Deep Learning of Invariant Spatio-Temporal Features from Video

Bo Chen
California Institute of Technology
Pasadena, CA, USA
bchen3@caltech.edu

Jo-Anne Ting
U. of British Columbia
Vancouver, Canada
jating@cs.ubc.ca

Benjamin Marlin
U. of British Columbia
Vancouver, Canada
bmarlin@cs.ubc.ca

Nando de Freitas
U. of British Columbia
Vancouver, Canada
nando@cs.ubc.ca

Abstract

We present a novel hierarchical, distributed model for unsupervised learning of invariant spatio-temporal features from video. Our approach builds on previous deep learning methods and uses the convolutional Restricted Boltzmann machine (CRBM) as a basic processing unit. Our model, called the *Space-Time Deep Belief Network* (ST-DBN), alternates the aggregation of spatial and temporal information so that higher layers capture longer range statistical dependencies in both space and time. Our experiments show that the ST-DBN has superior performance on discriminative and generative tasks including action recognition and video denoising when compared to convolutional deep belief networks (CDBNs) applied on a per-frame basis. Simultaneously, the ST-DBN has superior feature invariance properties compared to CDBNs and can integrate information from both space and time to fill in missing data in video.

1 Introduction

The problem of how to represent images and video sequences is the most fundamental problem in computer vision. Raw pixels values are well understood to be a highly redundant representation for natural images and video. Much of the progress in computer vision has been driven by the introduction of increasingly more complex hand-crafted features, leading to improved representations. We believe that this development path is not sustainable and that future improvements in image representations will, instead, be driven by efficient and robust algorithms that learn to extract hierarchical, distributed feature representations from images and video in a fully unsupervised manner.

There are several desirable properties of image representations including i) invariance under common transformations [1, 2, 3], ii) retention of discriminative information [4, 5], and iii) retention of generative information [6, 7, 8]. These properties are generally in tension with each other. However, if they can be obtained simultaneously, the result would be a single representation applicable to both high and low-level computer vision tasks such as object detection, classification, action recognition, in-painting, and de-noising.

In this paper, we propose a new model that we call the Space-Time Deep Belief Network (ST-DBN). It is a hierarchical, distributed probabilistic model for unsupervised learning of invariant spatio-temporal features from video. It builds upon recently proposed deep learning approaches [6, 9, 10]. The basic building block in the model is the convolutional Restricted Boltzmann machine (CRBM), which was developed for images and time series [7, 11, 12]. The ST-DBN model aggregates over

space and time using alternating layers of spatial and temporal CRBMs. Higher layers have the ability to capture longer range statistical dependencies in both space and time. As a result, even if a video changes rapidly from frame to frame, the high-level representation of the video sequence can still exhibit a large degree of invariance, assuming that the sequence has a predictable structure that is captured by the model.

We review related work in Sec. 2, describe the CRBM model in Sec. 3, and introduce the ST-DBN model in Sec. 4. We present experimental results in Sec. 5, where we evaluate the invariance properties, discriminative performance and de-noising ability of the ST-DBN model compared to convolutional deep belief networks (CDBNs) applied to individual frames. Our results on invariance show that the ST-DBN model has significantly better invariance properties than the CDBN. Our results on discrimination show that the ST-DBN model performs significantly better than the CDBN on the well-known KTH action recognition data set. In addition, the ST-DBN's performance approaches that of the best known hand-crafted feature extraction methods for this data set. Our results show that the ST-DBN model's ability to de-noise video sequences is again superior to the CDBN. Finally, we illustrate the ST-DBN model's ability to fill in missing portions of frames in a video sequence.

2 Related Work

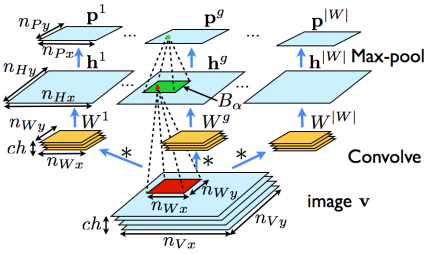
Related work on representing video falls into two main categories: hand-engineered feature extraction and models for unsupervised feature extraction. We begin with a discussion of work in unsupervised feature extraction and then briefly review work in hand-engineered feature extraction.

