - c^{(i)}\|_1, \tag{4.5}$$

where $\text{softmax}(a)_k = \frac{exp(a_k)}{\sum_{k'} exp(a_{k'})}, k = 1, ..., K$ for $a \in \mathbb{R}^K$. $c'^{(i)} = \text{softmax}(Tp^{(i)})$. The label c is a binary vector with a softmax unit that allows one element to be 1 out of K dimensions for K-way classification problem.

When the input $x^{(i)}$ is local patches, the label c of $x^{(i)}$ is very hard to obtain since the object and non-object can possibly share the same local patches. To maintain the discriminative property, we can enforce the loss function at the image level instead of each local patch. We perform average pooling on the image from the feature maps of the local patches and the loss function can be modified as:

$$E'_{dis} = \sum_{j=1}^{N_I} \|\text{softmax}(\frac{1}{N_p} T \sum_{t=1}^{N_p} p^{(tj)}) - c^{(j)}\|_1. \tag{4.6}$$

where we sum over N_I training images and the representation of image j is calculated by averaging the activation p of all the patches from image j. We can efficiently learn the features W using stochastic gradient descent. The by-product of this algorithm is the weights T which is leaned jointly with W and we can utilize it in later classification process.

In our semi-supervised classification problem, the labeled data at an earlier stage does not represent the distribution of the whole data. To avoid from overfitting by the discriminative loss function, we further combine the discriminative and generative loss function to learn the discriminative features as follows:

$$E = E_{gen} + \beta E'_{dis}, \tag{4.7}$$

where β is a coefficient balancing E_{gen} and E'_{dis}. The first term is very common to most unsupervised learning algorithm. It makes the system model the structure and the dependencies among the input components of x. The second term represents the supervised goal. It ensures the features are also going to be good for discriminating between class. In the rest of the paper, for simplicity, we will call the features learned by equation 4.7 as discriminative ones and equation 4.4 as generative ones.

4.2.4 Learning higher levels

We learn the features W from small image patches (small receptive field size) sampled from the training images at the beginning as the first level. Each feature in the first level is capturing local edges or color information. However, we expect to learn a more complex set of features, which can capture the conjunction of edges or even a global structure of the object, within a larger receptive field. To learn the higher-level features, we adopt a convolutional neural network architecture [79, 42] that progressively makes use of auto-encoders as sub-units as shown in Figure 4.3. The key ideas are as follow. We learn the first level filters by minimizing equation 4.7 on small input

patches. Then we use the learned m filters to convolve with a larger region of the input image to obtain m feature maps. The max pooling operation is then performed over a certain neighborhoods. We can therefore extract local patches from these locally-invariant multidimensional feature maps and feed them to another level which is also implemented by auto-encoders. In our experiments, the stacked model is trained in a greedy manner layerwise in the same manner as other algorithms proposed in the deep learning literature [24]. More specifically, we train the first level features until convergence before training the second level.

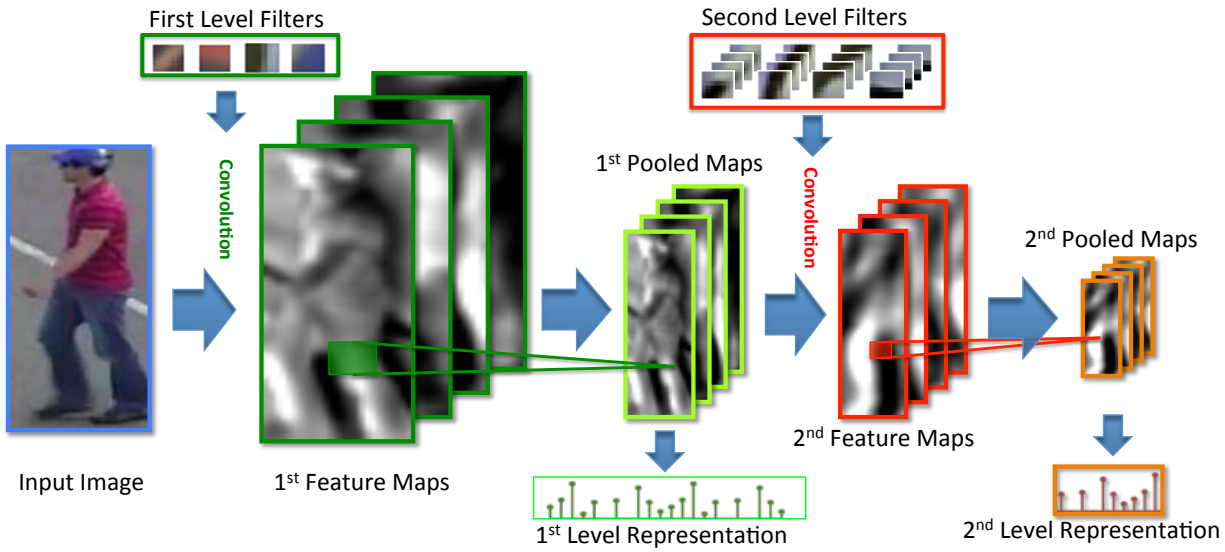


Figure 4.3: Object representation using a 2-level model. The first level learns features from color local patches using proposed algorithm. The feature maps are obtained by convolving each feature with the input image. Each feature map is then pooled from a 4×4 pixel non-overlapping grid to generate the pooled map. The concatenation of the pooled map serves as the representation of that level. The only difference of the second level with the first level is that: the features in the second level are learned from the pooled local feature maps, instead of the larger local patches from the input images.

4.3 Experiments and discussion

4.3.1 Experimental setup

We extensively experimented on the proposed method using four benchmark dataset: Oxford Town Center dataset [5], PETS2009 Dataset, PNNL Parking Lot datasets [76] and CAVIAR cols1 dataset [1] for human detection. Besides, we collected three videos from YouTube for horse detection. The frame resolution of the three videos is 450×360 and each video length is around 5 to 10 minutes. The number of frames containing horses is around 3000. We manually annotated the dataset. The challenge is the clutter background, occlusion and various poses of the horse. In all the sequences, we only use the detector scores and do not use any tracking results nor any annotation from the video.

In our experiments, we use the pre-trained pedestrian and horse model from [14]. We set a high recall and low precision point for this method in order to obtain almost all true detections and many false alarms. According to the detector confidence, we divide all detections into two groups: the ones with confident above a threshold are the positive examples; and the rest are hard examples and will be classified later. All the examples are resized to 128×64.

We learn three levels of features to represent the images. The first two levels are learned from 8×8 pixel wise patches from the input image and the first level pooled feature map respectively. The third level is learned directly from the second level pooled feature maps. The number of filters at each level is set as $m = 400$, the subspace size is 4. The number of feature maps at each level is therefore 100. The connected pooled value from each grid of each feature map serves as the final representation of that level. The final image representation is the combination of the three levels as illustrated in Figure 4.3. A linear SVM classifier is trained from the training set on the learned image representation and used for re-scoring the testing images. We adopt the evaluation criterion of PASCAL VOC challenge [12]. A detection is treated as a true positive if it has more than 0.5 overlap with the ground truth. We report the detection average precision (AP) to compare

the performances.

In the following subsections, we first report our human detection performance, then compare the generative and discriminative features quantitatively and qualitatively. The features at each level are then analyzed based on the detection performance and we report our horse detection results in the end.

4.3.2 Human detection performance

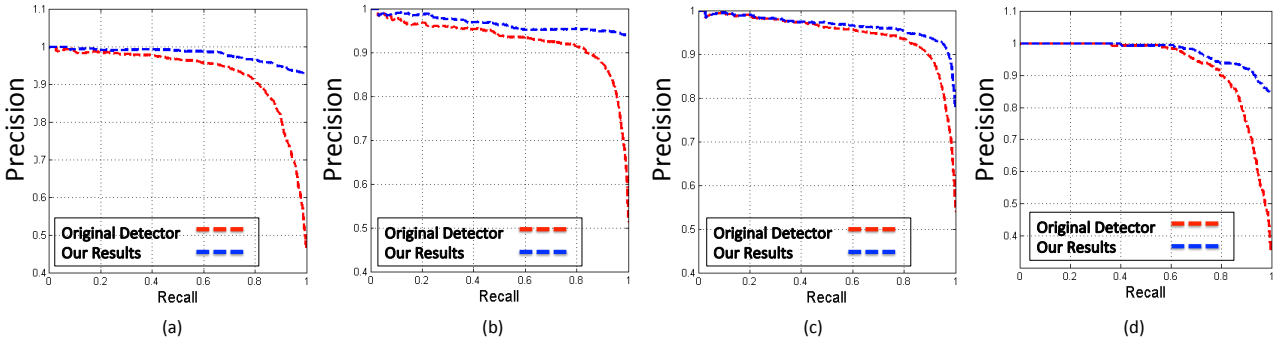


Figure 4.4: The precision-recall curve of four human detection datasets (a. TownCenter b. ParkingLot c. PETS09 d. CAVIAR). The red curves show the standard detector results and the blue curves show our results.

