Energy-Based Localized Anomaly Detection in Video Surveillance

Hung Vu[1], Tu Dinh Nguyen[1], Anthony Travers[2(✉)], Svetha Venkatesh[1], and Dinh Phung[1]

[1] Center for Pattern Recognition and Data Analytics, Deakin University, Geelong, Australia
{hungv,tu.nguyen,svetha.venkatesh,dinh.phung}@deakin.edu.au
[2] Defence Science and Technology Organization (DSTO), Melbourne, Australia
Anthony.Travers@dsto.defence.gov.au

Abstract. Automated detection of abnormal events in video surveillance is an important task in research and practical applications. This is, however, a challenging problem due to the growing collection of data without the knowledge of what to be defined as "abnormal", and the expensive feature engineering procedure. In this paper we introduce a unified framework for anomaly detection in video based on the restricted Boltzmann machine (RBM), a recent powerful method for unsupervised learning and representation learning. Our proposed system works directly on the image pixels rather than hand-crafted features, it learns new representations for data in a completely unsupervised manner without the need for labels, and then reconstructs the data to recognize the locations of abnormal events based on the reconstruction errors. More importantly, our approach can be deployed in both offline and streaming settings, in which trained parameters of the model are fixed in offline setting whilst are updated incrementally with video data arriving in a stream. Experiments on three publicly benchmark video datasets show that our proposed method can detect and localize the abnormalities at pixel level with better accuracy than those of baselines, and achieve competitive performance compared with state-of-the-art approaches. Moreover, as RBM belongs to a wider class of deep generative models, our framework lays the groundwork towards a more powerful deep unsupervised abnormality detection framework.

1 Introduction

Developing intelligent video surveillance systems has been attracting research and application interest in computer vision community [11,15]. One of the most important surveillance problems is to automatically detect and analyze the abnormal events in video streams. The anomalous events are commonly assumed to be rare, irregular or significantly different from the others [15]. Examples include accesses

This work was partially supported by the Australian Research Council under the Discovery Project DP150100031 and the DST Group.

J. Kim et al. (Eds.): PAKDD 2017, Part I, LNAI 10234, pp. 641–653, 2017.
DOI: 10.1007/978-3-319-57454-7_50

to restricted area, leaving strange packages, movements in wrong direction, which can be captured by the camera monitoring systems in airports, car parks, stations and public spaces in general. Identifying the anomaly behaviors allows early intervention and in-time support to reduce the consequent cost.

The existing literature of anomaly detection on video data offers two approaches: supervised learning and unsupervised learning. Typical supervised methods include support vector data description [17], mixture of dynamic texture models [7] and supervised sparse coding [8], that use data labeled as *normal* to learn the model parameters and then judge the testing data as *abnormal* based on their probabilities or distances to the model. The methods in this approach, however, require the training data annotated with labels which are labor-intensive for large-scale data, rendering them inapplicable to the video streaming from surveillance systems where the amount of data grows super-abundantly. Moreover, it is also infeasible to model the diversity of normal event types in practice.

The unsupervised learning approach overcomes this issue by modeling the data without the need for labels. Typical methods include principle component analysis (PCA) [13], one-class support vector machines (OC-SVM) [1,16], Gaussian mixture models (GMM) [2,9], dynamic sparse coding [18], Bayesian non-parametric factor analysis (BNF) [10] and scan statistics [6]. The PCA learns a linear transformation to a lower dimensional linear space called "residual subspace", and then detect the anomalies using the residual signals of the projection of this data onto the residual subspace. The OC-SVM learns a hyperplane that achieves maximum separation between the normal data points and the origin, and then use the distance from a data point to this hyperplane to determine the abnormality. Alternatively, the GMM is a probabilistic method that models the data distribution, and use the posterior as the signal for anomaly detection. Other methods, such as sparse coding [18], compute the anomaly signal as the error of reconstructing data from a learned dictionary. Meanwhile, the BNF detects anomaly events using rareness scores that are based on the contributions of latent factors to reconstruct the scene. Scan statistics [6] measures the difference between statistical information inside and outside a region to discover anomalous objects. These methods, however, critically depend on the hand-crafted, low-level features extracted for video and image, such as histograms of oriented gradients (HOG) [18], optical flow features [6,9,13,16,17] and histograms of optical flow (HOF) [18]. The hand-crafted features rely on the design of preprocessing pipeline and data transformation, which is labor-intensive and normally requires exhaustive prior knowledge.

