Learning individual human activities from short binary shape sequences

Christian Wolf[a] Graham W. Taylor[b] Jean-Michel Jolion[a]

Technical Report LIRIS

[a]Université de Lyon, CNRS
INSA-Lyon, LIRIS, UMR5205, F-69621, France
[b]New York University
Computer Science Department, Courant Institute of Mathematical Sciences, USA

Abstract

We present a new machine learning-based algorithm capable of classifying individual human activities from very short sequences. Our method is based on a "deep" multi-stage architecture where each layer is learned independently of the other layers. Low-level shape features are extracted from short sequences of binary shapes and fed to a sequential probabilistic model (a conditional deep belief network), which learns the evolution of the low-level features through time through interactions with binary latent variables. No appearance model is needed. Actions are classified using an SVM trained on the posterior probabilities of the latent features extracted by the motion model. The method is capable of not only recognizing actions but also localizing them in space and time. We evaluated the algorithm on two different databases, the well known Weizmann dataset and our own, more challenging, dataset.

1 Introduction

Applications such as video surveillance, robotics, source selection, and video indexing often require the recognition of actions based on the motion of different actors in a video, for instance, people or vehicles. Certain applications may require assigning activities to several predefined classes, while others may rely on the detection of abnormal or infrequent activities. In this paper we deal with the former in a realistic surveillance setting. As opposed to a large part of the published state of the art, where decisions are often made globally on a whole video showing a single person, in our work multiple people are allowed to appear in a scene and classification decisions are taken per person and per "instant", i.e. per short sequence of several frames (typically around 7). This is achieved by extracting binary shape sequences for each moving object using background subtraction, and learning the evolution of shapes over time through a probabilistic model.

The amount of literature on action recognition has sky-rocketed in the last few years and it is not possible anymore to give an exhaustive account in this restricted space. We refer the interested reader to some very recently published surveys [1, 2, 3]. While early work on modeling human activities focused on articulated motion (e.g. [4]), most recent work on activity and event recognition does not explicitly model the human body. Instead, the current state of the art focuses on sparse local features like interest points and space-time interest points [5, 6, 7, 8, 9, 10, 11], or on motion segmentation through background subtraction [12, 13, 14], dense optical flow [15] or other holistic features [16, 17], with possible hybrid

methods [18, 19, 20] and classification through dense matching [21, 22] or graph matching [23].

Holistic methods generally compute features directly on the whole space-time (ST) cube of a video, or on ST-patches of it, e.g. moments [13], and volumetric box features on optical flow components [15]. Local methods combine features from several local primitives, an inherently structural data representation. In order to convert this representation to a numerically useful representation, most of them discard the structural part and resort to the bag-of-words (BoW) formalism [6] or its extensions. This tends to improve invariance but severely hurts discrimination power. Proposed extensions are, for instance, correlograms [24], local grouping and compound features [11], motion context [17], spatial co-occurrences of pairs of features [9, 25, 10], and parts-based models [26]. Fully taking into account spatial relationships through graph matching has recently been proposed [23], but this requires matching against several graph models per action class.

In contrast to object recognition problems, pure statistical and unstructured machine learning without feature extraction is difficult in this context due to several reasons: (i) the non-rigid nature of the relevant information; (ii) the mixture of relevant motion information and irrelevant texture information in the signal; and (iii) the extremely high dimensional space in which spatio-temporal data is embedded.

The question of whether and how to use learning is related to the choice of features, which in the context of video analysis is dominated by the question whether to extract features following segmentation (e.g. through BG-subtraction), or from local primitives like interest points. It has often been claimed that segmentation may fail in some cases and "segmentation-free" methods have become fashionable lately. On the other hand, methods based on local primitives like interest points or spatio-temporal interest points also suffer from severe drawbacks. Stable points are hard to extract in numbers large enough to provide high discriminative power on small and short patches. This requires the approach to classify long sequences, often whole test videos. This lack of a sufficient number of points has especially been reported for space-time interest points [5]. They also tend to lack efficiency in cases of slow or smooth motions of non textured objects since no (or only unstable) points are detected in these areas. Finally, and most importantly, the inherent nature of a set of features extracted on a set of points makes the features inherently structural, requiring aggregations into vectors or histograms (like a BoW representation) or motion context [17, 27] in order to apply the majority of statistical learning methods. Much of the useful discriminative information (including geometry and temporal relationships) can be lost during aggregation.

For this reason, apart from the BoW methods mentioned above, machine learning of human actions has been dominated by methods learning the temporal evolution of features like HMMs, Semi-Markov models and dynamic belief networks [28, 27, 29, 30, 31, 32, 33, 28, 29, 34]. Typically, a vectorial description is created frame by frame and its temporal evolution is modeled and learned. HMMs and dynamic belief networks share with our work the property that dynamical processes are modeled through hidden states. However, mixture models such as HMMs typically generate each observation from a single category or prototype. Distributed (also known as *componential*) state models generate each observation from a set of features that each contain some aspect of its static or dynamic nature. Our motion model uses a deep, componential representation capable of efficiently capturing complex interactions. In [35], a chain graph model for action recognition requires a priori knowledge of the nature (e.g. causality or correlation) and semantic meaning of relationships between different variables. In contrast, we assume that no such knowledge of the visible or latent variables exists - it is learned directly from the data.