Models for Unsupervised Feature Extraction: Recent examples of models proposed in the machine learning community for spatio-temporal data include models based on restricted Boltzmann machines (RBMs) and their derivatives such as deep belief networks and deep Boltzmann machines, as well as models based on extensions of sparse coding. The work on RBM-related models is the closest to our work.

Taylor et al. [13] proposed a conditional RBM model where the hidden and visible units are conditioned on previous states of the visible units. More recently, Memisevic and Hinton have developed a factored RBM model to model image transformations [8]. Factored RBMs have also been adapted to model motion styles [14]. Together with the cuboids of [15] and sparse coding, Taylor et al. have used factored RBMs to extract features for human activity recognition [16]. This most recent work by Taylor et al. is closest to ours. Our approach differs in that we construct deep hierarchies using a single building block, i.e., the CRBM, while Taylor et al. use a mixture of architectures. Their work also focuses only on discrimination, while we demonstrate a broader range of properties including invariance and generative capabilities.

Deep architectures for spatio-temporal data based on sparse coding are also very interesting. A recent extension of sparse coding by Cadieu et al. [17] uses a factorized hierarchical model of videos where sparse representations of a sequence of images are separated into amplitude and phase. The amplitude is stable over time, and the phase dynamics are captured by a time-series model that linearizes the first derivative of the phase sequence. Preliminary results suggest that the model learns transformational invariants. The suitability of the model for classification remains to be studied, however.

Hand-Engineered Feature Extraction: Hand-engineered feature extraction has received considerable attention in the computer vision community. The processing pipeline for such systems usually involves several steps including pre-processing, feature detection, feature description, and classification based on feature descriptors. Spatio-temporal detectors, which select locally salient regions to reduce the amount of data, fall into two categories: i) detectors that treat time as an additional dimension to space and ii) detectors that decouple time and space. The first category extends 2D detectors to the space-time domain [18, 15, 19, 20]. The popular local cuboid detector of Dollár et al. [15] consists of a space-time interest point operator that acts as a temporal filter and extracts space-time blocks. Once salient blocks are located, a feature descriptor is applied to summarize their motion and shape. Most descriptors use a *bag-of-visual-words* model and discard the position of blocks. More sophisticated unsupervised learning methods have also been used to build descriptors given salient blocks, including the work of of Dean et al. [5]. Dean et al. used recursive sparse



Variable	X-Size	Y-Size	Channels
$\mathbf{v}$: Input Image	n_{Vx}	n_{Vy}	ch
W^g: Weight Matrix g	n_{Wx}	n_{Wy}	ch
$\mathbf{h}^g$: Hidden Group g	n_{Hx}	n_{Hy}	1
$\mathbf{p}^g$: Pooled Group g	n_{Px}	n_{Py}	1

Figure 1: The left figure shows the CRBM for a single image (video frame). The table on the right lists the key variables and their dimensions.

coding [5], applied to longer video blocks that were centered at locations returned by the cuboid detector of [15]. They clustered the descriptors and used a bag-of-visual-words for classification.

The main criticisms of the standard pipeline is that it is based on an *a priori* notion of saliency and ignores any structure specific to a given data set. The complexity of the features that can be extracted is also inherently limited by the common use of clustering and bag-of-visual-words representations that discard all position information.

3 Convolutional Restricted Boltzmann Machines

As Fig. 1 shows, the CRBM consists of three layers of units: visible units $\mathbf{v}$, binary hidden units $\mathbf{h}$, and binary max-pooling units $\mathbf{p}$. The visible units consists of one variable $\mathbf{v}_{c,i,j}$ for each color channel c and pixel location (i, j). In a standard RBM for two-dimensional image data, the visible units are connected to the hidden units via symmetric, undirected edges. Each hidden unit $\mathbf{h}^k$ has a corresponding weight matrix W^k with one element $W^k_{c,i,j}$ for each visible unit $\mathbf{v}_{c,i,j}$. The main difference between a standard RBM and a CRBM is that in the CRBM the hidden units are collected into groups $\mathbf{h}^g$, and each group defines a single weight matrix W^g of size much smaller than the input image. This weight matrix is applied to the input image convolutionally to determine the hidden unit activities. The result is a two-dimensional sheet of hidden units, which we call a group. Each group is the same size as the input layer as seen in Fig. 1[1].