We first evaluate our proposed method in terms of the human detection performance. The precision recall curve are shown in figure 4.4, where the red curves show the standard detector results and the blue curves show our results. Note that although the original detection results already had a sharp drop in precision near the maximum recall, our algorithm is still able to push the curve up. The AP is reported in the first and third row of table 4.1. Overall, our proposed method improves the generic offline detector's results 3-5% on the four benchmark human detection datasets. Further, we compared our learned feature with the classical hand-designed HOG feature. We use the same re-scoring pipeline and just replace the feature learning algorithm with the HOG feature. The AP is shown in table 4.1 second row. The average improvement is around 3%.

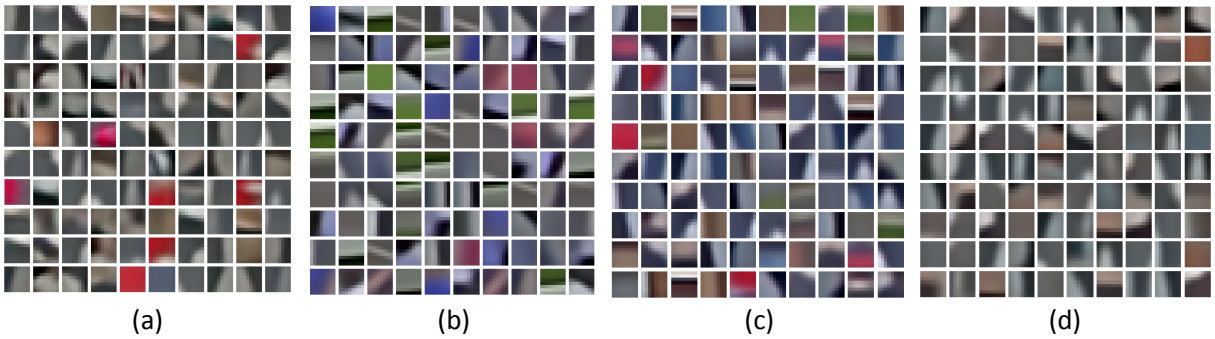


Figure 4.5: The 100 filters learned discriminatively from four human detection datasets (a. TownCenter b. ParkingLot c. PETS09 d. CAVIAR). The filters are visually different, especially in color, since each is learned from a specific video.

One of the advantage of our method is that it learns a specific discriminative compact set of features from the data itself, instead of using the combination of different classical hand-designed features. We show the final learned discriminative features from each dataset in figure 4.5. (a)-(d) are the corresponding learned first level features from TownCenter, ParkingLot, PETS09 and CAVIAR. As we can see that, the four set of features are visually very different from each other. Each captures the specific representative color and edge information in the corresponding dataset. We argue that using the learned features is more efficient and effective, especially in videos. In contrast to boosting, which selects the good features from a pre-defined feature pool, our method dynamically selects and learns the good features from the raw pixels.

4.3.3 Discriminative vs. generative

As described previously, the features learned in a generative manner are usually over-complete, which are good for reconstruction, but are not necessarily effective for recognition. Hence, we propose to directly learn the discriminative features for particular video by adding the discriminative loss function. We train the classifier and the filters at the same time to find out which features are good for recognition. To qualitatively visualize the differences of the two set

of features, we show the features generatively learned from TownCenter and CAVIAR dataset in figure 4.6. By observing the original video, we found that color information is more distinguished for detecting a person in the TownCenter than CAVIAR. Comparing figure 4.5(a) with 4.6(a), interestingly, the color information is more emphasized by the discriminatively learned features. Comparing figure 4.5(b) with 4.6(d), the color information is now more emphasized by the generative learned feature, since most of the negative examples are the colored background and most of the people are wearing dark clothes.

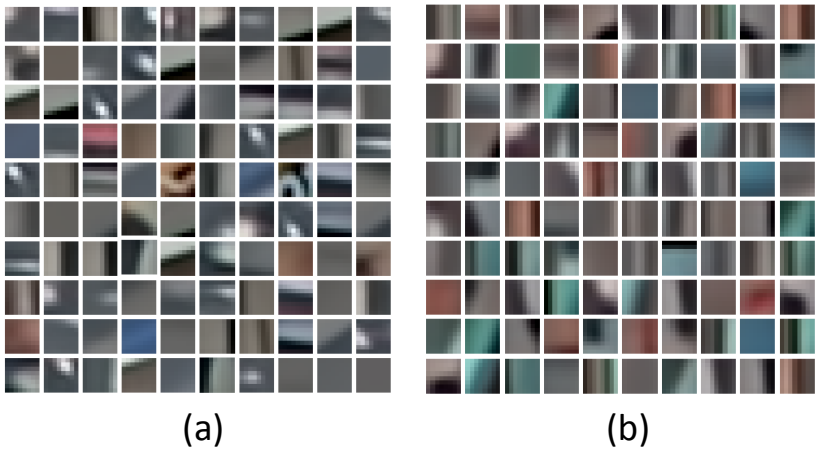


Figure 4.6: The 100 filters learned generatively from Towncenter(a) and CAVIAR(b) dataset. Compared with the corresponding discriminative filters (a) and (d) in figure4.5, the generative features are quite different especially in color.

To quantitatively measure the generatively and discriminatively learned features, we compute the Average Precision (shown in table 4.1 third and fourth row) of the two sets of features on the four dataset. On average, the discriminative learned features are 2% better than the generative learned features. Further, we show that the discriminative learned feature set is more compact than the generative learned feature. We compute the AP by increasing the learned features at each level from 40 to 800, the results are shown in figure 4.7. Interestingly, we found that the discriminative

features reach the highest average precision when 400 features are learned. More features do not improve further in terms of the classification accuracy. Whereas, the generative learned features do need a large set of over-complete features to capture enough discriminative information. This demonstrate that our proposed method can not only improve the classification accuracy, but also boost the computational efficiency.

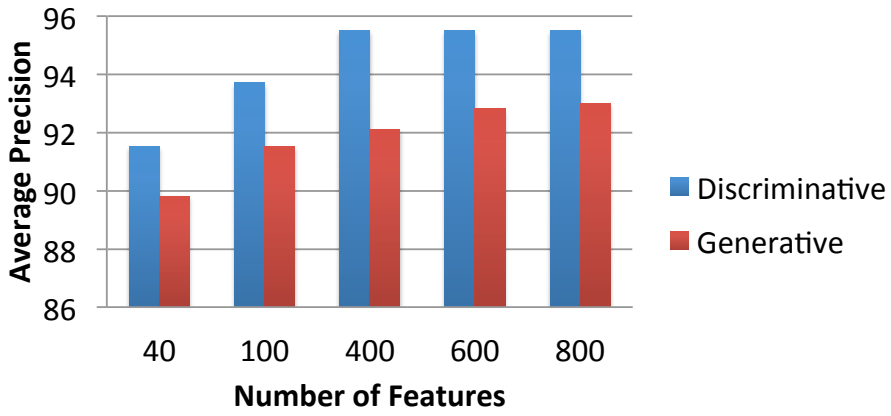


Figure 4.7: The average precision of discriminative and generative method over different number of feature. Given a target AP, the discriminative method reaches it with less number of features. This means the discriminative method gives more compact features than the generative one method.

4.3.4 Analysis at each level

As described in section 4.2 equation 4.6, the by-product of our learning algorithm is the weights T, which can be also used as a classifier in our task. In table 4.1 (last four rows), we show the performance of each level and their combination in a late fusion manner, when we only use weights T as a linear classifier. As we can see that the first level plays an important role in terms of classification accuracy. The performance of the second and the third levels are lower than the first level, but the combination of them performs the best.