Other learning-based methods include biologically-inspired ones [36], convolutional deep learning [37], methods based on boosting low-level features [38], topic models [7], trajectory matching [39], statistics calculated on the results of tracking [40], learning of spatio-temporal predicates and grammars [10, 41, 42] and other not yet mentioned probabilistic graphical

models [43].

In this work we advocate for the advantages of techniques based on segmentation, which for the moment are focused on applications where the camera is fixed. However, we argue that this situation is currently changing, as segmentation is becoming increasingly less difficult for situations with moving cameras: BG subtraction algorithms have also been proposed for PTZ cameras [44] and, more importantly, the recent success of depth cameras — like MS Kinect[1] or time of flight cameras — delivering RGB as well as depth information, makes it easier to obtain stable BG subtraction algorithms in generic situations.

It has already been argued that actions can be recognized from short sequences ("*snippets*") instead of lengthy videos. In [45], shape and motion features are calculated separately and combined before being fed into an SVM classifier. To our advantage, our method does not require the computation of dense optical flow, just background subtraction. The importance of combining shape and motion has been reported several times [19, 45, 46]. However, in most established methods where shape and motion are both used, they are treated separately. One of the strongest links between them has been achieved in [46], where shape and motion are combined into 2D motion history and motion energy images. Still, the evolution of shapes across time is incomplete due to spatio-temporal occlusions and difficult to exploit directly from these images. In [34], the evolution of silhouettes over time is modeled through HMM-like models over representations in shape space. The work is close to ours, though, from a learning perspective, differs in its use of a non-componential discrete hidden state.

In contrast, our approach directly models the evolution of shapes over time through successive stages: i) a background subtraction algorithm with post-processing produces a sequence of binary windows for each moving object; ii) low-level shape features are extracted from each binary window iii) high-level shape features are extracted efficiently with a single pass through a previously learned probabilistic model which captures dynamic interactions through latent variables; and iv) the top-level features are input to an SVM to obtain the final decision.

The paper is organized as follows: section 2 first briefly outlines how we extract binary shapes using background-subtraction. Our primary contributions are described in sections 3 and 4 which cover, respectively, the low-level shape features and the generative motion model, as well as its use for classification. Section 5 describes the experiments we performed on several different datasets and section 6 finally concludes.

2 Extracting shape sequences with background subtraction

In many situations, including those encountered in standard databases, background subtraction (the initial segmentation step of our algorithm) is not difficult to perform. This is evidenced by the fact that simple frame differencing often produces results of good quality. We successfully applied frame differencing with some postprocessing in all experiments described in this paper (see section 5). However, frame differencing can fail in some difficult conditions, for example, complex outdoor scenes. Algorithms based on Gaussian mixture models (GMM) have become the de facto standard for background subtraction in these conditions. They create an explicit background model, which allows the detection of objects at a standstill, and they are capable of tracking multiple background distributions. This permits them to handle more complex backgrounds, for example, the leaves on a tree or a flag blowing in the wind. Perhaps the most widely used member of this class is the Stauffer-Grimson algorithm [47] which also constitutes the basis of a method which we successfully applied as preprocessing step for our algorithm.

To both the frame differencing and GMM algorithms, we added a filter designed to fill very large holes which may appear with non textured and uniformly colored moving objects. Standard morphological operations like closing tend to connect neighboring moving people

[1] `http://www.xbox.com/en-US/kinect`

into a single block. Our filter sets a pixel to foreground if every single one of 8 walks in 8 different directions from this pixel encounters another foreground pixel. This approximates a topological procedure which fills holes but does not create bridges between neighboring connected components. The length of the walk is roughly set to the estimated width of a human body in the video.

Binary shapes are extracted and combined into sequences by thresholding sizes of connected components and associating them across frames according to their overlap. In particular, given a connected component (CC) at frame t, one or several CCs at frame $t+1$ are associated:

- if a single CC at $t+1$ overlaps the one of at t, it is associated.
- if several CCs overlap the one at t, we check for a possible split from t to $t+1$: all CCs at $t+1$ which sufficiently overlap the CC at t are associated.

No explicit model based tracking is necessary and occlusions are not handled at this moment.

3 The shape/motion model

We argue that actions can be classified accurately by modeling the evolution of associated binary shapes over time. While we could directly classify the binary image maps extracted by our background subtraction algorithm (described in section 2), we aim to provide the classifier with a more statistically salient representation of the input. There is much evidence that learning several layers of feature detectors can improve performance in vision tasks [48, 49]. This has typically been validated in the domain of static object recognition, but the arguments for building so called "deep architectures" also extend to sequences [38, 36, 50].

Learning more than one layer of latent variables can motivated in many ways. Foremost, learning multiple layers of feature extractors is representationally sound. There exist families of functions that can be represented much more efficiently with deep networks (those with multiple layers of latent variables) than shallow networks (one hidden layer) [51]. The fact that deep networks produce hierarchical representations is attractive, because humans organize their ideas hierarchically. This is not only intuitive, but permits non-local generalization. What this means is that prototype-based methods like mixture models and clustering require exponentially more parameters than distributed methods that employ features. Since features at each layer are derived from features from the layer below, features can become increasingly abstract (c.f. [52] for empirical evidence). In the case of temporal data, this also translates to higher layers capturing increasingly longer-term temporal dependencies. Finally, these layers of feature extractors can be trained unsupervised, that is, without labeled data that is often expensive to obtain.