Recently there have been several studies that use deep learning techniques to automatically learn high-level representations for data to avoid the requirement of domain experts in designing features. When applying to anomaly detection for video data, the common approach is to extract features at the first stage (cf. autoencoders in [12,16]), and then use a separate classifier (e.g., OC-SVM) for detection at the second stage. An alternative method is to use the convolutional autoencoder (ConvAE) [4] to optimize the error when reconstructing the training data, and then use the reconstruction errors to recognize the abnormalities in testing data. Training these methods, however, is non-trivial due to their complicated architectures with multiple models or multiple layers.

In this paper, we propose a unified framework for anomaly detection in video based on the restricted Boltzmann machine (RBM) [3,5], a recent powerful energy-based method for unsupervised learning and representation learning. Our proposed system employs RBMs as core modules to model the complex distribution of data, capture the data regularity and variations, as a result effectively reconstruct the normal events that occur frequently in the data. The idea is to use the errors of reconstructed data to recognize the abnormal objects or behaviors that deviate significantly from the common. This is similar to the idea of using ConvAE. However, the key difference between our method and that approach is the ConvAE is a deterministic method that faces difficulty in modeling data which follow a set of probabilistic distributions, whilst ours is based on RBM, is probabilistic energy-based method that directly models the data distribution and captures data regularity.

Our framework is trained in a completely unsupervised manner that does not involve any explicit labels or implicit knowledge of what to be defined as abnormal. In addition, it can work directly on raw pixels without the need for expensive feature engineering procedure. Another advantage of our method is the capability of detecting the exact boundary of local abnormality in the video frame. To handle the video data coming in a stream, we further extend our method to incrementally update parameters without retraining the models from scratch. Our solution can be easily deployed in arbitrary surveillance streaming setting without the expensive calibration requirement.

We qualitatively and quantitatively evaluate the performance of our anomaly detection framework through comprehensive experiments on three real-world datasets. Our primary target is to investigate the capabilities of capturing data regularity, reconstructing the data and detecting local abnormalities of our system. The experimental results show that our proposed method can effectively reconstruct the data regularity, and thus detect and localize the abnormalities at pixel level with better accuracies than those of baselines, and competitive performance compared with state-of-the-art approaches.

In short, our contributions are: (i) a novel unified RBM-based framework that can act as a completely unsupervised model on raw pixels; thus there is no need to extract hand-crafted features; (ii) an incremental version of our system that can efficiently work in a streaming setting; and (iii) a comprehensive evaluation of the effectiveness of our method on real-world video surveillance application.

2 Framework

We now describe our energy-based framework to detect abnormal events in video surveillance data. First we briefly review restricted Boltzmann machines that are the key components in our proposed system. We then present our framework and the extension for streaming video data.

2.1 Restricted Boltzmann Machine

A restricted Boltzmann machine (RBM) [3,14] is a bipartite undirected graphical model wherein the bottom layer contains observed variables called visible units

and the top layer consists of latent *representational variables*, known as hidden units. Two layers are fully connected but there is no connection within layers.

Model Representation. More formally, assume a binary RBM with M visible units and K hidden units, let $\boldsymbol{x}$ denote the set of visible variables: $\boldsymbol{x} = [x_1, x_2, \ldots, x_M]^\top \in \{0,1\}^M$ and $\boldsymbol{h}$ indicate the set of hidden ones: $\boldsymbol{h} = [h_1, h_2, \ldots, h_K]^\top \in \{0,1\}^K$. The RBM assigns an energy function for a joint configuration over the state $(\boldsymbol{x}, \boldsymbol{h})$ as:

$$E(\boldsymbol{x}, \boldsymbol{h}; \psi) = -\left(\boldsymbol{a}^\top \boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{h} + \boldsymbol{x}^\top \boldsymbol{W} \boldsymbol{h}\right) \tag{1}$$