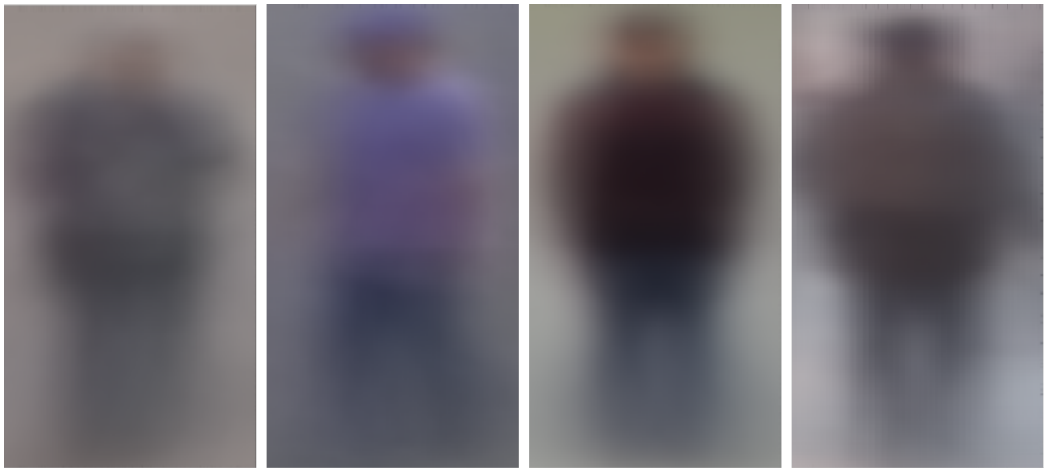
Table 4.1: The average precision of different methods or experimental setups on four benchmark datasets for human detection. The first row is the results from a generic detector. The second row is using the the same re-scoring process but HOG feature without our feature learning algorithm. The third row is the results of the proposed discriminative features. The fourth row is the generatively learned feature results. Overall, our proposed algorithm improves the detection results of the generic object detector by 5% and the HOG features by 3%.The last four row are the detection results using the by-product weights T of our method for re-scoring instead of training SVM on each level.

	Town Center	Parking Lot	PETS 09	CAVIAR
Detector [14]	91.2	91.3	93.5	91.1
HOG	92.1	92.9	94.5	92.5
Discriminative	**95.4**	**96.9**	**95.5**	**94.3**
Generative	94.3	94.1	93.7	93.4
first level	94.2	95.1	94.5	92.9
second level	93.7	93.2	93.1	91.6
third level	92.1	93.5	92.7	90.8
1+2+3 level	**94.6**	**96.3**	**95.5**	**93.1**

We further show the high-level representations learned from each dataset using our model. The representations are visualized by averaging all the input samples which have high responses on the label node $c^{(1)}$. Figure 4.8 (a)-(d) shows the high-level representations for Town Center, Parking Lot, PETS09 and CAVIAR dataset.

4.3.5 Performance on horse detection

To demonstrate the generality of our method, we performed further experiments for another object, horse, on the videos collected from YouTube. All the quantitative results are shown in table 4.2. The generic offline trained horse detector performs averagely 57% on the dataset, whereas, our approach achieves significantly better results than the original detector. Overall, we improve the AP by 7%. The high-level representation learned from horse videos is shown in figure 4.9.



(a) (b) (c) (d)

Figure 4.8: High-level representations learned from each dataset using our model. (a) Town Center, (b) Parking Lot, (c) PETS09 and (d) CAVIAR.

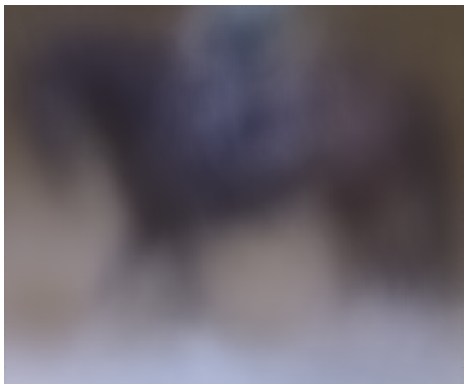


Figure 4.9: High-level representations learned from horse videos.

4.4 Summary

In this chapter, we present a method to learn the feature hierarchies for objects directly from the raw pixels in a particular video to improve a generic detector. Instead of learning the features in an unsupervised manner, we consider the discriminative property of features, and simultaneously learn discriminative and reconstructive features by using both a supervised and an

unsupervised objective. In this way, the learned discriminative features are more compact than the pure generative features. Extensive experiments results demonstrate the efficacy of our approach.

Table 4.2: The average precision of different methods or experimental setups on three horse videos. Overall, our proposed algorithm improves the detection results of the generic object detector by 7% and the HOG by 5-6%.

	Horse1	Horse2	Horse3
Detector [14]	51.2	56.5	63.4
HOG	53.1	58.1	66.6
Discriminative	**59.6**	**64.9**	**71.2**
Generative	54.7	61.3	67.1
First level	56.2	62.1	68.9
Second level	56.6	62.9	66.3
Third level	56.1	62.2	67.2
1+2+3 level	**58.7**	**64.1**	**69.4**

As our model adapts to the video in a semi-supervised way, the re-scored confident examples are used to learn the new model iteratively. The common issue for this kind of framework is the drifting problem, where the model learns from some wrong labeled examples and give the wrong scores to the hard examples, the accumulation of small errors results in a drift away from the target object. In this work, we address the drifting problem by avoiding over-fitting of the learned model using the generative feature learning objective. It regularize the learned model on both generative and discriminative property.

As described in the previous two chapters, we have proposed to learn the hierarchical representations from single input. In the next chapter, we propose to learn the relational feature pairs from two input videos for action verification.

CHAPTER 5: JOINTLY LEARNING FEATURES AND METRICS FOR MEASURING ACTION SIMILARITY

5.1 Introduction

Measuring the similarity of two human actions is an important task with many applications. It is challenging since matching video pairs is intimately tied to the invariance modeling: two videos are the same if they are invariant under some classes of allowable transformations.

The task of modeling invariances has received a fair amount of attention in the past. Invariances can be modeled at two levels. The first is the feature level. Researchers in action recognition have been proposing various descriptors [9, 52, 10, 85, 40, 38] trying to encode different types of invariance of view points, illumination changes, camera motions, etc. The descriptors are further used to build action representation followed with certain predefined distance metrics, such as Euclidean, χ^2 and histogram intersection. The second is the metric level. Once the feature representation is designed and fixed, metric learning [93, 73, 4, 80, 58, 7] can be used to learn the similarity distances between video pairs. The aim of learning the metric is to find the best distance measurement in high dimensional feature spaces for a specific task. However, as the features and metrics are tightly inter-depended to each other, designing them separately will likely degrade the overall performance.

In this work, we propose to learn the similarity metrics and the feature representations jointly. Figure 5.1 illustrates the proposed model. More specifically, we learn the spatial-temporal feature pairs and multiple metrics which can model the complex action transformations. In this way, the features and the metrics will co-operate to achieve an optimal solution.

Oftentimes two distinct actions share the same scene background (this happens a lot in sport videos). Existing generative feature learning approaches [40] tend to be distracted by the common scene instead of learning discriminative features to tell apart the actions. In order to

improve the discriminative ability of the learned features, we propose a new learning method using both generative and discriminative objectives based on gated auto encoders [54].