The other main difference, relative to standard RBMs, is that an additional max-pooling step is used to reduce the spatial resolution of the hidden layer and to achieve greater invariance to spatial transformations of the input image. This is important when CRBMs are stacked into deep models. Given an integer-valued pooling ratio ρ, each sheet of hidden units $\mathbf{h}^g$ is partitioned into contiguous, non-overlapping $\rho \times \rho$ blocks $B_1, B_2, ...$ Max-pooling is applied to each block B_α of each hidden unit group $\mathbf{h}^g$ individually to determine the value of the corresponding unit $\mathbf{p}^g_\alpha$ in the pooled group $\mathbf{p}^g$. Importantly, this pooling step preserves the two-dimensional structure of hidden unit groups as seen in Fig. 1.

In addition to the weight matrix W^g for each group, the CRBM also has visible bias terms d_c (one per input channel) and hidden bias terms b_g (one per hidden unit group). Formally, the set of parameters in the CRBM is denoted by $\theta = \{W, b, d\}$. The energy function $E_\theta(\mathbf{v}, \mathbf{h}) \propto -\log P_\theta(\mathbf{v}, \mathbf{h})$ of the CRBM is defined in Eq. (1). Note that the symbol $*$ represents the two dimensional convolution operator. The constraint on the hidden and max-polled units simply come from the definition that the max-pooled unit p^g_α is on if and only if at least one of the hidden units $h^g_{i,j}$ is on in the corresponding pooling block B_α (see [7] for additional details).

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{g=1}^{|W|} \sum_{i,j=1}^{n_H} h^g_{i,j} (W^g_c * \mathbf{v}_c)_{ij} - \sum_{g=1}^{|W|} \sum_{i,j=1}^{n_H} b_g h^g_{i,j} - \sum_{c=1}^{ch} \sum_{i,j=1}^{n_V} d_c v_{c,i,j}$$

$$\text{subject to} \quad \sum_{i,j \in B_\alpha} h^g_{i,j} + (1 - p^g_\alpha) = 1, \forall g, \alpha = 1, .., |B| \tag{1}$$

[1] If the "valid" mode for convolution is used, the size will instead be $n_{Hx} = n_{Vx} - n_{Wx} + 1$ and $n_{Hy} = n_{Vy} - n_{Wy} + 1$

Training CRBMs using Monte Carlo methods requires sampling from both the conditional distribution of the hidden units given the visible units and the conditional distribution of the visible units given the hidden units. If we define the visible unit activations as $A^v_c = d_c + \sum_{g=1}^{|W|} W^g_c * \mathbf{h}^g$ and the hidden unit activations for group g by $A^g = b_g + \sum_{c=1}^{ch} W^g_c * \mathbf{v}_c$ (again using convolution) we can express the conditional probabilities as seen in Eqs. (2) and (3).

$$P(h^g_{i,j} = 1|\mathbf{v}) = \frac{\exp(A^g_{i,j})}{1 + \sum_{r,s \in B_\alpha} \exp(A^g_{r,s})}, \quad P(p^g_\alpha = 0|\mathbf{v}) = \frac{1}{1 + \sum_{r,s \in B_\alpha} \exp(A^g_{r,s})} \tag{2}$$

$$P(v_{c,i,j} = 1|\mathbf{h}) = \frac{1}{1 + \exp(-A^v_{c,i,j})} \tag{3}$$

CRBMs are highly overcomplete by construction [12, 7], so additional regularization is required during training. As in [21], we place a penalty term on the activations of the max-pooling units to encourage them to be close to a small constant value r. Given a dataset of K images $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(K)}\}$, the problem is to find the set of parameters θ that minimizes the objective:

$$-\sum_{k=1}^{K} \log \sum_{\mathbf{h}} P(\mathbf{v}^{(k)}, \mathbf{h}^{(k)}) + \lambda \sum_{g=1}^{|W|} \left[r - \left(\frac{1}{K|B|} \sum_{k=1}^{K} \sum_{\alpha=1}^{n_H} P(p^g_\alpha = 1|\mathbf{v}^{(k)}) \right) \right]^2 \tag{4}$$

where $|B|$ is the number of max-pooled units in $\mathbf{p}^g$, λ is a regularization constant, and r is a constant that controls the sparseness of activated max-pooled units. We use 1-step contrastive divergence [6] to get an approximate gradient of the log-likelihood term, coupled with stochastic gradient descent on the regularization term [7] to optimize Eq. (4).