where $\psi = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{W}\}$ is the set of parameters. $\boldsymbol{a} = [a_m]_M \in \mathbb{R}^M$, $\boldsymbol{b} = [b_k]_K \in \mathbb{R}^K$ are the biases of hidden and visible units respectively, and $\boldsymbol{W} = [w_{mk}]_{M \times K} \in \mathbb{R}^{M \times K}$ represents the weights connecting the hidden and visible units. The model admits a Boltzmann distribution (also known as Gibbs distribution) as follows:

$$p(\boldsymbol{x}, \boldsymbol{h}; \psi) = \frac{1}{\mathcal{Z}(\psi)} \exp\{-E(\boldsymbol{x}, \boldsymbol{h}; \psi)\} \tag{2}$$

where $\mathcal{Z}(\psi) = \sum_{\boldsymbol{x}, \boldsymbol{h}} \exp\{-E(\boldsymbol{x}, \boldsymbol{h}; \psi)\}$ is the normalization constant, also called partition function. This guarantees that the $p(\boldsymbol{x}, \boldsymbol{h}; \psi)$ is a proper density function.

Since the network has no intra-layer connections, units in one layer become conditionally independent given the other layer. Thus the conditional distributions over visible and hidden units are factorized as:

$$p(\boldsymbol{h} \mid \boldsymbol{x}; \psi) = \prod_{k=1}^{K} p(h_k \mid \boldsymbol{x}; \psi) \tag{3}$$

$$p(\boldsymbol{x} \mid \boldsymbol{h}; \psi) = \prod_{m=1}^{M} p(x_m \mid \boldsymbol{h}; \psi) \tag{4}$$

Parameter Estimation. As an energy-based model, the learning goal of RBM is to minimize the energy in Eq. (1) of the observed data. As the visible probability is inversely proportional to the energy as shown in Eq. (2), it is equivalent to maximize the following log-likelihood of data: $\log p(\boldsymbol{x}; \psi) = \log \sum_{\boldsymbol{h}} p(\boldsymbol{x}, \boldsymbol{h}; \psi)$. The parameters are updated in a gradient ascent fashion as follows:

$$\psi \leftarrow \psi + \eta \left(\mathbb{E}_{p(\boldsymbol{x}, \boldsymbol{h}; \psi)} [\nabla_\psi E(\boldsymbol{x}, \boldsymbol{h}; \psi)] - \mathbb{E}_{p(\boldsymbol{h} \mid \boldsymbol{x}; \psi)} [\nabla_\psi E(\boldsymbol{x}, \boldsymbol{h}; \psi)]\right)$$

for a learning rate $\eta > 0$. Here $\mathbb{E}_{p(\boldsymbol{x}, \boldsymbol{h}; \psi)}$ denotes the expectation with respect to the full model distribution and $\mathbb{E}_{p(\boldsymbol{h} \mid \boldsymbol{x}; \psi)}$ the data expectation with respect to the conditional distribution given the observed $\boldsymbol{x}$. Whilst $\mathbb{E}_{p(\boldsymbol{h} \mid \boldsymbol{x}; \psi)}$ can be computed efficiently, $\mathbb{E}_{p(\boldsymbol{x}, \boldsymbol{h}; \psi)}$ is generally intractable. Thus we must resort to approximate methods, and in this paper, we choose contrastive divergence (CD) [5] as it proves to be fast and accurate.

Data Reconstruction. Once the model parameters ψ has been learned, the RBM can project an input data $\boldsymbol{x}$ onto the hidden space to obtain the new representation $\tilde{\boldsymbol{h}} = [\tilde{h}_1, \tilde{h}_2, \ldots, \tilde{h}_K]^\top$ where $\tilde{h}_k$ is shorthand for the posterior

$\tilde{h}_k = p\left(h_k = 1 \mid \boldsymbol{x}\right) = \sigma\left(b_k + \sum_m w_{mk} x_m\right)$, in which $\sigma(x)$ is the sigmoid function $\sigma(x) = (1 + e^{-x})^{-1}$. This hidden posterior vector is then mapped back into the input space to form the reconstructed data $\tilde{\boldsymbol{x}} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_M]^\top$ where $\tilde{x}_m = p\left(x_m = 1 \mid \tilde{\boldsymbol{h}}; \psi\right) = \sigma\left(a_m + \sum_k w_{mk} \tilde{h}_k\right)$, similarly to the hidden posterior. These projection and mapping are very efficient due to the nice factorizations in Eqs. (3, 4).