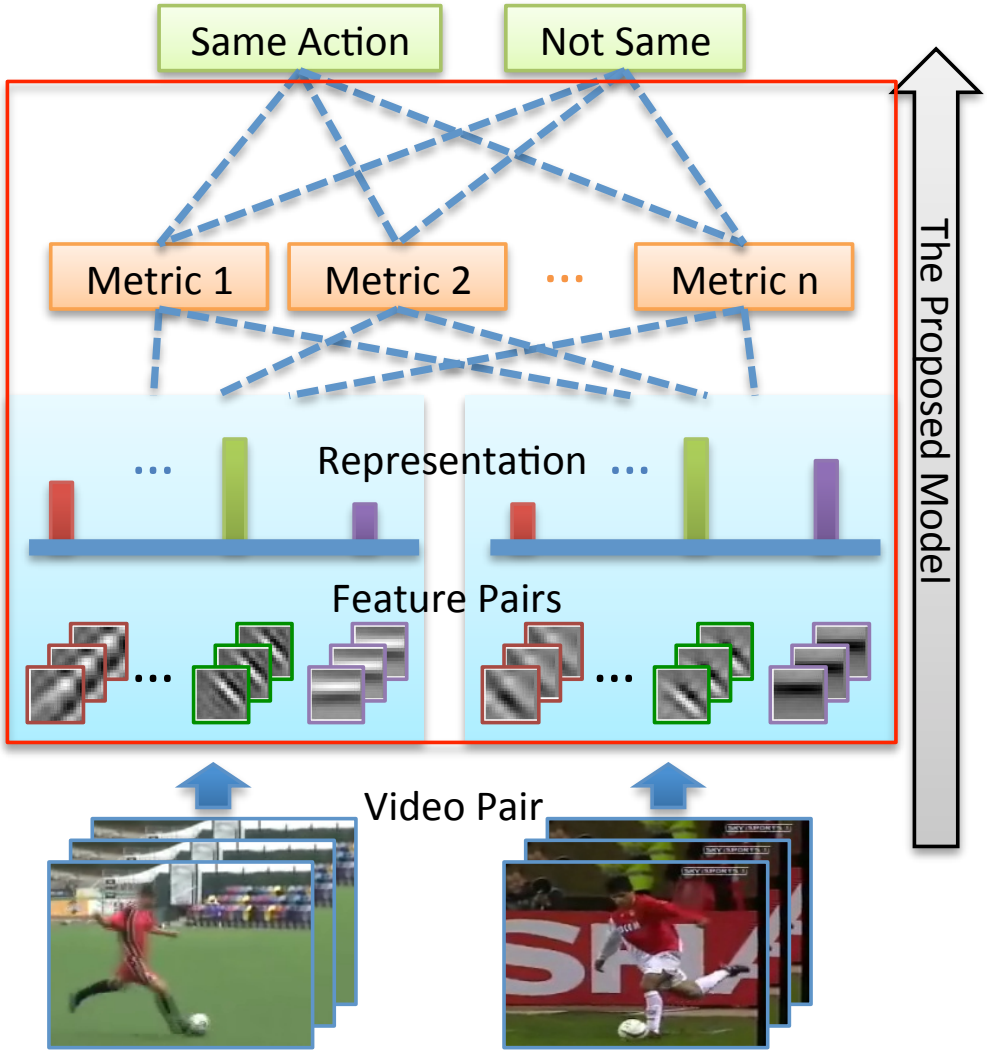


Figure 5.1: An illustration of the proposed model. The model (as shown in the red rectangle) learns the features and metrics simultaneously. The features are discovered as the spatial-temporal feature pairs which find the similarity between two videos. We show three pairs of features in red, green and purple in the illustration figure. The model learns multiple metrics to model the complex transformations between two videos. From the multi-metrics output, we can further build a classifier to get the final labels which tell whether the two input videos contain the same action or not.

Our model differs from the existing methods in two ways: first it learns the features and metrics jointly, while others either fix one of them or learn them separately; second, it optimizes both discriminative and generative properties of the model simultaneously, which gives our model better discriminative ability. Extensive experiments with qualitative and quantitative results on various tasks of action verification, k-shot learning and large scale action recognition, demonstrate the efficacy of our approach.

The rest of the section is organized as follows. The proposed method is presented in section 2 in this order: gated auto encoder, discriminative learning. Extensive experiment results, comparisons and analysis are reported in section 3. Finally, we conclude in section 4.

5.2 The model

The proposed model is illustrated in figure 5.2. Given a video pair x and y together with the "same or not" label c, the model learns the features U, V, the metrics Z and classifier T simultaneously by minimizing a hybrid objective function. The hybrid function is composed of a generative term and a discriminative term. We require the model to be generative, which can produce good reconstructions of the input videos, at the same time, to be discriminative, which can give good predictions about whether two videos are the same or not. We will describe the proposed model in details in the following subsection. We start with introducing gated auto-encoders for pair matching, and specify how we deal with video pair. Then, we will describe how to learn features and metrics discriminatively.

5.2.1 Gated auto encoder

As described in previous chapters, Auto-encoders [83] have been used widely as a basic learning module. They are an unsupervised learning architecture used to pre-train deep networks. The underlying idea of this module is to minimize the reconstruction error of the input from the

network. More specifically, suppose we have N_s randomly sampled local patches $x^{(i)} \in \mathbb{R}^{Dx}$ from the training set (shown as the left red nodes in figure 5.2), to learn features from them, the conventional auto-encoders attempts to reconstruct the data by minimizing the following loss function:

$$L_{AE} = \sum_{n=1}^{N_s} \|x^{(n)} - U_2 s(U_1 x^{(n)} + b_1) + b_2\|^2 \tag{5.1}$$

where $U_1 \in \mathbb{R}^{N_f \times Dx}$ is a weight matrix which maps the visible nodes to hidden nodes, $U_2 \in \mathbb{R}^{Dx \times N_f}$ is a weight matrix which reconstructs the visible node from the hidden node. $b_1 \in \mathbb{R}^{N_f}$ is a hidden bias vector, $b_2 \in \mathbb{R}^{Dx}$ is an input bias vector. $s(x) = \frac{1}{1+exp(-x)}$ is a non-linear sigmoid function. To simplify the formulation, we ignore the regularization term and use linear activation, no biases and tied weights ($U = U_1 = U_2^T$). Hence, the loss function of auto-encoders can be simplified as:

$$L_{AE} = \sum_{n=1}^{N_s} \|x^{(n)} - U^T f^{(n)}\|^2 \tag{5.2}$$

where, we let hidden units $f^{(n)} = Ux^{(n)}$ as the left light blue nodes in the neural network shown in figure 5.2. From $f^{(n)}$, we can reconstruct back $x'^{(n)}$ using $x'^{(n)} = U^T f^{(n)}$.

From equation 5.2 we see that the auto-encoder models the relationship between the input units X and the hidden units F by minimizing the reconstruction error. The learned weights U serve as the filters (features) which will be used in the feature extraction process once the learning is done.

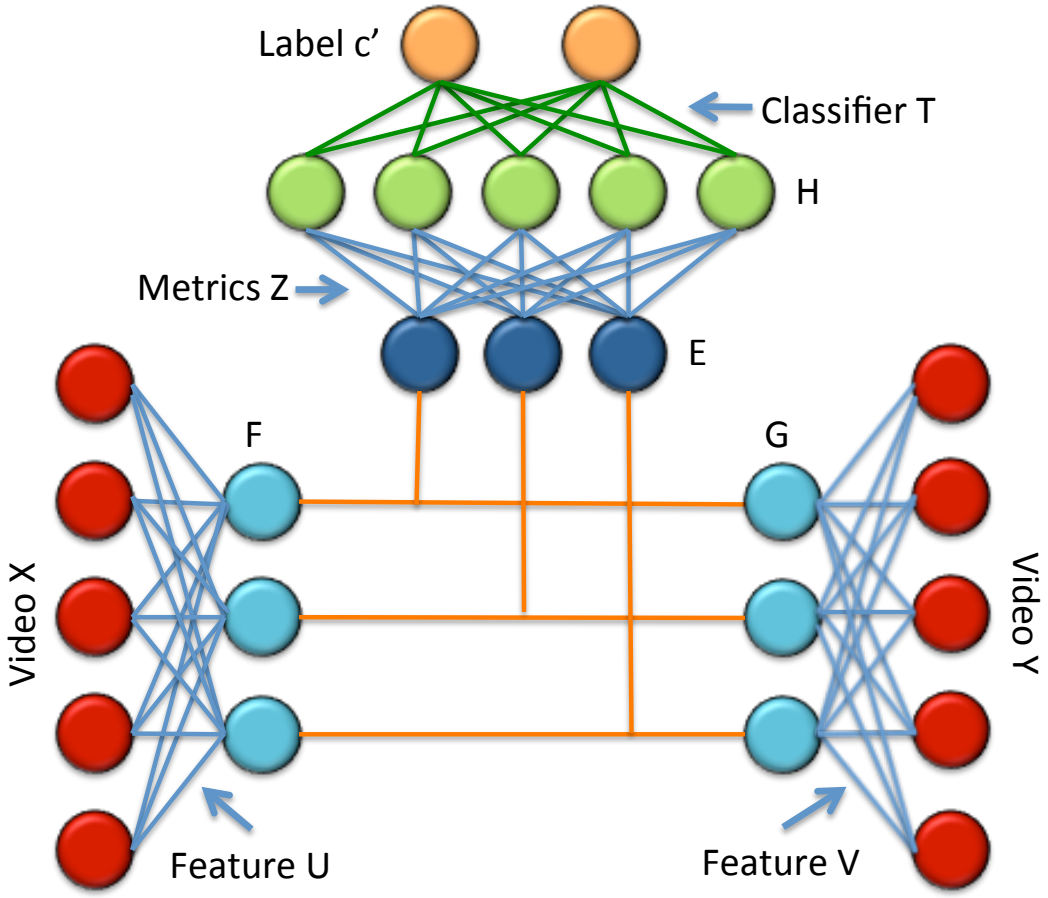


Figure 5.2: An illustration of the proposed neural networks. Video X and Y are the input video pair. c' is the predicted label telling whether X, Y are the same action. U, V are the learned feature pairs. F, G are the feature representation of video X, Y. Z is the multi-metrics learned together with U, V. H is the hidden unit. T is the learned classifier. E is computed by the element-wise multiplication of F, G.