We thus propose to extract multiple layers of features from each frame of the binary image maps before attempting classification. This is done by recursively applying a generative model that jointly models a "visible" representation of the input at time step, $\boldsymbol{v}_t$, and a latent, or "hidden" representation of the input, $\boldsymbol{h}_t$. While it may seem natural to simply choose $\boldsymbol{v}_t$ to be the pixels of each frame of the binary image maps, we prefer to model sequences of compact and robust shape descriptors. We first describe our input representation before focusing on how to extract features from the shape descriptors.

3.1 Low level features

Complex Zernike moments have been proven to capture shapes robustly and are widely used as features for object recognition [53, 54]. They are constructed using a set of complex polynomials which form a complete orthogonal basis set defined on the unit disc. Figure 1 illustrates the representational power of the Zernike decomposition on a binary example shape extracted

Figure 1: From left to right: an input image from one our datasets, then its reconstruction with different Zernike moments of different order (8, 12, 16, 20, 24, 28, 32).

from one of our test video databases. Depending on the complexity of the shape, quite accurate reconstructions can be obtained from the moments up to order 20-25, which correspond to 121 to 182 complex coefficients. We should mention that a reconstruction of good quality is not necessary to classify a shape or an action.

In addition to the shape descriptors of the binary object, additional features have been chosen to boost discrimination performance. The visible observations of the model are therefore comprised of a sequence of real valued vectors $\boldsymbol{v}_t$, where each vector $\boldsymbol{v}_t$ corresponds to a time instant t and holds the real part as well as the imaginary part of the complex Zernike moments of a single binary subwindow. To this representation we add three additional features: (i) the ratio between the height and the width of the bounding box, which is necessary since the Zernike moments are calculated on a normalized unit disk; (ii) two values for the gradient of the center of mass of the shape between instants $t-1$ and t. The manually selected additional features are necessary since the automatically selected features are calculated from a normalized bounding box with constant width and height and without any positional information. Some classes, like walking and running, are characterized by similar relative frame to frame movement. The main difference in the actions is the speed of the person's displacement. The additional features boost the discrimination performance between these classes.

3.2 Conditional restricted Boltzmann machines

A Restricted Boltzmann Machine (RBM) [55] is a bipartite Markov Random Field consisting of a layer of stochastic "visible" variables, $\boldsymbol{v} = \{v_i\}$, connected to a layer of stochastic latent variables, $\boldsymbol{h} = \{h_j\}$ (see figure 2a). The lack of direct connections among the latent variables ensures that they are conditionally independent given a setting of the visible variables, which simplifies inference and learning. RBMs typically use binary visible and latent variables, but for real-valued data (e.g. Zernike moments) we can use a modified RBM with Gaussian, real-valued variables and binary latent variables [56].

The RBM can be extended to dynamical data in the form of sequences $\boldsymbol{v}=\{\boldsymbol{v}_t\}$ of sets $\boldsymbol{v}_t$ of visible variables $\boldsymbol{v}_t=\{v_{it}\}$ and similarly indexed latent variables. This allows the model to capture temporal dependencies by making its latent and visible variables receive additional input from previous states of the visible variables. Originally introduced for generative modeling of motion capture data [50], this so-called Conditional RBM (CRBM) is illustrated in figure 2b. The number, N, of previous observations to which each layer connects is referred to as the order of the model. Conditioning on past data does not change its most important computational properties: simple, exact inference and efficient approximate learning.

For the case of real-valued input data, the CRBM defines a joint probability distribution over a real-valued observation, $\boldsymbol{v}_t$, and a collection of binary latent variables, $\boldsymbol{h}_t, h_{jt} \in \{0, 1\}$:

$$p(\boldsymbol{v}_t, \boldsymbol{h}_t|\boldsymbol{v}_{<t}) = \exp\left(-E(\boldsymbol{v}_t, \boldsymbol{h}_t|\boldsymbol{v}_{<t})\right)/Z(\boldsymbol{v}_{<t}). \quad (1)$$

The distribution is conditional on the history of past N observations, $\boldsymbol{v}_{<t}$, where $\boldsymbol{v}_{<t} = \boldsymbol{v}_{t-N}, \ldots, \boldsymbol{v}_{t-1}$ and it is normalized by constant Z which is intractable to compute exactly[2].

[2]To compute Z exactly we would need to integrate over the joint space of all possible inputs and all settings of the binary latent variables.