2.2 Anomaly Detection Using RBM

We now describe our proposed framework that is based on the RBM to detect anomaly events for each frame in video data. In general, our system is a two-phase pipeline: training phase and detecting phase. Particularly in the training phase, our model: (i) takes a series of video frames in the training data as a collection of images, (ii) divides each image into patches, (iii) gathers similar patches into clusters, and (iv) learns separate RBM for each cluster using the image patches. The detecting phase consists of three steps: (i) collecting image patches in the testing video for each cluster, and then using the learned RBM to reconstruct the data for the corresponding cluster of patches, (ii) proposing the regions that are *potential* to be abnormal by applying a predefined threshold to reconstruction errors, and then finding connected components of these candidates and filtering out those too small, and (iii) updating the model incrementally for the data stream. The overview of our framework is illustrated in Fig. 1. In what follows, we describe training and detecting phases in more details.

Training Phase. Assume that the training data consists of N video frames with the size of $H \times W$ pixels, let denote $\mathcal{D} = \{\boldsymbol{x}_t \in \mathbb{R}^{H \times W}\}_{t=1}^{N}$. In real-life video surveillance data, $H \times W$ is usually very large (e.g., hundreds of thousand pixels), hence it is often infeasible for a single RBM to handle such high-dimensional image. This is because the high-dimensional input requires a more complex model with an extremely large number of parameters (i.e., millions). This makes the parameter learning more difficult and less robust since it is hard to control the bounding of hidden activation values. Thus the hidden posteriors are easily collapsed into either zeros or ones, and no more learning occurs.

To tackle this issue, one can reduce the data dimension using dimensionality reduction techniques or by subsampling the image to smaller size. This solution, however, is computational demanding and may lose much information of the original data. In this work we choose to apply RBMs directly to raw imaginary pixels whilst try to preserve information. To that end, we train our model on $h \times w$ patches where we divide each image $\boldsymbol{x}_t$ into a grid of $N_h \times N_w$ patches: $\boldsymbol{x}_t = \{\boldsymbol{x}_t^{i,j} \mid 1 \le i \le N_h, 1 \le j \le N_w\}$. This approach greatly reduces the data dimensionality and hence requires smaller models. One way is to learn independent RBMs on patches at each location (i, j). However, this would result in an excessive number of models, for example, 400 RBMs to work on the 240×360 image resolution and 12×18 patch size, hence leading to very high computational complexity and memory demand.

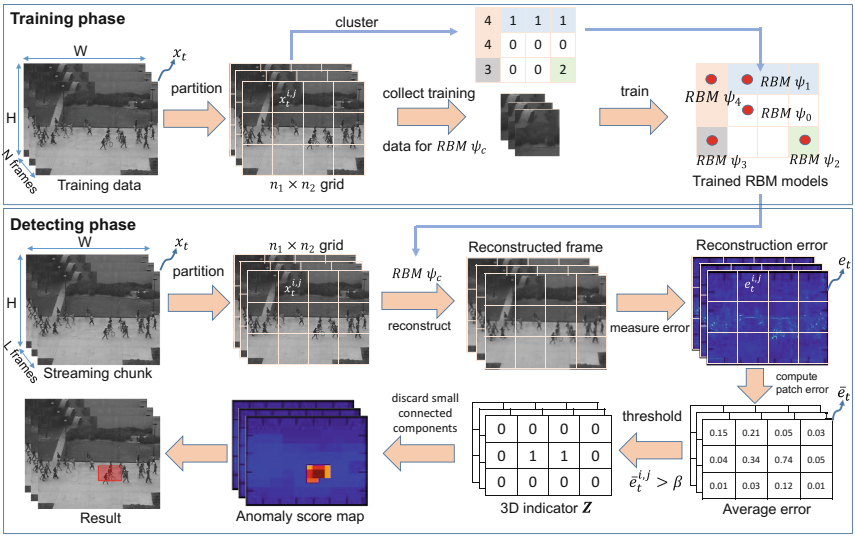

Fig. 1. The overview of our proposed framework.