A practical issue that arises during training is the effect of boundaries [12] on convolution. If the image has no zero-padded edges, then boundary visible units will have fewer connections to hidden units than interior visible units. The connectivity imbalance will cause filters to collapse into the corner regions in order to reconstruct the boundary pixels well. To alleviate this problem, we pad each input image with a border of zeros having the same width and height as the weight matrices.

4 Space-Time Deep Belief Network

The Space-Time Deep Belief Network takes a video as input and processes it such that subsequent layers in the hierarchy aggregate over progressively longer-range input patterns in space and time. Fig. 2(a) shows the first layer of the ST-DBN—a spatial pooling layer—which takes an input video of n_{Vt} frames $\{\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, ..., \mathbf{v}^{(n_{Vt})}\}$. At every time step t, each spatial CRBM takes an input frame $\mathbf{v}^{(t)}$ of size $(ch \times n_{Vx} \times n_{Vy})$ and outputs a stack $\mathbf{p}^{(t)}$ of size $(|W| \times n_{Px} \times n_{Py})$, where W is the set of weights (defined in Sec. 3) shared across all spatial CRBMs. All CRBMs in the same spatial pooling layer share the same parameter vector θ.

The second layer of the network is a temporal pooling layer, which takes the low-resolution image sequence $\{\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, .., \mathbf{p}^{(n_{Vt})}\}$ from the spatial pooling layer and outputs a shorter sequence $\{\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, ..., \mathbf{s}^{(n_{St})}\}$. Fig. 2(b) shows that each pixel at location (i, j) in the image frame is collected over time to form a temporal sequence $s_{I_{ij}}$ of size $(|W| \times n_{Vt} \times 1)$. Each $s_{I_{ij}}$ is fed into a CRBM, which convolves the temporal sequence $s_{I_{ij}}$ with temporal filter weights W'. Similar to the spatial CRBM in Sec. 3, the temporal CRBM uses a set of weights W' where the g-th temporal filter W'^g has length n_{Wt}. However, unlike the spatial CRBM, the temporal CRBM max-pools only over one dimension, i.e., time.

The temporal pooling layer has a total of $(n_{Px} \times n_{Py})$ CRBMs since every pixel in the image frame is collected over time and then processed by a CRBM. Given $s_{I_{ij}}$, a temporal sequence of the (i, j)-th pixel, the temporal CRBM outputs $s_{O_{ij}}$, a stack of shorter sequences. $s_{O_{ij}}$ has a size of $(|W'| \times n_{St} \times 1)$, where $n_{St} \leq n_{Vt}$. The final step of the temporal pooling layer re-arranges the temporal sequence of each pixel into the original two-dimensional spatial layout of a video frame. As seen in Fig. 2(b), the final output of the temporal pooling layer is a representation $\{\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, ..., \mathbf{s}^{(n_{St})}\}$ that is reduced in both spatial and temporal resolution. Similar to spatial CRBMs, all CRBMs in the temporal pooling layer share the same parameters (temporal filter weights and bias terms).

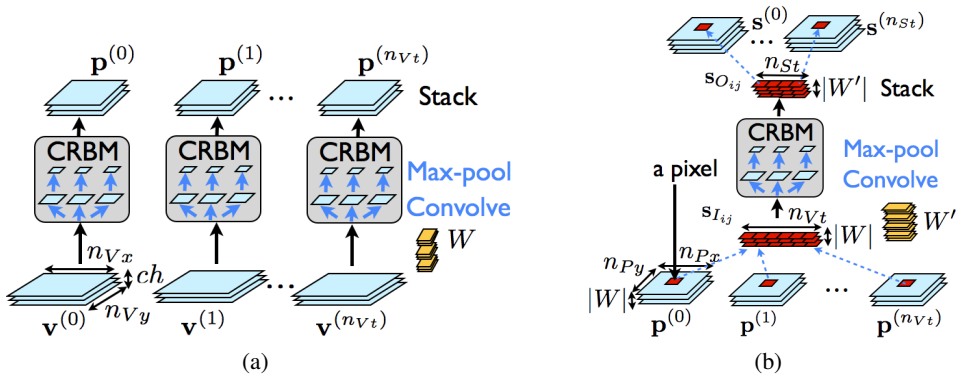


Figure 2: (a) Spatial pooling layer for an input video with n_{Vt} frames. Each input frame is fed into a CRBM. (b) Temporal pooling layer. Each pixel sequence is fed into a CRBM.