The conventional auto-encoders models individual image x which only emphasizes on the content presented in x itself. When dealing with video pairs ($x^{(n)} \in \mathbb{R}^{Dx}, y^{(n)} \in \mathbb{R}^{Dy}$), our interest is not only in each individual, but more about the relationship between x and y. By relationship, here, we mean the complex transformations which make two videos the "same" (same action) though the appearance can be very different (different view point, background, etc.). Hence, instead of using the hidden units to model what kind of content is presented in individual x, we want

to use them to model what kind of relationship or transformations does the pair x and y have.

One can think that each hidden unit can contribute a "basis transformation" to model the overall dependency between x and y. The activation of hidden units, now, is not only depend on x, but also on y. In other words, one can think of the outputs x as a function of the input image y, in that different inputs y give rise to different transformation over x. Hence, we can define an auto-encoder as a conditional model of x given y, called as gated auto-encoder. The kth hidden unit activation is:

$$h_k(x; y) = \sum_{j=1}^{Dy} \sum_{i=1}^{Dx} w_{ijk} x_i y_j, \tag{5.3}$$

where $W \in \mathbb{R}^{Dx \times Dy \times Dh}$ is the learned model. Dx, Dy and Dh are the dimension of x, y and h respectively. This shows that in the conditional model, hidden variable activities are now given by a simple basis expansion of x and y. Similarly, the output x' are given by a basis expansion of y and h:

$$x'_i(h; y) = \sum_{k=1}^{Dh} \sum_{j=1}^{Dy} w_{ijk} y_j h_k. \tag{5.4}$$

Further, we can simplify this model by factorizing W [56] and we can rewrite equation 5.3 as the following:

$$h^{(n)} = Z^T(f^{(n)}) \cdot (g^{(n)}). \tag{5.5}$$

where $\cdot$ indicates the element-wised multiplication, and $f^{(n)} = U^T x^{(n)}, g^{(n)} = V^T y^{(n)}$. As shown in figure 5.2, U, V are the filters (features) applied on x and y respectively. Z is the multi-metrics related to each hidden unit h. To encode the relationship between x and y, the hidden units h

correlate x and y by using element-wise products (shown in fig 5.2 orange line) between the filter response f, g of x and y as inputs to the hidden variables. The reason for using multiple hidden units is to model complex transformations. Different y causes different h.

Similarly, given hidden units h and y, the reconstructed x' (equation 5.4) are computed as

$$x'^{(n)} = U^T(e^{(n)}) \cdot (g^{(n)}), \tag{5.6}$$

where $e^{(n)} = Z^T h^{(n)}$.

To train the parameters, we can deploy the standard learning criteria, minimize the reconstruction error using gradient-based optimization on the loss function:

$$L_{GAE} = \sum_{n=1}^{N_s} \|x'^n - x^n\|^2. \tag{5.7}$$

In our verification task, we are interested in the joint distribution over x and y as oppose to the conditional one. As x and y can be interchangeable in the pair matching problem, in practice, we train the model symmetrically by reconstructing both y from x and x from y. The overall objective function as the sum of the two asymmetric objectives is defined as:

$$L_{GAE-sym} = \sum_{n=1}^{N_s} \|y'^n - y^n\|^2 + \sum_{n=1}^{N_s} \|x'^n - x^n\|^2. \tag{5.8}$$

When the input pair are videos, a global transformation on x and y is not practical since the input dimension is extremely high. To avoid it, we adopt the convolution neural network architecture, which shares the weights (features) across space and time in the video. Specifically, we divide the video x into overlapped spatial-temporal cuboids $x_c^{(t)}$ and learn features U from the cuboids (U are the same size with $x_c^{(t)}$ now). Then average pooling is performed from the feature

maps of the video. Hence, the activation of units f can be rewritten as:

$$f = \frac{1}{N_c} \sum_{t=1}^{N_c} U^T x_c^{(t)}, \tag{5.9}$$

where N_c is the number of cuboids in x. Similarly, the activation of units g for video y is:

$$g = \frac{1}{N_c} \sum_{t=1}^{N_c} V^T y_c^{(t)}. \tag{5.10}$$

In the experiments, we can efficiently learn the parameters using stochastic gradient descent.

5.2.2 Discriminative learning

So far, the features U, V and metrics Z of the model are learned by a generative loss function which measures an average reconstruction error between the input x, y and the reconstruction x', y'. The reason is that if the model achieves a good reconstruction from the model, then we can be sure that the representation has preserved most of the information from original signal. However, a good generative property does not necessarily implies a good discriminative ability. In our case, we want to focus on learning action similarity, rather than the distinguishing features of particular actions. Notice that we have the label information of video pairs, in order to incorporate this information, we need to add another objective into the model. The model is now not only learned from reconstructing the input x, y, but also from a classifier predicting the label c from the transformation activation h. A discriminative objective function computes an average classification loss between the actual label $c \in [0, 1]^2$ and the predicted label $c' \in [0, 1]^2$. More precisely, the loss function is used as a performance measure and we pose an optimization problem as follows:

$$L_{dis} = \sum_{n=1}^{N} \|\text{softmax}(Th^{(n)}) - c^{(n)}\|_1, \tag{5.11}$$

where softmax$(a)_k = \frac{exp(a_k)}{\sum_{k'} exp(a_{k'})}, k = 1, 2$ for $a \in \mathbb{R}^2$ (since in our case there is only same or not classes). $c'^{(n)} = \text{softmax}(Th^{(n)})$. The label c is a binary vector with a softmax unit that allows one element to be 1 out of 2 dimensions for binary classification problem. T is the classifier to be learned.

To avoid from overfitting by the discriminative loss function, we further combine the discriminative and generative loss function to learn a hybrid model as follows:

$$L_{hyb} = L_{GAE-sym} + \alpha L_{dis}, \tag{5.12}$$

where α is a coefficient balancing $L_{GAE-sym}$ and L_{dis}. The generative term makes the system model the structure of the input components of x, y. The generative term ensures that model are also going to be good for discriminating the similarity between actions. In the rest of the paper, for simplicity, we will call the model learned by equation 5.12 as hybrid and the one learned by equation 5.8 as generative.

5.3 Experimental results

We extensively experimented the proposed method on the task of action verification, k-shot learning and action recognition on a large composite of benchmarks. In the following subsections, we will first show the quantitate and qualitative results of action verification on ASLAN dataset, then report the performance of k-shot learning on UCF YouTube and HMDB51 datasets, and conclude with the action recognition on the composite dataset.

5.3.1 Action verification

The Action Similarity Labeling (ASLAN) dataset [33] is a recent action verification benchmark which includes thousands of video clips collected from YouTube, and over 400 complex action classes. The challenge is to identify the "same/not-same" video pairs, which change the

action recognition problem from a multi-class task into a binary one. The goal is to answer the question of whether a pair of video clips presents the same action or not. We use View1 to select the best parameters and test on View2 using the same evaluation criteria of [33]. The performance is reported based on average performance of ten separate experiments in a leave-one-out cross validation fashion. Each of the ten folds contains 300 pairs of same action videos and 300 not-same pairs. All the videos are first resized to 240×360. The cuboid size is $16 \times 16 \times 10$ pixels. The number of features is 300 for both video inputs and the number of metrics (H) is set as 40.

Table 5.1: The average accuracy of different single features with only pre-defined metric $\sqrt{\sum(a \cdot b)}$ on ASLAN dataset. HOG, HOF, HNF, MIP are the best performance on ASLAN reported by [33, 32]. MBH and ISA which have been demonstrated as the state-of-the-art features on several action benchmarks. The last second row is the generative learned feature (using equation 5.8) and the last row is the hybrid features learned by equation 5.12. The performance is reported as accuracy with standard error and Area Under the Curve (AUC). One can see that the learned features(last two row) perform almost equal with other features.