Our solution is to reduce the number of models by grouping all similar patches from different locations for learning a single model. We observe that it is redundant to train a separate model for each location of patches since most adjacent patches such as pathways, walls and nature strips in surveillance scenes have similar appearance and texture. Thus we first train a RBM with a small number of hidden units ($K = 4$) on all patches $\{\boldsymbol{x}_t^{i,j}\}$ of all video frames. We then compute the hidden posterior $\tilde{\boldsymbol{h}}$ for each image patch $\boldsymbol{x}_t^{i,j}$ and binarize it to obtain the binary vector: $\tilde{\boldsymbol{h}} = \left[\mathbb{I}\left(\tilde{h}_1 > 0.5\right), \ldots, \mathbb{I}\left(\tilde{h}_K > 0.5\right)\right]$ where $\mathbb{I}(\bullet)$ is the indicator function. Next this binary vector is converted to an integer value in decimal system, e.g., 0101 converted to 5, which we use as the *pseudo-label* $\lambda_t^{i,j}$ of the cluster of the image patch $\boldsymbol{x}_t^{i,j}$. The cluster label $c^{i,j}$ for all patches at location (i, j) is chosen by voting the pseudo-labels over all N frames: $\lambda_1^{i,j}, \lambda_2^{i,j}, \ldots, \lambda_N^{i,j}$. Let C denote the number of unique cluster labels in the set $\{c^{i,j} \mid 1 \leq i \leq N_h, 1 \leq j \leq N_w\}$, we finally train C independent RBMs with a larger number of hidden units ($K = 100$), each with parameter set ψ_c for all patches with the same cluster label c.

Detecting Phase. Once all RBMs have been learned using the training data, they are used to reveal the irregular events in the testing data. The pseudocode of this phase is given in Algorithm 1. Overall, there are three main steps: reconstructing the data, detecting local abnormal objects and updating models incrementally. In particular, the stream of video data is first split into chunks of L non-overlapping frames, each denoted by $\{\boldsymbol{x}_t\}_{t=1}^{L}$. Each patch $\boldsymbol{x}_t^{i,j}$ is then reconstructed to obtain the reconstruction $\tilde{\boldsymbol{x}}_t^{i,j}$ using the learned RBM with parameters $\psi_{c^{i,j}}$, and all together form the reconstructed data $\tilde{\boldsymbol{x}}_t$ of the frame $\boldsymbol{x}_t$. The reconstruction error $\boldsymbol{e}_t = [\boldsymbol{e}_t^{i,j}] \in \mathbb{R}^{H \times W}$ is then computed as: $\boldsymbol{e}_t^{i,j} = |\boldsymbol{x}_t^{i,j} - \tilde{\boldsymbol{x}}_t^{i,j}|$.

To detect anomal pixels, one can compare the reconstruction error $\boldsymbol{e}_t$ with a given threshold. This approach, however, may produce many false alarms when normal pixels are reconstructed with high errors, and may fail to cover the entire anomaly objects in such a case that they are fragmented into isolated high error parts. Our solution is to work on the average error $\bar{e}_t^{i,j} = ||\boldsymbol{e}_t^{i,j}||_2 / (h \times w)$ over patches rather than individual pixels. These errors are then compared with a predefined threshold β. All pixels in $\boldsymbol{x}_t^{i,j}$ are considered abnormal if $\bar{e}_t^{i,j} \geq \beta$.

Applying the above procedure and then concatenating L frames, we obtain a binary 3D rectangle $\boldsymbol{Z} \in \{0,1\}^{L \times H \times W}$ wherein $z_{i,j,k} = 1$ indicates the abnormal voxel whilst $z_{i,j,k} = 0$ the normal one. Throughout the experiments, we observe that most of abnormal voxels in $\boldsymbol{Z}$ are detected correctly, but there still exist several small groups of voxels are incorrect. We further filter out these false positive voxels by connecting all their *related* neighbors. More specifically, we first build a sparse graph whose nodes are abnormal voxels $z_{i,j,k} = 1$ and edges are the connections of these voxels with their abnormal neighbors $z_{i+u,j+v,k+t} = 1$ where $u, v, t \in \{-1, 0, 1\}$ and $|u| + |v| + |t| > 0$. We then find all connected components in this graph, and discard small components spanning less than γ contiguous frames. The average error $\bar{e}_t^{i,j}$ after this component filtering step can be used as final anomaly score.