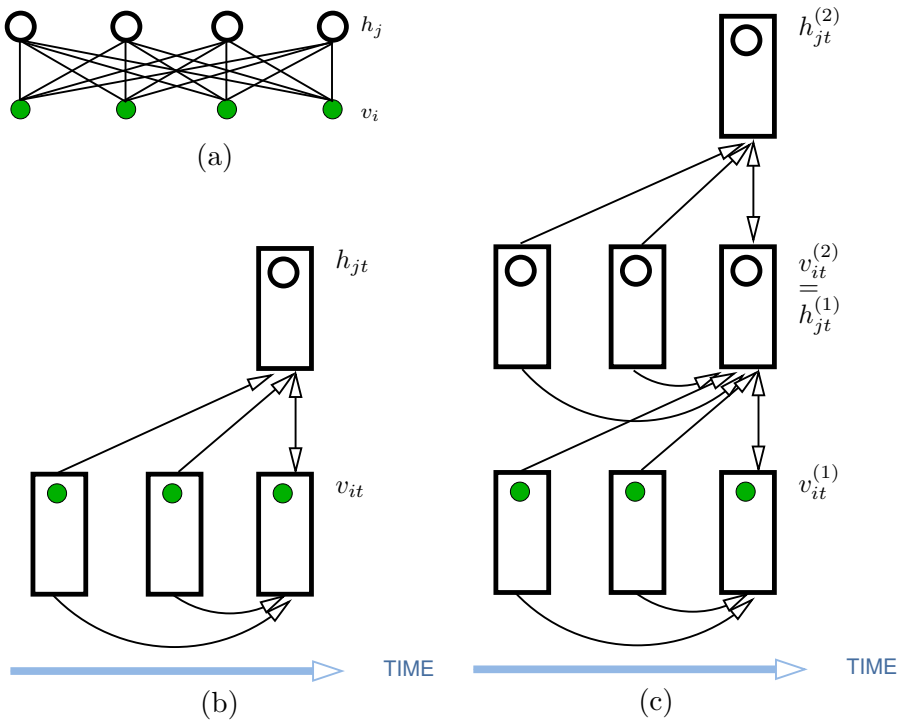

Figure 2: (a) a restricted Boltzmann machine (RBM); (b) a Conditional RBM for dynamic data. A single rectangle corresponds to a set of observed or hidden variables of an RBM; (c) a conditional deep belief network, i.e. a CRBM in multiple layers.

The joint distribution is characterized by an energy function:

$$E(\boldsymbol{v}_t, \boldsymbol{h}_t | \boldsymbol{v}_{<t}) = \sum_i \frac{1}{2}(v_{it} - \hat{c}_{it})^2 - \sum_j h_{jt}\hat{d}_{jt} - \sum_{ij} W_{ij} v_{it} h_{jt} \tag{2}$$

which captures the pairwise interactions between variables, assigning high scores to improbable configurations and low scores to probable configurations. Each visible variable contributes a quadratic offset to E (first term) that dominates Eq. 2 when it deviates too far from a "dynamical mean" that is a linear function of the previous observations: $\hat{c}_{it} = c_i + \sum_l A_{il} \boldsymbol{v}_{l,<t}$. The dynamical mean is much like a prediction from an autoregressive model of order N with constant offsets c_i.

Each latent variable contributes a linear offset to E (second term) which is also a function of the past N observations: $\hat{d}_{jt} = d_j + \sum_l B_{jl} \mathbf{v}_{l,<t}$, where d_j are, again, constant offsets. The third term is a bilinear constraint on the interaction between (current) visible and latent variables, characterized by weights W. A large value of W_{ij} means that v_i and h_j are strongly correlated. While other energy functions could be considered, Eq. 2 leads to analytically convenient conditional distributions, which are required for inference and learning.

Learning a CRBM

Ideally we would like to maximize the marginal conditional likelihood, $p(\boldsymbol{v}_t | \boldsymbol{v}_{<t})$, over parameters $\theta = \{W, A, B, \mathbf{c}, \mathbf{d}\}$ but this is difficult for all but the smallest models due to the intractability of computing Z. Learning, however, still works well if we approximately follow the gradient of another function called the contrastive divergence (CD) [57].

For sake of brevity, we refer the reader to [50] for details of learning a CRBM by CD. In short, learning relies on two main operations: 1) sampling the latent variables, given a window of training data, $\{\boldsymbol{v}_t, \boldsymbol{v}_{<t}\}$:

$$p(h_{jt} = 1|\boldsymbol{v}_t, \boldsymbol{v}_{<t}) = \left(1 + \exp(-\sum_i W_{ij} v_{it} - \hat{d}_{jt})\right)^{-1}, \tag{3}$$

and 2) reconstructing[3] the data, given the latent variables:

$$v_{it} \sim \mathcal{N}\left(v_{it};, \sum_j W_{ij} h_{jt} + \hat{c}_{it}, 1\right). \tag{4}$$

where $\mathcal{N}$ is the normal probability density function. Both Eq. 3 and 4 follow from Eq. 1. Note that we always *condition* on the past: it is never updated.

Given a trained CRBM and a N-step history of observations, we can obtain a joint sample from $p(\boldsymbol{v}_t, \boldsymbol{h}_t|\boldsymbol{v}_{<t})$ by alternating Gibbs sampling, i.e. starting at some reasonable initialization of $\boldsymbol{v}_t$ (e.g. $\boldsymbol{v}_{t-1}$) then alternating between Eq. 3 and 4 for some fixed number of steps (typically between 30 and 100).

Learning multi-layer representations

CRBMs can form the building blocks of deep networks through a greedy, sequential process. Once we have trained a CRBM, we can add additional layers of latent variables in the same way as a Deep Belief net (DBN) [49]. The previous layer CRBM is kept, and the sequence of hidden state vectors, while driven by the data, is treated as a new kind of "fully observed" data. The next level CRBM has the same architecture as the first (though we could alter the number of its hidden units) and is trained in the exact same way. Upper levels of the network can then model higher-order structure. The resulting model is called a conditional DBN or CDBN (see figure 2c). While inference in the single-layer CRBM is exact, inference in the CDBN is only approximate because of the directed connections between hidden variables (we ignore these connections when performing bottom-up inference).