	Accuracy+std err	AUC
HOG [32]	$58.55 \pm 0.8\ \%$	61.59
HOF [32]	$56.82 \pm 0.6\ \%$	58.56
HNF [32]	$58.67 \pm 0.9\ \%$	62.16
MIP [32]	$62.23 \pm 0.8\ \%$	67.5
MBH [85]	$59.85 \pm 0.8\ \%$	61.5
ISA [40]	$59.11 \pm 0.7\ \%$	60.3
Generative feature	$61.49 \pm 0.7\ \%$	65.5
Hybrid feature	$62.05 \pm 0.9\ \%$	**67.1**

We first quantitatively compare our proposed method with multiple algorithms, including HOG, HOF, HNF, MIP which gives best performance on ASLAN reported by [33, 32], MBH and

ISA which has been demonstrated as the state-of-the-art features on several action benchmarks. Besides, we also test the performance of the generative model (learned using equation 5.8). The performance is tested in terms of "feature only" and "feature+metrics". The accuracy with standard error and AUC is shown in table 5.1 for "feature only" comparison. The aim of this experiment is to first test the features without the effect or aid of the learned metrics. In this experiments, we use the hand designed features (HOG, HOF, MIP, HNF, MBH) and the learned features (ISA, Generative, Hybrid) followed by the pre-defined metric $\sqrt{\sum(a \cdot b)}$ as suggested by [33]. From table 5.1 we can see that the learned features either learned by a generative or hybrid objective perform almost equally well with other state-of-the-art features.

Table 5.2: The average accuracy of different models on ASLAN dataset. Each model is composed of features and metrics. All the models with CSML design the features and metrics separately. In contrast, the generative model and hybrid model learn the features and metrics simultaneously. Compare with table 5.1, one can see that the performance can be improved using metric learning. Learning the metrics and features jointly is better than learning them separately as our proposed model is better than MIP+CSML by 3% on average accuracy. Moreover, learning them discriminatively and generatively, is better than pure generative method as the hybrid model also incorporates the label information for classification tasks.

	Accuracy+std err	AUC
HOG+CSML [32]	$60.15 \pm 0.6\ \%$	64.2
HOF+CSML [32]	$58.62 \pm 1.0\ \%$	61.8
HNF+CSML [32]	$57.2 \pm 0.8\ \%$	60.5
MIP +CSML [32]	$64.62 \pm 0.8\ \%$	70.4
MBH+CSML [85, 58]	$61.67 \pm 0.9\ \%$	63.26
ISA+CSML [40, 58]	$60.97 \pm 0.9\ \%$	62.64
Generative model	$65.71 \pm 0.7\ \%$	70.16
Hybrid model	$67.55 \pm 0.8\ \%$	**71.83**

We further compare the proposed model in terms of "feature+matrics" as shown in table 5.2. For the methods which only focus on designing the features, we use CSML [58] to learn the metrics on top of those feature representations. The CSML has been demonstrated to have the best results on MIP as reported in [32]. We also compare our proposed hybrid model with the generative model (learned by equation 5.8). Since the generative model only gives the hidden units response, we train a linear SVM on top of it. Compare table 5.2 with table 5.1, we see that using metric learning method such as CSML improves the overall performance of low-level features on ASLAN dataset in general. Further, by combining the co-trained metrics and the hybrid features, our hybrid model outperforms the MIP+CSML method by 3% on average accuracy and 1% on AUC. This demonstrate that to learn the features and metrics jointly can boost the overall performance given that the hybrid features perform equally with MIP in terms of 'feature only' as shown in table 5.1. Moreover, the hybrid model is better than the pure generative model by $1-2\%$ as shown in the last two rows in table 5.2. This means that learning the features and metrics in a discriminative and generative manner is more suitable for classification tasks.

To qualitatively evaluate the hybrid model, we further visualize the learned features U, V trained on the ASLAN dataset. We randomly sampled 18 pairs of features from the total of 300 as shown in figure 5.3. Each row of figure 5.3(a) is a filter (feature) randomly sampled from U, and similarly, filters from V is shown in 5.3(b). The filter pair (filters in the same row from (a)(b)) correspond with each other as the hybrid model learns the correlation between the feature responses on videos x and y. From each pair of filters, we can see that each pair captures different kind of motion transformations, such as translation and rotation.

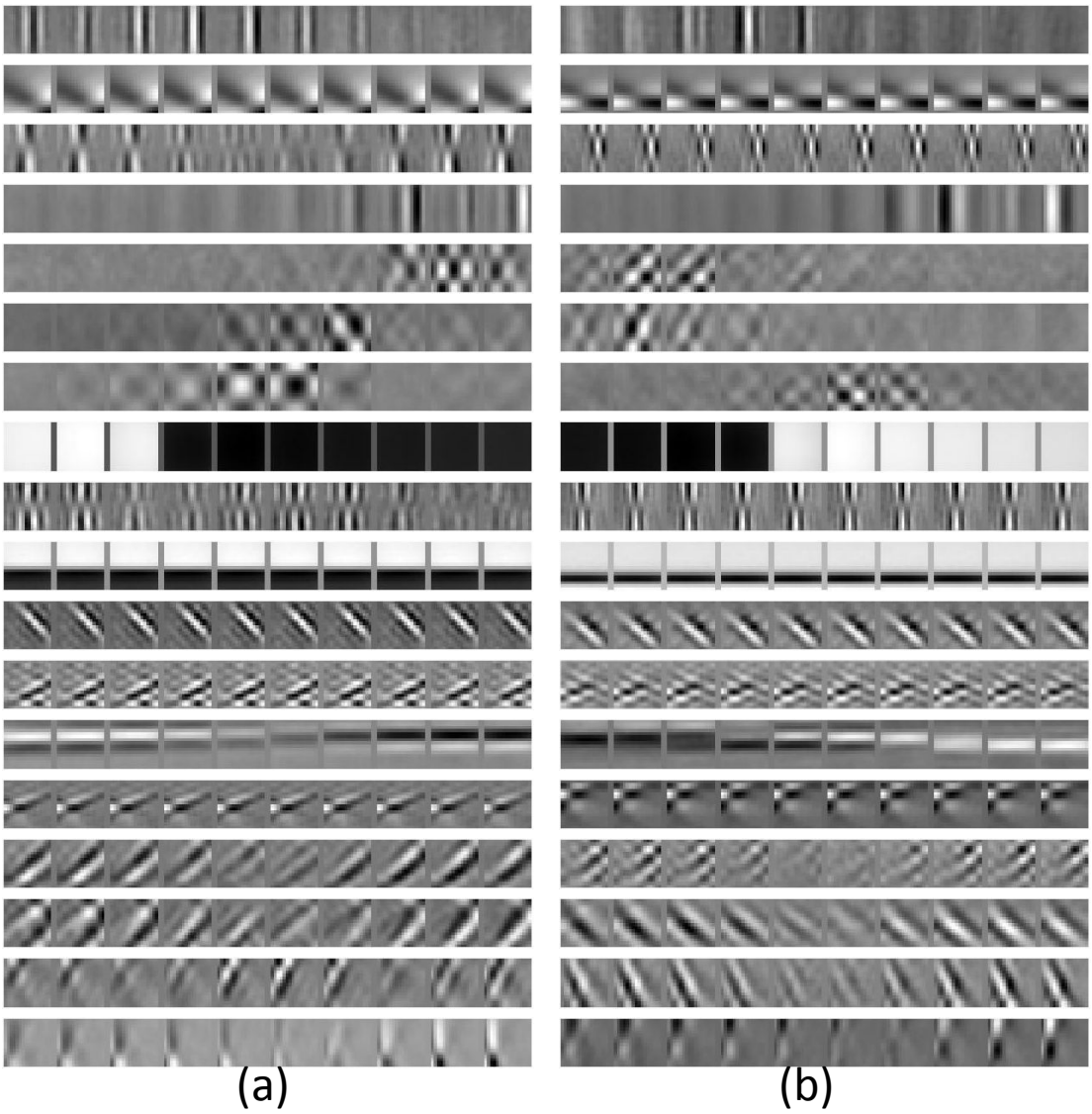


Figure 5.3: 18 feature pairs random sampled from 300 learned from the proposed hybrid model. Each row in (a) or (b) is a filter with size $16 \times 16 \times 10$. The filters in the same row from (a)(b) are a filter pair, which captures different motion transformations, such as translation and rotation.

Figure 5.4: Feature maps of important feature pairs. Each row is a video pair with the same action. The blue ones are the generatively learned feature maps. The red ones are the hybrid learned feature maps. The first column is the feature map of video x by the generatively learned features from U, the second column is the feature maps of video y with generatively learned features form V. The third column is the feature map of video x with the discriminatively features from U, the fourth column is the feature map of y with hybrid features from V. The discriminatively learned features capture more motion information compared with the hybrid ones which also captures lots of static edges information.