Algorithm 1. RBM anomaly detection

Input: Video chunk $\{\boldsymbol{x}_t\}_{t=1}^{L}$, models $\{\psi_c\}_{c=1}^{C}$, thresholds β and γ
Output: Detection $\boldsymbol{Z}$, score $\{\bar{e}_t^{i,j}\}$

for $t \leftarrow 1, \ldots, L$ **do**
  for $\boldsymbol{x}_t^{i,j} \in \boldsymbol{x}_t$ **do**
    $\tilde{\boldsymbol{x}}_t^{i,j} \leftarrow$reconstruct$(\boldsymbol{x}_t^{i,j}, \psi_{c^{i,j}})$
    $\boldsymbol{e}_t^{i,j} \leftarrow |\boldsymbol{x}_t^{i,j} - \tilde{\boldsymbol{x}}_t^{i,j}|$
    $\bar{e}_t^{i,j} \leftarrow \frac{1}{h \times w} \left\|\boldsymbol{e}_t^{i,j}\right\|_2$
    if $\bar{e}_t^{i,j} \geq \beta$ **then**
      for $p \in \boldsymbol{x}_t^{i,j}$ **do**
        $\boldsymbol{Z}(p) \leftarrow 1$
      end for
    else
      for $p \in \boldsymbol{x}_t^{i,j}$ **do**
        $\boldsymbol{Z}(p) \leftarrow 0$
      end for
    end if
  end for
  for $c \leftarrow 1, \ldots, C$ **do**
    $\boldsymbol{X}_t^c \leftarrow \{\boldsymbol{x}_t^{i,j} \mid c^{i,j} = c\}$
    $\psi_c \leftarrow$ updateRBM$(\boldsymbol{X}_t^c, \psi_c)$
  end for
end for
$\boldsymbol{Z} \leftarrow$remove_small_components$(\boldsymbol{Z}, \gamma)$

In the scenario of streaming videos, the scene frequently changes over time and it could be significantly different from those are used to train RBMs. To tackle this issue, we extend our proposed framework to enable the RBMs to adapt themselves to the new video frames. For every incoming frame t, we extract the image patches and update the parameters $\psi_{1:C}$ of C RBMs in our framework following the procedure in the training phase. Recall that the RBM parameters are updated iteratively using gradient ascent, thus here we use several epochs to ensure the information of new data are sufficiently captured by the models.

One problem is the anomalous objects can be presented in different sizes in the video. To deal with this issue, we apply our framework to the video data at different scales whilst keeping the same patch size $h \times w$. This would help the patch partially or entirely cover objects at certain scales. To that end, we rescale the original video into different resolutions, then employ the same procedure

above to compute the average reconstruction error map $\bar{e}_t$ and 3D rectangular indicators $\boldsymbol{Z}$. The average error maps are then aggregated into one matrix using max operation. Likewise, indicator tensors are merged into one before finding the connected components. We also use overlapping patches to localize anomalous objects more accurately. Pixels in the overlapping regions are averaged when combining patches into the whole map.

3 Experiment

In this section, we empirically evaluate the performance of our anomaly detection framework both qualitatively and quantitatively. Our aim is to investigate the capabilities of capturing data regularity, reconstructing the data and detecting local abnormalities of our system. For quantitative analysis, we compare our proposed method with several up-to-date baselines.

We use 3 public datasets: UCSD Ped 1, Ped 2 [7] and Avenue [8]. Under the unsupervised setting, we disregard labels in the training videos and train all methods on these videos. The learned models are then evaluated on the testing videos by computing 2 measures: area under ROC curve (AUC) and equal error rate (EER) at frame-level (no anomaly object localization evaluation) and pixel-level (40% of ground-truth anomaly pixels are covered by detection), following the evaluation protocol used in [7] and at dual-pixel level (pixel-level constraint above and at least α percent of detection is true anomaly pixels) in [12]. Note that pixel-level is a special case of dual-pixel where $\alpha = 0$. Since the videos are provided at different resolution, we first resize all into the same size of 240×360.