There are a few practical changes we need to make when training the second layer (and potentially, layers beyond that). The first layer CRBM has real-valued visible variables and binary latent variables. Since we are learning a representation on top of the first layer CRBM, and treating the activations of the latent variables while driven by the data as "observed", the second CRBM has binary visible variables and binary hidden variables. This changes the energy function of the second layer CRBM to:

$$E(\boldsymbol{v}_t, \boldsymbol{h}_t|\boldsymbol{v}_{<t}) = -\sum_i v_{it}\hat{c}_{it} - \sum_j h_{jt}\hat{d}_{jt} - \sum_{ij} W_{ij} v_{it} h_{jt} \tag{5}$$

where $\hat{c}_{it}$ and $\hat{d}_{jt}$ have remained the same. Inference (via Eq. 3) does not change (since latent variables remain binary) but the reconstruction distribution (Eq. 4) becomes:

$$p(v_{it} = 1|\boldsymbol{h}_t, \boldsymbol{v}_{<t}) = \left(1 + \exp(-\sum_j W_{ij} h_{jt} - \hat{c}_{it})\right)^{-1}. \tag{6}$$

For simplicity, we have omitted indicies indicating the particular layer of the network.

In the context of our method for activity recognition, training two layers of feature extractors amounts to 1) learning a "real to binary" CRBM from the low-level features described in section 3.1; 2) Collecting the sequence of latent features from the first layer CRBM while driven by the training data; and 3) training a "binary to binary" CRBM on the sequence of latent features extracted by the first layer CRBM.

[3] In practice, we sample the hidden state but set the updated visible state to the mean. This suppresses noise and learns slightly faster.

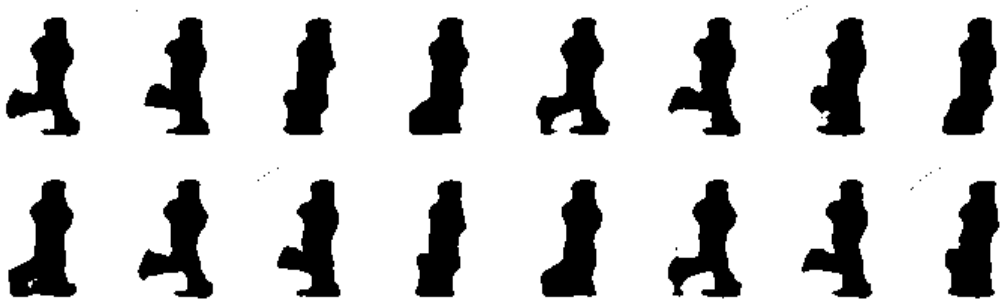

Figure 3: A sequence of consecutive artifical frames reconstructed from the Zernike moments sampled from a CDBN learned from training data of the running class.

It is standard practice to use the real-valued probabilities of the first layer of latent variables (rather than their binary activities) to train the second layer CRBM. We also use the real-valued probabilities of the second layer of latent variables (instead of binary activities) as input to the classifier. This suppresses noise, though we expect that thresholding would also give similar results.

Figure 3 shows consecutive individual frames reconstructed from Zernike moments sampled from a two-layer CDBN with 150 latent variables in each layer and an order of 3 for each layer, trained on sequences of Zernike moments up to order 40. The sequence has been initialized with 6 observations which were not part of the training set. By generalizing to unseen shape sequences, we see that the model is capable of modeling the dynamics of 2D shape rather than simply storing sequences of repeating frames.

4 Classification

Given a training set of videos and per-frame labels, a classification decision is obtained with a series of stages:

1. **Background subtraction** We pre-process each frame of video with a background subtraction algorithm (details are given in section 2). This yields binary shape sequences.

2. **Computing low-level features** The Zernike moments and additional low-level features (see section 3.1) are computed for each frame.

3. **Extraction of mid-level features** A CRBM with linear-Gaussian visible units and binary hidden units is trained on all sequences of low-level features (regardless of label). A single motion model is thus learned for the entire training dataset. We then pass each sequence through the trained CRBM and collect the sequences of real-valued hidden-unit posteriors, $p(h_{jt} = 1|\boldsymbol{v}_t)$ (i.e. we compute Eq. 3 for each frame of the low-level features).

4. **Extraction of high-level features** A CRBM with binary visible units and binary visible units is trained on all sequences of low-level features (i.e. the posteriors of the first CRBM while driven by the training data). Once the CRBM is trained, we then apply $p(h_{jt} = 1|\boldsymbol{v}_)$ (again through Eq. 3) for every frame of the mid-level features, where $\boldsymbol{h}$ are the hidden units of the topmost CRBM. We store the sequences of real-valued hidden-unit posteriors. These will be the input to the classifier.

5. **Training an SVM** We note that, up to this point, the labels have not been used. We now train a standard classifier such as an SVM, k-NN, or AdaBoost — in our case, an SVM — using the pairs of top-level features produced by the conditional DBN and the associated labels (from the original data).

When presented with a novel (i.e. test) example video, we proceed through the above stages, though we simply perform bottom-up inference in the trained conditional DBN (its parameters are not adjusted). Likewise, we simply perform prediction in the SVM using the top-level features as input.