Since the generatively learned feature pairs also capture different motion transformations, it is very hard to tell which method is better directly from seeing the filters. To compare them qualitatively, we further conduct an experiment which, given a pair of videos with same action content, finds several most important feature pairs based on the high responses of units E. The feature pairs are further used to calculate the feature maps of the given videos and we select the feature maps with a high threshold for better visualization. Figure 5.4 shows the feature maps for the given video pairs. Each row is for one video pair with the same action. The left two columns (blue masks) are the feature maps of a pair of generative learned filters. Similarly, the right two columns (red masks) are the hybrid ones. From the feature maps, we find that the hybrid learned features are more focused on the motion similarity compared to the generative ones which capture the static horizontal or vertical edges as well.

5.3.2 Performance on k-shot learning

We further test our method on the task of K-shot ($K >= 2$) learning on YouTube action dataset [48] and HMDB51 dataset [36]. K is choosed to be equal to 2, 10, 20, 30, 40 respectively. For each run, we pick K videos from each category, and learn the features and metrics using pair of them. The rest of the videos in the dataset in each run will be the testing examples. Once the model is learned, given a test video, we perform the similarity measurement using the learned model and find the majority votes of the test video. We compare our method with the state-of-the-art descriptor MBH, ISA using KNN (nearest neighbor) or linear SVM. The experimental results are summarized in table 5.3 and 5.4. We can see that the hybrid model outperforms the state-of-the-art descriptors in the K-shot tasks as it focuses on the similarity measurement between two videos.

Table 5.3: The K-shot average accuracy on UCF YouTube dataset compared with HOG, MBH and ISA using KNN or linear SVM.

	K=2	K=10	K=20	K=30	K=40
HOG+KNN	8.4	16.1	23.1	31.5	37.4
HOG+SVM	8.9	17.2	23.3	29.4	36.7
MBH+KNN [85]	11.8	21.5	26.5	36.6	40.1
MBH+SVM [85]	11.2	20.6	28.9	35.4	39.4
ISA+KNN [40]	10.3	21.8	29.1	35.9	38.2
ISA+SVM [40]	11.5	21.8	29.5	34.1	37.8
Generative+KNN	12.2	22.3	30.9	35.5	40.5
Generative+SVM	12.6	21.5	30.2	37.7	41.4
Hybrid Model	**13.9**	**25.7**	**32.3**	**38.6**	**43.9**

Table 5.4: The K-shot average accuracy on HMDB51 dataset compared with HOG, MBH and ISA using KNN or linear SVM.

	K=2	K=10	K=20	K=30	K=40
HOG+KNN	1.9	4.1	7.0	10.4	15.3
HOG+SVM	1.9	4.0	6.6	10.2	16.6
MBH+KNN [85]	2.8	4.5	7.5	12.6	17.1
MBH+SVM [85]	2.2	4.2	7.9	11.4	16.4
ISA+KNN [40]	2.3	5.8	7.2	12.9	16.2
ISA+SVM [40]	3.5	5.3	7.5	12.7	16.8
Generative+KNN	3.2	6.2	8.7	13.5	17.2
Generative+SVM	3.6	6.5	8.9	13.6	18.8
Hybrid Model	**4.9**	**8.2**	**9.8**	**14.8**	**20.3**

5.3.3 Performance on composite dataset

To demonstrate the ability of our proposed method for dealing with large category dataset, we further build a composite dataset from four benchmark dataset: KTH, Weizmann, YouTube and HMDB51. The performance using ten-fold-cross-validation of different methods on this composite dataset is shown in table 5.5. We use linear SVM for the single features.

Table 5.5: The average accuracy on composite dataset.

	HOG	HOF	MBH	ISA	Hybrid
Composite	50.9	54.1	57.0	58.4	**63.3**

5.4 Summary

Measuring the similarity of human actions in videos is a challenging task. Two critical factors that affect the performance include low-level feature representations and similarity metrics. However, finding the right representations and metrics is hard. In this chapter, we describe a novel approach that learns both of them directly from the data. We propose a generative plus discriminative learning method based on gated auto encoders to simultaneously learn the features and their associated metrics. Our method differs from existing representation or metric learning methods in two ways: 1) while other methods treat feature learning and metric learning as independent tasks, we argue that they should be learned jointly since features and metrics are tightly inter-depended; 2) our method learns more discriminative features than its purely generative counterparts. Extensive experimental results on action verification, k-shot learning and large category recognition on ASLAN, HMDB51, YouTube benchmarks demonstrate significant performance improvements with our proposed method.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Summary of contributions

In this thesis, we have proposed learning hierarchical representations using deep learning methods for video analysis. Our models learn the good features and the higher level representations directly from the raw signal without the need of hand-crafted features. We have developed systems and algorithms for three challenging video analysis tasks: complex event recognition, object detection in videos and measuring action similarity.

In the automatic event detection task, we use TICA to learn the good features of three modalities (audio, scene and motion) from the large collection of unconstrained videos. Further, instead of manually collecting and labeling the concepts and training a discriminative concept detectors, we propose to discover the data-driven concepts from three modalities based on the learned features using deep belief nets (DBNs). So the final representation of each complex video is the similarity score of the clips in that video with the data-driven concepts. Finally, a compact and robust sparse representation is learned to jointly model the concepts from all three modalities.

In the object detection task, we proposed a novel approach to boost the performance of generic object detectors on videos by learning video-specific features using a deep neural network. The insight behind our proposed approach is that an object appearing in different frames of a video clip should share similar features, which can be learned to build better detectors. Unlike many supervised detector adaptation or detection-by-tracking methods, our method does not require any extra annotations or utilize temporal correspondence. We start with the high-confidence detections from a generic detector, then iteratively learn new video-specific features and refine the detection scores. In order to learn discriminative and compact features, we proposed a new feature learning method using a deep neural network based on auto encoders. It differs from the existing unsupervised feature learning methods in two ways: first it optimizes both discriminative and generative

properties of the features simultaneously, which gives our features better discriminative ability; second, our learned features are more compact, while the unsupervised feature learning methods usually learn a redundant set of over-complete features.

In the action similarity measuring task, we proposed to measure the similarity of human actions in videos by learning the features and metrics jointly and directly from the data. We proposed a generative plus discriminative learning method based on gated auto encoders to simultaneously learn the features and their associated metrics. Our method differs from existing representation or metric learning methods in two ways: 1) while other methods treat feature learning and metric learning as independent tasks, we argue that they should be learned jointly since features and metrics are tightly inter-depended; 2) our method learns more discriminative features than its purely generative counterparts. Extensive experimental results on action verification, k-shot learning and large category recognition on ASLAN, HMDB51, YouTube benchmarks demonstrate significant performance improvements with our proposed method.

6.2 Future work

The approaches proposed in this work can be improved in many ways. We describes some ideas in the following subsections.

6.2.1 Incremental feature learning

In many deep learning models, the optimal number of hidden units are usually predefined based on cross validation, because determining model complexity is an important problem in machine learning. As the digital data grows exponentially, it is usually challenging for online learning to perform from a massive stream of data. Instead of fixing the number of hidden units, an incremental feature learning algorithm is needed to determine the optimal model complexity for large-scale, online datasets. The algorithm can first learn a set of features from a small dataset,

then adding or deleting features based on the new data. It may composed of two processes: adding features and merging features. Adding new features can be achieved by minimizing the objective function. Through merging similar features, the model can obtain a compact feature set and prevent over-fitting to the current data. Also, when the data distribution changes, the model should be effective in recognizing new patterns and quickly adapt to it.

For the work on object detection in videos, one drawback of the proposed model is the huge computational time the model takes to learn from the new data. One way to further improve it will be to start from a pre-learned feature set, then to add or merge the features to adapt to the new videos. In this way, the computational time will be decreased and the learning will be more efficient.

6.2.2 Improving the pooling scheme

The current deep learning methods are very effective at capturing low-level image structure, such as the Gabor like filters. While the next challenge is to find representations appropriate for mid and high-level structures, i.e. corners, junctions, and object parts, which are surely important for understanding images and videos. From the low-level representation to higher level, in this thesis, we simply using the average or max pooling which loosely keep the geometric information but can not generate back to the original input. Recently, a novel inference scheme [97] that ensures each layer reconstructs the input, has been proposed based on sparse coding with switches providing direct paths to the input. To qualitatively visualize the learned high level representations, such as the high-level representations learned in the human detection task, it will be very useful to adopt the switch scheme and reconstruct the input to see whether it is meaningful.

6.2.3 Learning good features for tracking

In the human detection work, the features are learned only based on appearance information (the color patches). It will be more interesting to learn the features from motion information such

as optical flow and find the correlation of it with the appearance features. So then a good tracker can be built based on the learned features instead of brutally combining multiple hand-designed features into a huge feature vector. Moreover, the good features in terms of tracking can be also discovered using the same discriminative plus generative framework.