For our framework, we duplicate and rescale video frames to multiscale copies with the ratios of 1.0, 0.5 and 0.25, and then use 12×18 image patches with 50% overlapping between two adjacent patches. Each RBM now consists of 216 visible units and 4 hidden units for clustering step whilst 100 hidden units for training and detecting phases. All RBMs are trained using CD_1 with learning rate $\eta = 0.1$. To simulate the streaming setting, we split testing videos in non-overlapping chunks of $L = 20$ contiguous frames and use 20 epochs to incrementally update parameters of RBMs. The thresholds β and γ to determine anomaly are set to 0.003 and 10 respectively. Those hyperparameters have been tuned to reduce false alarms and to achieve the best balanced AUC and EER scores.

3.1 Region Clustering

In the first experiment, we examine the clustering performance of RBM. Figure 2 shows the cluster maps discovered by RBM on three datasets. Using 4 hidden units, the RBM can produce a maximum of 16 clusters, but in fact, the model returns less and varied number of clusters for different datasets at different scales. For example, (6, 7, 10) similar regions at scales (1.0, 0.5, 0.25) are found for Ped 1 dataset, whilst these numbers for Ped 2 and Avenue dataset are (9, 9, 8) and (6, 9, 9) respectively. This suggests the capability of automatically selecting the appropriate number of clusters of RBM.

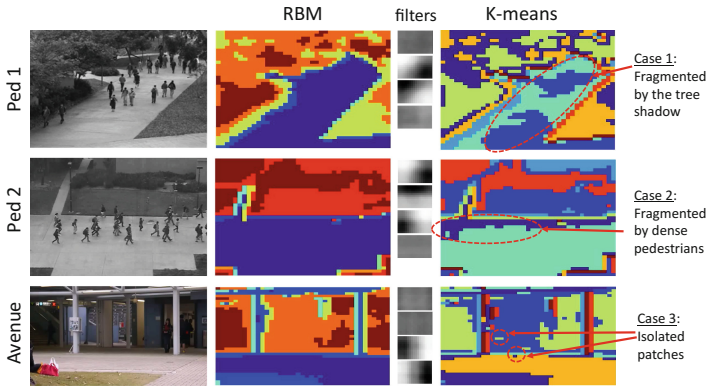

Fig. 2. Clustering result on some surveillance scenes at the first scale: (first column) example frames; (second) cluster maps produced by RBM; (third) filters learned by RBM; and (fourth) cluster maps produced by k-means.

For comparison, we run k-means algorithm with $k = 8$ clusters, the average number of clusters of RBM. It can be seen from Fig. 2 that the k-means fails to connect large regions which are fragmented by the surrounding and dynamic objects, for example, the shadow of tree on the footpath (Case 1), pedestrians walking at the upper side of the footpath (Case 2). It also assigns several wrong labels to small patches inside a larger area as shown in Case 3. By contrast, the RBM is more robust to the influence of environmental factors and dynamic foreground objects, and thus produces more accurate clustering results. Taking a closer look at the filters learned by RBM at the third column in the figure, we can agree that the RBM learns the basic features such as homogeneous regions, vertical, horizontal, diagonal edges and corners, which then can be combined to construct the entire scene.

3.2 Data Reconstruction

We next demonstrate the capability of our framework on the data reconstruction. Figure 3 shows an example of reconstructing the video frame in Avenue dataset. Here the abnormal object is a girl walking toward the camera. It can be seen that our model can correctly locate this outlier behavior based on the reconstruction errors shown in Figure 3(c) and (d). This is because the RBM can capture the data regularity, thus produces low reconstruction errors for regular objects and high errors for irregular or anomalous ones as shown in Figure 3(b) and (c).