5 Experimental results

Our proposed method has been evaluated on two different datasets:

- the well known Weizmann dataset [13] comprised of 93 short videos and 10 action classes (*bend, jack, forward jump, vertical jump, side jump, run, walk, skip, one handed wave, two handed wave*);

- a dataset which we collected ourselves and which contains 120 videos in 4 classes (*run, walk, vertical jump, sit down*). We attempted to make our dataset more difficult than Weizmann in three ways: i) different activities are performed in different directions with respect to the camera (see figure 4); ii) multiple people are present in each video simultaneously performing different activities; and iii) we include videos as short as between 3 and 10 frames.

We ran two kinds of experiments. Detection and recognition performance have been measured quantitatively through a widely used protocol employed for action recognition. As usual, when testing on a video showing a given subject, all videos of the same subject (even showing a different activity) have been removed from the training set. This ensures that the learned features do not depend on the appearance of the subjects, but rather on their motion only. The splits into training and test sets have been done per video. For the Weizmann dataset we report accuracy using 9-fold cross-validation.

Classification — classification performance is given on two different levels: per classification entity, i.e. per units of short sequences of 7 frames, as well as per whole video, where the latter decisions are obtained with a voting strategy. On the Weizmann dataset we obtain a classification rate of 94.7%, a result which has also been obtained by other methods very recently [45, 38]. We achieve a score of 84.2% per sequence of 7 frames, which indicates that activities can be recognized robustly from short sub sequences, mainly due to the CRBM.

Figure 4 shows some of the bounding boxes and shapes extracted with the frame differencing algorithms as well as the corresponding binary shape sequences. Although the shapes are only partially formed for some of the actions, in particular the ones for the class *hand waving*, classification performance is excellent on the short sequences.

Localization+recognition — The second experiment qualitatively evaluates performance in a multiple person environment. Figure 5 shows 16 consecutive frames from a video with 3 different people performing two different actions (walking and jumping). Although we do not rely on a tracker to determine robust positions of the different moving people, the decision on an individual level are impressive. A video showing the performance in more detail can be downloaded on our website[4].

The different design choices of our system have been set as follows: all binary shapes have been extracted with the frame differencing algorithm. The Zernike features have been calculated up to order 30. The motion model is a CDBN with two layers and Gaussian units in the first layer. The order of both layers was set to 3 (3 connections from the past to the current time instant), the first layer contained 800 hidden units and the second layer 400 hidden units. Classification was done with an SVM and RBF kernel. The system has been implemented in Matlab R2010b and includes parts of Taylor et al.'s CRBM code available

[4] `http://liris.cnrs.fr/christian.wolf/vids/st-shapes.avi`

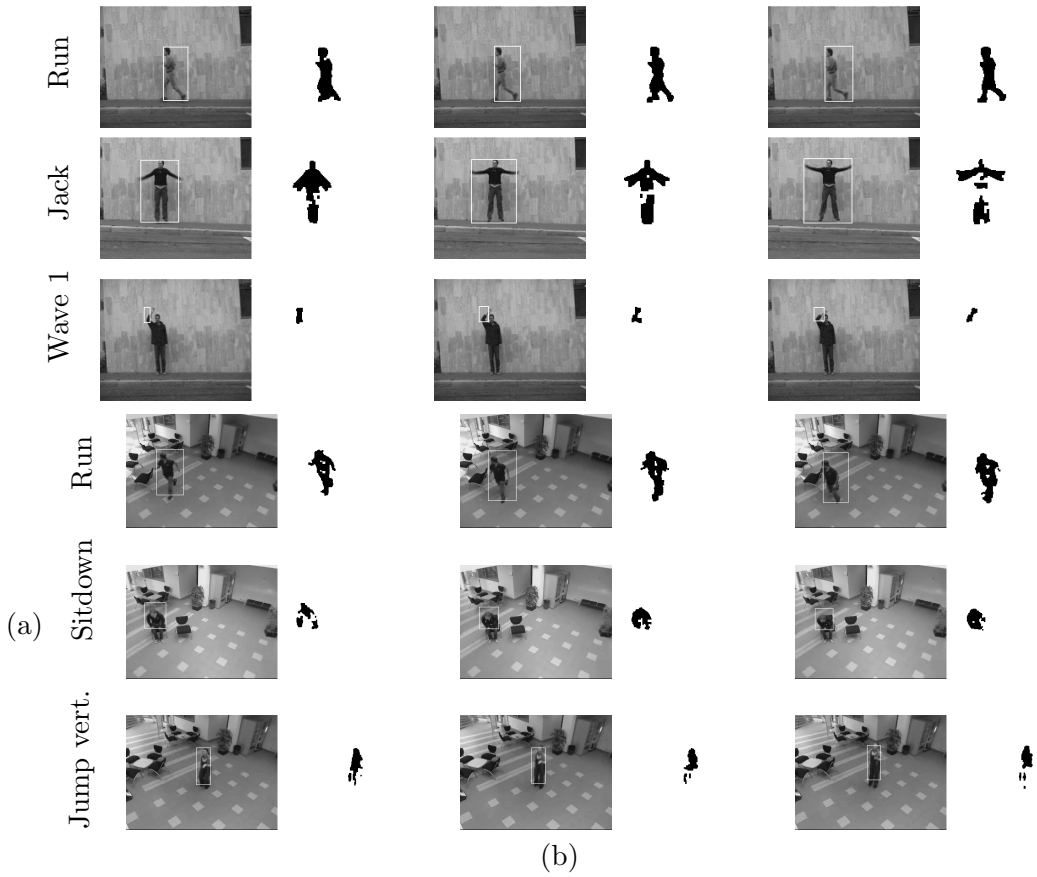

Figure 4: Segmentation and detection results on some consecutive frames of the two databases: (a) The Weizmann dataset (b) Our dataset.