The sequence $\{\mathbf{s}^{(0)}, \mathbf{s}^{(1)}, ..., \mathbf{s}^{(n_{St})}\}$ is passed on to subsequent layers for further spatial and temporal pooling. The entire model is trained in a greedy layer-wise manner following [22], with every spatial/temporal-pooling layer trained as described in Sec. 3. We first initialize the hidden and max-pooling layers with the corresponding mean-field approximation to the filtering distribution, using Eq. (2). Following the procedure in [7], we then repeatedly perform block Gibbs sampling until convergence.

5 Experiments

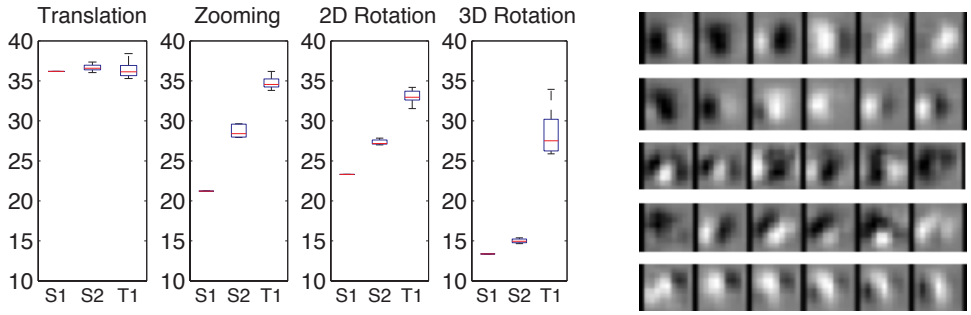
We evaluate three aspects of the ST-DBN. First, we quantify the degree of invariance of its hidden representations, using a previously proposed invariance score [2]. In Sec. 5.2, we use ST-DBN's features at various layers in order to assess their effectiveness in discriminative tasks. We report results on human action recognition using the KTH dataset [23]. Finally, in Sec 5.3, we demonstrate ST-DBN's ability to de-noise video and to infer missing data in video.

5.1 Measuring Invariance

Dataset & Training: We use natural videos from [2] to compare ST-DBNs and convolutional deep belief networks (CDBNs) [7], which consist of stacked layers of CRBMs. The 40 natural videos contain transformations of natural scenes, e.g., translations, planar rotations and 3D rotations, with modest variations in viewpoints between successive frames. We extract the relevant transformations by downsampling the videos of typical size $(640 \times 320 \times 200)$ into snippets of size $(110 \times 110 \times 50)$. For videos with rotating objects, we randomly select 50 consecutive frames in time and crop out a 220×220 window (that is then resized to 110×110) from the center of each frame. The resulting 40 snippets are standardized on a per frame basis. Finally, we split the snippets evenly into training and test sets, each containing 20 snippets.

We train a 2-layer ST-DBN (a temporal pooling layer stacked above a spatial pooling layer) and a 2-layer CDBN. The ST-DBN is similar to the CDBN in terms of the involved optimization steps and structural hyperparameters that need to be set. We use stochastic approximation with two-step averaging and mini-batches[2] to optimize parameters. The first layers of the ST-DBN and CDBN are the same, and we use $n_{Wx} = n_{Wy} = 10$ and $\rho = 3$ for layer 1. After cross-validation, we settled on the following hyperparameter values: 25 filters for the first layer and 64 filters for higher layers, a learning rate of 0.1, a sparsity level r of 0.01, and a regularization value λ of 1. The filter weights were initialized with white Gaussian noise, multiplied with a small scalar 0.1. For layer 2, we use $n_{Wx} = n_{Wy} = 10, \rho = 3$ for the CDBN and $n_{Wt} = 6$ with a pooling ratio of 3 for the ST-DBN. Within a reasonably broad range, we found results to be insensitive to these hyperparameter settings.