6.2.4 Learning action primitives

So far the current deep learning methods have shown the first level local spatial-temporal learned features capturing different kind of motion information across space and time. From a vision point of view, the learned features can now describe motion in the videos just like the function of optical flow. The high level representation might be the action primitive which would be the spatial-temporal connection of local motion. To capture the action primitives, an other method [95] used connectivity measurement to get motion instance and further merging the similar instances into a primitive. To capture it using the deep learning methods, one will need to remodel the pooling scheme and add the time information into the model using a recurrent neural network.

LIST OF REFERENCES

[1] Caviar. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[2] Trecvid 2011. http://www-nlpir.nist.gov/projects/tv2011/tv2011.html.

[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[4] B. Babenko, P. Dollár, and S. J. Belongie. Task specific local region matching. In *ICCV*, 2007.

[5] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR*, 2011.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005.

[8] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *ICDAR*, pages 440–445, 2011.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

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

[11] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):743–761, 2012.

[12] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[13] D. J. Felleman and D. C. V. Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cereb Cortex*, pages 1–47, 1991.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *PAMI*, 2010.

[15] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[16] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *International Journal of Computer Vision*, 73(1):41–59, 2007.

[17] E. Goddard, D. J. Mannion, J. S. McDonald, S. G. Solomon, and C. W. G. Clifford. Color responsiveness argues against a dorsal component of human v4. *J Vis*, 11(4), 2011.

[18] I. Goodfellow, Q. Le, A. Saxe, H. Lee, and A. Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems*, 2009.

[19] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV08, pages 234–247, 2008.

[20] A. Graves, A. rahman Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013.

[21] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariance sparse coding for audio classification. In *UAI*, pages 149–158, 2007.

[22] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

[23] G. E. Hinton. Deep belief nets. In *Encyclopedia of Machine Learning*, pages 267–269. 2010.

[24] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18:1527–1554, July 2006.

[25] G. E. Hinton, S. Osindero, and Y. whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 2006.

[26] A. Hyvarinen and P. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 2000.

[27] A. Hyvarinen, P. Hoyer, and M. Inki. Topographic ica as a model of v1 receptive fields. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 4, pages 83 –88 vol.4, 2000.

[28] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke. Application of pretrained deep neural networks to large vocabulary speech recognition. In *Proceedings of Interspeech 2012*, 2012.

[29] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR (1)*, pages 696–701, 2005.

[30] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[31] C. Kanan and G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2472–2479. IEEE, 2010.

[32] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[33] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.

[34] A. Krizhevsky and G. E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, 2011.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[36] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[37] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, pages 432–439, 2003.

[38] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[39] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. W. Koh, and A. Y. Ng. Tiled convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2010.

[40] 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 *CVPR*, 2011.

[41] 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 *CVPR*, pages 3361–3368, 2011.

[42] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, pages 97–104, 2004.

[43] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems 20*, pages 873–880. 2008.

[44] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, page 77, 2009.

[45] H. Lee, P. T. Pham, Y. Largman, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *NIPS*, pages 1096–1104, 2009.

[46] H. Lee, R. Raina, A. Teichman, and A. Y. Ng. Exponential family sparse coding with application to self-taught learning. In *IJCAI*, pages 1113–1119, 2009.

[47] A. Levin, P. A. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, pages 626–633, 2003.

[48] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". *CVPR*, 2009.

[49] J. Liu, Y. Yang, I. Saleemi, and M. Shah. Learning semantic features for action recognition via diffusion maps. *Computer Vision and Image Understanding*, 116(3):361–377, 2012.

[50] X. Liu and B. Huet. Automatic concept detector refinement for large-scale video semantic annotation. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 97 –100, sept. 2010.

[51] A. C. Loui, J. Luo, S.-F. Chang, D. Ellis, W. Jiang, L. S. Kennedy, K. Lee, and A. Yanagawa. Kodak's consumer video benchmark data set: concept definition and annotation. In *Multimedia Information Retrieval*, pages 245–254, 2007.

[52] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, November 2004.

[53] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

[54] R. Memisevic. Gradient-based learning of higher-order image features. In *ICCV*, 2011.

[55] R. Memisevic. On multi-view feature learning. In *ICML*, 2012.

[56] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order boltzmann machines. *Neural Computation*, 22(6), 2010.

[57] V. Nair and J. J. Clark. An unsupervised, online learning framework for moving object detection. In *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, CVPR'04, pages 317–325, Washington, DC, USA, 2004. IEEE Computer Society.

[58] H. V. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *ACCV*, 2010.

[59] B. A. Olshausen. Sparse coding of time-varying natural images. In *In proc. of the int. conf. on independent component analysis and blind source separation*, pages 603–608, 2000.

[60] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series, Englewood Cliffs, New Jersey, 1993.

[61] A. rahman Mohamed, G. E. Dahl, and G. E. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech & Language Processing*, 20(1):14–22, 2012.

[62] A. rahman Mohamed, T. N. Sainath, G. E. Dahl, B. Ramabhadran, G. E. Hinton, and M. A. Picheny. Deep belief networks using discriminative features for phone recognition. In *ICASSP*, 2011.

[63] M. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2551 –2558, june 2010.

[64] M. Ranzato, C. S. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *NIPS*, pages 1137–1144, 2006.

[65] M. Ranzato and M. Szummer. Semi-supervised learning of compact document representations with deep networks. In *ICML*, pages 792–799, 2008.

[66] M. A. Ranzato, J. Susskind, and G. Hinton. On deep generative models with applications to recognition. *Neural Computation*, 56(2):2857–2864, 2011.

[67] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*. IEEE Computer Society, 2008.

[68] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *WACV/MOTION*, pages 29–36, 2005.

[69] P. M. Roth, H. Grabner, D. Sko?aj, and H. Bischof. On-line conservative learning for person detection. In *In Proc. VS-PETS*, 2005.

[70] P. M. Roth, S. Sternig, H. Grabner, and H. Bischof. Classifier grids for robust adaptive object detection. In *CVPR*, pages 2727–2734, 2009.

[71] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. *Journal of Machine Learning Research - Proceedings Track*, 5, 2009.

[72] G. Schindler, L. Zitnick, and M. Brown. Internet video category recognition. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –7, june 2008.

[73] B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. In *ICANN*, 1997.

[74] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *IJCNN*, pages 2809–2813, 2011.

[75] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005.

[76] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, pages 1815–1821, 2012.

[77] C. G. M. Snoek. Early versus late fusion in semantic video analysis. In *In ACM Multimedia*, pages 399–402, 2005.

[78] J. Susskind, G. E. Hinton, R. Memisevic, and M. Pollefeys. Modeling the joint density of two images under a variety of transformations. In *CVPR*, 2011.

[79] G. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. *Computer VisionECCV 2010*, pages 140–153, 2010.

[80] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[81] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, 2010.

[82] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *IEEE International Conference on Computer Vision*, 2009.

[83] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008.

[84] P. Viola and M. Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.

[85] H. Wang, A. Kl, C. Schmid, and C.-l. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[86] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, page 127, sep 2009.

[87] M. Wang, W. Li, and X. Wang. Transferring a generic pedestrian detector towards specific scenes. In *CVPR*, pages 3274–3281, 2012.

[88] M. Wang and X. Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *CVPR*, pages 3401–3408, 2011.

[89] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, pages 32–39, 2009.

[90] X. Wang, G. Hua, and T. X. Han. Detection by detections: Non-parametric detector adaptation for a video. In *CVPR*, pages 350–357, 2012.

[91] X.-Y. Wei, Y.-G. Jiang, and C.-W. Ngo. Concept-driven multi-modality fusion for video search. *Circuits and Systems for Video Technology, IEEE Transactions on*, 21(1):62 –73, jan. 2011.

[92] H. Wersing and E. Körner. Learning optimized features for hierarchical models of invariant object recognition. *Neural Computation*, 15(7):1559–1588, 2003.

[93] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, 2002.

[94] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1794–1801, 2009.

[95] Y. Yang, I. Saleemi, and M. Shah. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1635–1648, 2013.

[96] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. *Learning*, 22(x):1–9, 2009.

[97] M. Zeiler, G. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. *ICCV*, 2011.

[98] C. Zhang, R. Hamid, and Z. Zhang. Taylor expansion based classifier adaptation: Application to person detection. In *CVPR*, 2008.

[99] G. Zhou, K. Sohn, and H. Lee. Online incremental feature learning with denoising autoencoders. *Journal of Machine Learning Research - Proceedings Track*, 22:1453–1461, 2012.