Figure 5: Recognition results on 16 consecutive frames of a video of our dataset.

Method	Accuracy
Proposed method	**94.7**
Schindler and Gool [45]	100.0
Fathi and Mori [38]	100.0
Gorelick et al. [13]	99.6
Sun et al. [19]	97.8
Bregonzio et al. [20]	96.7
Niebles et al. [7]	90.0
Klaser et al. [58]	84.3
Scovanner et al. [59]	82.6
Niebles et al. [43]	72.8

(a)

Method	Accuracy
Proposed method	**91.7**

(b)

Table 1: Classification accuracy per video on the two standard datasets: (a) The Weizmann dataset; (b) The KTH dataset - subset D1.

online[5] as well as Chang et al.'s libsvm[6]. It runs with 1.4 frames per second on a laptop using a Intel Core 2 processor with 2.5Ghz and 4GB RAM. Realtime performance should be easily achievable if the system is re-implemented in C++.

6 Conclusion

We have presented a novel method for the recognition and classification of individual human actions which makes decisions on very short sequences of binary shapes. The system learns several layers of features and is able to classify a new unseen sequence quickly in one bottom-up pass of the multi-layer model. Due to the rich componential nature of the hidden states, the model is able to learn human motion efficiently from shape which results in highly discriminative features and excellent classification performance. In addition to classification performance, the fact that we have learned a dynamical model of human motion from shape alone is a notable result, as previous such approaches (e.g [50]) have only considered motion capture data. This has been achieved, in part, by using the Zernike decomposition. The current limitation of the system, addressed in future work, is its lack of robustness to occlusions. This is shared, to the best of our knowledge, with all established work.

Acknowlegements

This project was financed through the French National grant ANR-CaNaDA *Comportements Anormaux : Analyse, Dtection, Alerte*, No. 128, which is part of the call for projects CSOSG 2006 *Concepts Systemes et Outils pour la Securite Globale*.

We acknowledge that a large part of the calculations of this work have been performed on the computing clusters of the French National Institute of nuclear physics and particle physics[7].

[5] http://www.cs.nyu.edu/∼gwtaylor/publications/nips2006mhmublv

[6] http://www.csie.ntu.edu.tw/∼cjlin/libsvm

[7] `http://www.in2p3.fr`

References

[1] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer Vision and Image Understanding 115 (2011) 224–241.

[2] J. Aggarwal, M. Ryoo, Human activity analysis: A review, ACM Computing Surveys (to appear).

[3] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine Recognition of Human Activities: A Survey, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1473–1488.

[4] Y. Song, L. Concalves, P. Perona, Unsupervised learning of human motion, IEEE Tr. on PAMI 25 (7) (2003) 814–827.

[5] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: ICCV Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.

[6] I. Laptev, On space-time interest points, International Journal of Computer Vision (IJCV) 64 (2/3) (2005) 107–123.

[7] J. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporalwords, I.J. of Computer Vision 79 (3) (2008) 299–318.

[8] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: ICPR, Vol. 3, 2004, pp. 32–36 Vol.3.

[9] A. Oikonomopoulos, I. Patras, M. Pantic, An implicit spatiotemporal shape model for human activity localization and recognition, In CVPR 0 (2009) 27–33.

[10] M. S. Ryoo, J. K. Aggarwal, Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities, in: ICCV, 2009.

[11] A. Gilbert, J. Illingworth, R. Bowden, Mined Hierarchical Compound Features, International Journal of Computer Vision 33 (5) (2011) 883 – 897.

[12] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3d exemplars, in: ICCV, 2007, pp. 1–7.

[13] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Tr. on PAMI 29 (12) (2007) 2247–2253.

[14] L. Wang, X. Geng, C. Leckie, K. Ramamohanarao, Moving shape dynamics: A signal processing perspective, in: CVPR, 2008.

[15] Y. Ke, R. Sukthankar, M. Hebert, Efficient visual event detection using volumetric features, in: ICCV, 2005, pp. 166–173.

[16] K. Mikolajczyk, H. Uemura, Action recognition with motion-appearance vocabulary forest, In CVPR, 2008 0 (2008) 1–8.

[17] Z. Zhang, Y. Hu, S. Chan, L.-T. Chia, Motion context: A new representation for human action recognition, in: ECCV (4), 2008, pp. 817–829.

[18] J. Liu, S. Ali, M. Shah, Recognizing human actions using multiple features, in: CVPR, 2008.

[19] X. Sun, M. Chen, A. Hauptmann, Action recognition via local descriptors and holistic features, in: CVPR 2009 WS on Human Comm. Behav. Anal., 2009, pp. 58–65.

[20] M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space-time interest points, in: CVPR, 2009, pp. 1948–1955.

[21] E. Shechtman, M. Irani, Space-time behavior based correlation, in: CVPR, Vol. 1, 2005, pp. 405–412.

[22] H. J. Seo, P. Milanfar, Action Recognition from One Example., IEEE transactions on pattern analysis and machine intelligence 33 (5) (2011) 867 – 882.