[2]We use an initial momentum of 0.5 for the first 2 epochs before switching to a value of 0.9. In order to fit the data in memory, we use small batch sizes—2 and 5 for spatial and temporal pooling layers, respectively—and train on subsampled spatio-temporal patches that are approximately 16 times larger than the filter. Despite subsampling, the patches are sufficiently large to allow for competition among hidden units during convolution.



(a) (b)

Figure 3: (a) Invariance scores for common transformations in natural videos, computed for layer 1 (S1) and layer 2 (S2) of a CDBN and layer 2 (T1) of ST-DBN (higher is better). (b) Learned layer 2 ST-DBN filters on KTH. Time goes from left to right for each row.

Invariance Measure: To evaluate invariance, we use the measure proposed by [2] for a single hidden unit i, which balances its global firing rate $G(i)$ with its local firing rate $L(i)$. The invariance measure for a hidden unit i is $S(i) = L(i)/G(i)$, with:

$$L(i) = \frac{1}{|Z|} \sum_{z \in Z} \frac{1}{|T(z)|} \sum_{x \in T(z)} f_i(x) \qquad G(i) = E[f_i(x)]$$

where $f_i(x)$ is an indicator function that is 1 if the neuron fires in response to input x and is 0 otherwise; Z is the set of inputs that activate the neuron i; and $T(z)$ is the set of stimuli that consists of the reference stimulus x with transformations applied to it. $L(i)$ measures the proportion of transformed inputs that the neuron fires in response to. $G(i)$ measures the neuron's selectivity to a specific type of stimuli. For each video and hidden unit i, we select a threshold such that i fires $G(i) = 1\%$ of the time. We then select 40 stimuli that activate i the most (these are single frames for the spatial pooling layers and short sequences in the temporal pooling layers) and extend the temporal length of each stimulus both forward and backward in time for 8 frames each. The local firing rate $L(i)$ is then i's average firing rate over 16 frames of stimuli, and the invariance score is $L(i)/0.01$. The invariance score of a network layer is the mean score over all the max-pooled units.

Fig. 3(a) shows invariance scores for translations, zooming, and 2D and 3D rotations using layer 1 of the CDBN (S1), layer 2 of the CDBN (S2), and layer 2 of ST-DBN (T1). S1 serves as a baseline measure since it is the first layer for both CDBN and ST-DBN. We see that ST-DBN **yields significantly more invariant representations** than CDBN (S2 vs. T1 scores). ST-DBN shows the greatest invariance for 3D rotations—the most complicated transformation. While a 2-layer architecture appears to achieve greater invariance for zooming and 2D rotations, ST-DBN has more pronounced improvement. For translation, all architectures have built-in invariance, leading to similar scores. Since ST-DBN is trained on video sequences, whereas the CDBN is trained on images only, a comparison to CDBN is unfair. Nonetheless, this experiment highlights the importance of training on temporal data in order to achieve invariance.

5.2 Unsupervised Feature Learning for Classification

Dataset & Training: We used the standard KTH dataset [23] to evaluate the effectiveness of the learned feature descriptors for human activity recognition. The dataset has 2391 videos, consisting of 6 types of actions (walking, jogging, running, boxing, hand waving and hand clapping), performed by 25 people in 4 different backgrounds. The dataset includes variations in subject, appearance, scale, illumination and action execution. First, we downsampled the videos by a factor of 2 to a spatial resolution of 80×60 pixels each, while preserving the video length (~ 4 sec long each, at 25 fps). Subsequently, we pre-processed the videos using 3D local contrast normalization.

We divided the dataset into training and test sets following the procedure in [24]. For a particular trial, videos of 9 random subjects were used for training a 4-layer ST-DBN, with videos of the remaining 16 subjects used for test. We used leave-one-out (LOO) cross-validation to calculate classification results for the 16 test subjects. For each of the 16 rounds of LOO, we used the remaining

Table 1: Average classification accuracy results for KTH actions dataset.

LOO protocol		Train/test protocol of [23]	
Method	Accuracy (%)	Method	Accuracy (%)
4-layer ST-DBN	90.3 ± 0.83	4-layer ST-DBN	85.2
3-layer ST-DBN	$\mathbf{91.13} \pm 0.85$	3-layer ST-DBN	**86.6**
2-layer ST-DBN	89.73 ± 0.18	2-layer ST-DBN	84.6
1-layer ST-DBN	85.97 ± 0.94	1-layer ST-DBN	81.4
Liu & Shah [20]	94.2	Laptev et al. [25]	91.8
Wang & Li [24]	87.8	Taylor et al. [16]	89.1

24 subjects to train a multi-class linear SVM classifier and tested on the one test subject. For a trial, the classification accuracy is averaged over all 6 actions and 16 test subjects.

There exists another train/test procedure, adopted in the original experiment setup [23], that does not use LOO. Videos from 9 subjects (subjects 2, 3, 5, 6, 7, 8, 9, 10 & 22) were chosen for the test set, and videos from the remaining 16 subjects were divided evenly into training and validation sets. We trained a 4-layer ST-DBN using videos from the training set. We used a nonlinear SVM with a Gaussian RBF kernel, where the parameters of the kernel were set using 5-fold cross-validation on the combined training and validation set (similar to [25]).

For both train/test protocols, we used the following settings to train the ST-DBN: $n_{Wx} = n_{Wy} = 8, \rho = 3$ for spatial pooling layers and $n_{Wt} = 6$ with a pooling ratio of 3 for temporal pooling layers. Fig. 3(b) shows a subset of ST-DBN temporal (layer 2) filters learned from the KTH data. The image filters in the first layer were similar to the ones widely reported in the deep learning literature. Each row of Fig. 3(b) shows a temporal filter over 6 time steps (note that temporal transformations are easily observed in a video).

Classification Performance: Table 1 shows classification results for both train/test protocols. For the LOO protocol, classification results for a ST-DBN with up to 4 layers are shown, averaged over 4 trials. We included comparisons to three other methods.We see that having additional (temporal and spatial) layers yields an improvement in performance, achieving a competitive accuracy of 91% (for the 3rd layer). Interestingly, a 4th layer leads to slightly worse classification accuracy—likely due to excessive temporal pooling on an already short video snippet (around 100 frames long). This demonstrates that precautions need to be taken to avoid too compact representations if the original videos are short.

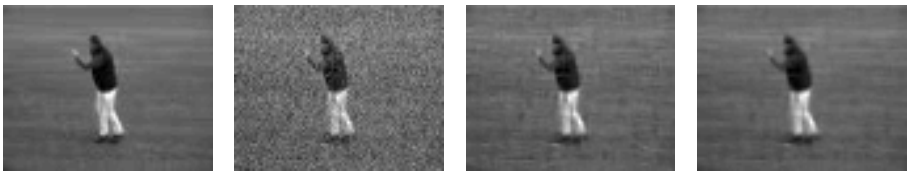
The best reported result to date, 94.2% by Liu & Shah [20], used a bag-of-words approach on cuboid feature detectors and maximal mutual information to cluster the video-words. In contrast to their work, we **do not use handcrafted feature detectors and take the entire video as input**, processing it in a completely automated way in one modeling framework.

We see similar patterns in performance of ST-DBN for the train/test protocol of [23] (values are for a single trial result, unlike LOO, which averages over 4 trials). The best result by Laptev et al. [25] uses a 3D extension of the Harris operator, and a combination of HoG and HoF (histograms of gradients and optic flow) descriptors. Taylor et al. [16] use dense sampling of space-time cubes, along with a factored RBM model and sparse coding to produce codewords that are then fed into an SVM classifier. Results from Table 1 suggest that ST-DBN has competitive discriminative abilities for action recognition, given its versatile single modeling framework. Unlike ST-DBN, all the other architectures in the table are unable to perform other tasks like in-painting and de-noising.

5.3 De-noising and Prediction

We demonstrate how a 2-layer ST-DBN can be used for video de-noising and to infer missing portions of frames in a video sequence. Fig. 4 shows de-noising results for a sample test frame in a video from the KTH dataset: the test frame is in Fig. 4(a) and the noisy test frame (corrupted with additive Gaussian noise[3]) in Fig. 4(b). We see that the 1-layer ST-DBN (Fig. 4(c)) de-noises the image frame well. The 2-layer ST-DBN (with an additional temporal pooling layer) gives slightly

[3]Each pixel is corrupted with additive mean-zero Gaussian noise with a standard deviation of s, where s is the standard deviation of all pixels in the entire (clean) video.