[23] A.-P. Ta, C. Wolf, G. Lavoué, A. Baskurt, Recognizing and localizing individual activities through graph matching, in: International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2010.

[24] J. Liu, M. Shah, Learning human actions via information maximization, in: CVPR, 2008.

[25] A.-P. Ta, C. Wolf, G. Lavoué, A. Baskurt, J. Jolion, Pairwise features for human action recognition, in: I.C. on Pattern Recognition, 2010.

[26] K. Mikolajczyk, H. Uemura, Action recognition with appearancemotion features and fast search trees, Computer Vision and Image Understanding 115 (3) (2011) 426–438.

[27] Q. Shi, L. Cheng, L. Wang, A. Smola, Human Action Segmentation and Recognition Using Discriminative Semi-Markov Models, International Journal of Computer Vision 93 (1) (2010) 22–32.

[28] N. Cuntoor, B. Yegnanarayana, R. Chellappa, Activity modeling using event probability sequences, IEEE Tr. on image proc. 17 (4) (2008) 594–607.

[29] O. Boiman, M. Irani, Detect. irreg. in images and in video, International Journal of Computer Vision (IJCV) 74 (1) (2007) 17–31.

[30] T. Xiang, S. Gong, Activity based surveillance video content modelling, Pattern Recognition 41 (7) (2008) 2309–2326.

[31] T. Xiang, S. Gong, Incremental and adaptive abnormal behaviour detection, Computer Vision and Image Understanding (CVIU) 11 (1) (2008) 59–73.

[32] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, Semi-supervised adapted hmms for unusual event detection, in: CVPR, Vol. 1, 2005, pp. 611–618.

[33] H. Zhou, D. Kimber, Unusual event detection via multi-camera video mining, in: ICPR, Vol. 3, 2006, pp. 1161–1166.

[34] M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, R. Chellappa, Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian shape manifolds, Computer Vision and Image Understanding 115 (3) (2010) 439–455.

[35] L. Zhang, Z. Zeng, Q. Ji, Probabilistic image modeling with an extended chain graph for human activity recognition and image segmentation, IEEE Transactions on Image Processing (to appear).

[36] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: International Journal of Computer Vision (IJCV), 2007, pp. 1–8.

[37] G. W. Taylor, R. Fergus, Y. Lecun, C. Bregler, Convolutional Learning of Spatio-temporal Features, in: European conference on Computer vision, 2010.

[38] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: IEEE (Ed.), I.C. on CVPR, 2008, pp. 1–8.

[39] A. Dyana, S. Das, Trajectory representation using gabor features for motion-based video retrieval, Pattern Recognition Letters 30 (10) (2009) 877–892.

[40] C. Stauffer, W. Grimson, Learning patterns of activity using real-time tracking, IEEE Tr. on PAMI 22 (8) (2000) 747–757.

[41] M. S. Ryoo, J. K. Aggarwal, Stochastic Representation and Recognition of High-Level Group Activities, International Journal of Computer Vision 93 (2) (2010) 183–200.

[42] L. Wang, Y. Wang, W. Gao, Mining Layered Grammar Rules for Action Recognition, International Journal of Computer Vision 93 (2) (2010) 162–182.

[43] J. Niebles, F. Li, A hierarchical model of shape and appearance for human action classification, in: CVPR, 2007, pp. 1–8.

[44] C. Guillot, M. Taron, P. Sayd, Q. Pham, C. Tilmant, J.-M. Lavest, Background subtraction adapted to ptz cameras by keypoint density estimation, in: British Machine Vision Conference, 2010, pp. 34.1–34.10.

[45] K. Schindler, L. van Gool, Action snippets: How many frames does human action recognition require?, in: CVPR, 2008.

[46] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Tr. on PAMI 23 (3) (2001) 257–267.

[47] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: CVPR, Vol. 2, 1999, pp. 2246–2253.

[48] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. of IEEE 86 (11) (1998) 2278–2324.

[49] G. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Comput 18 (7) (2006) 1527–1554.

[50] G. W. Taylor, G. E. Hinton, S. Roweis, Modeling human motion using binary latent variables, Proc. NIPS 19.

[51] Y. Bengio, Learning deep architectures for ai, Foundations and Trends in Machine Learning 2 (1) (2009) 1–127.

[52] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical repr., in: ICML, 2009, pp. 609–616.

[53] A. Khotanzad, Y. Hong, Invariant image recognition by zernike moments, IEEE Tr. on PAMI 12 (5) (1990) 489–497.

[54] J. Revaud, G. Lavoué, A. Baskurt, Improving zernike moments comparison for opt. similarity and rotation angle retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 31 (4) (2009) 627–636.

[55] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, in: D. E. Rumelhart, J. L. McClelland, et al. (Eds.), Parallel Distributed Processing: Volume 1: Foundations, MIT Press, Cambridge, MA, 1986, pp. 194–281.

[56] M. Welling, M. Rosen-Zvi, G. Hinton, Exponential family harmoniums with an application to information retrieval., in: Proc. NIPS 17, 2005.

[57] G. Hinton, Training products of experts by minimizing contrastive divergence., Neural Comput 14 (8) (2002) 1771–1800.

[58] A. Kläser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Machine Vision Conf., 2008, pp. 995–1004.

[59] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: ACM Multimedia, 2007, pp. 357–360.