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

Figure 4: De-noising results: (a) Test frame; (b) Test frame corrupted with noise; (c) Reconstruction using 1-layer ST-DBN; (d) Reconstruction with 2-layer ST-DBN.

Figure 5: Top video shows an observed sequence of gazes/foci of attention (i.e., frames 2-4). Bottom video shows reconstructions within the gaze windows and predictions outside them.

better background de-noising. The normalized MSEs of 1-layer and 2-layer reconstructions are 0.1751 and 0.155, respectively. For reference, the normalized MSE between the clean and noisy video has value 1. Note that the de-noising effects are more visible over time (compared to single frame results shown below) and can be easily observed in video format.

Fig. 5 illustrates the capacity of the ST-DBN to reconstruct data and generate spatio-temporal predictions. The test video shows an observed sequence of gazes in frames 2-4, where the focus of attention is on portions of the frame. The bottom row of Fig. 5 shows the reconstructed data within the gaze window and predictions outside this window. The blurry effect in predicted parts of the frame is due to the loss of information incurred with max-pooling. Though max-pooling comes at a cost when inferring missing parts of frames, it is crucial for good discriminative performance. Future research must address this fundamental trade-off. The results in the figure represent an important step toward the design of attentional mechanisms for gaze planning. While gazing at the subject's head, the model is able to infer where the legs are. This coarse resolution gist may be used to guide the placement of high resolution detectors.

6 Conclusions

In this paper, we introduced the ST-DBN model, a hierarchical distributed probabilistic model for learning invariant features from spatio-temporal data. Using CRBMs as a building block, our model has the ability to capture long range statistical dependencies in both space and time. ST-DBN has superior feature invariance and discriminative performance relative to CDBNs applied to individual frames. The discriminative performance of the ST-DBN model also approaches that of state-of-the-art methods on KTH using fully unsupervised feature extraction in a single consistent model architecture. Simultaneously, ST-DBN shows promising generative performance in terms of video de-noising video and filling-in of missing data.

An interesting direction for future work is to consider alternatives to probabilistic max-pooling. While the max-pooling operation allows feature invariance to be captured hierarchically from spatio-temporal data, it has an adverse affect on the ability to synthesize full resolution output from the model. We plan to examine how the information loss associated with max-pooling can be minimized when performing inference. We conjecture that combinations of models with and without pooling will be required. Additionally, precautions should be taken to ensure representations are not made too compact with too many layers in the architecture. Model selection is an open challenge in this line of research.

References

[1] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.*, 14(4):715–770, 2002.

[2] I. Goodfellow, Q. Le, A. Saxe, and A.Y. Ng. Measuring invariances in deep networks. *NIPS*, 2009.

[3] K. Kavukcuoglu, M.A. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. *CVPR*, 2009.

[4] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. *ICML*, 2009.

[5] T. Dean, G. Corrado, and R. Washington. Recursive sparse, spatiotemporal coding. *IEEE International Symposium on Multimedia*, 2009.

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

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

[8] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Comput.*, 2010.

[9] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. *NIPS*, 2006.

[10] M.-A. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. Lecun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[11] G. Desjardins and Y. Bengio. Empirical evaluation of convolutional RBMs for vision. Technical report, University of Montreal, 2008.

[12] M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. *CVPR*, 2009.

[13] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. *NIPS*, 2007.

[14] G. W. Taylor and G. E. Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. *ICML*, 2009.

[15] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *VS-PETS*, 2005.

[16] G. W. Taylor and C. Bregler. Learning local spatio-temporal features for activity recognition. In *Snowbird Learning Workshop*, 2010.

[17] C. Cadieu and B. A. Olshausen. Learning transformational invariants from natural movies. *NIPS*, 2008.

[18] I. Laptev and T. Lindeberg. Space-time interest points. *ICCV*, 2003.

[19] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. *ICCV*, 2005.

[20] J. Liu and M. Shah. Learning human action via information maximization. *CVPR*, 2008.

[21] H. Lee, C. Ekanadham, and A. Ng. Sparse deep belief net model for visual area V2. *NIPS*, 2008.

[22] G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural Comp.*, 18(7):1527–1554, 2006.

[23] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. *ICPR*, 2004.

[24] Z. Wang and B. Li. Human activity encoding and recognition using low-level visual features. *IJCAI*, 2009.

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