To examine the change of reconstruction errors in a stream of video frames, we visualize the maximum average reconstruction error in a frame as a function of frame index as shown in Fig. 4. The test video #1 in UCSD Ped 1 dataset contains some normal frames of walking on a footpath, followed by the appearance of a cyclist moving towards the camera. Our system could not detect the emergence of the cyclist since the object is too small and cluttered by many surrounding

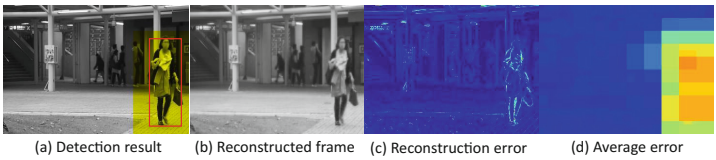

Fig. 3. Data reconstruction of our method on Avenue dataset: (a) the original frame with detected o2utlier female (yellow region) and ground-truth (red rectangle), (b) reconstructed frame, (c) reconstruction error image, (d) average reconstruction errors of patches. (Color figure online)

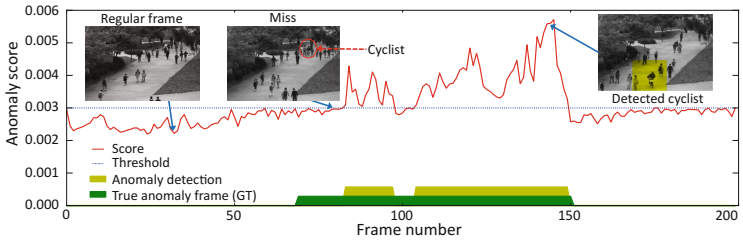

Fig. 4. Average reconstruction error per frame in test video #1 of UCSD Ped 1 dataset. The shaded green region illustrates anomalous frames in the ground truth, while the yellow anomalous frames detected by our method. The blue line shows the threshold. (Color figure online)

pedestrians. However, after several frames, the cyclist is properly spotted by our system with the reconstruction errors far higher than the threshold.

3.3 Anomaly Detection Performance

In the last investigation, we compare our offline RBM framework and its streaming version (called S-RBM) with the unsupervised methods for anomaly detection in the literature. We use 4 baselines for comparison: principal component analysis (PCA), one-class support vector machine (OC-SVM) [1], gaussian mixture models (GMM), and convolutional autoencoder (ConvAE) [4]. We use the variant of PCA with optical flow features from [13], and adopt the results of ConvAE from the original work [4]. The results of ConvAE are already compared with recent state-of-the-art baselines including supervised methods.

We follow similar procedures to what of our proposed framework for OC-SVM and GMM, but apply these baselines on image patches clustered by k-means. The kernel width and lower bound of the fraction of support vectors of OC-SVM are set to 0.1 and 10^{-4} respectively. In GMM model, the number of Gaussian components is set to 20 and the anomaly threshold is -50. These hyperparameters are also tuned to obtain the best cross-validation results. It is noteworthy that it is not straightforward to implement the incremental versions of the baselines, thus we do not include them here.

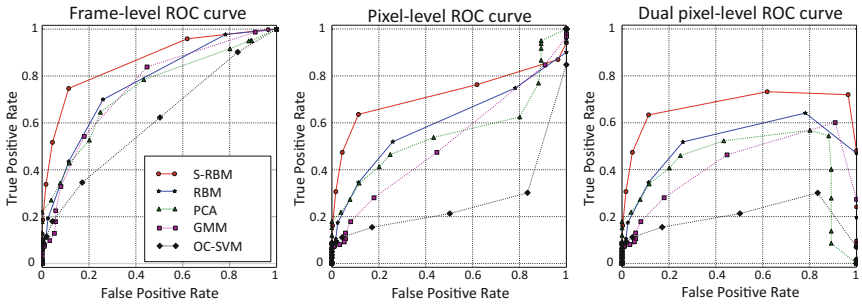

Fig. 5. Comparison ROC curves on UCSD Ped 2. Three figures share the same legend. Higher curves indicate better performance. It is notable that, unlike frame and pixel-level evaluations, dual-pixel level curves may end at any points lower than (1,1).

The ROC curves are shown in Fig. 5 whilst AUC and EER scores are reported in Table 1. Both RBM and S-RBM outperform the PCA, OC-SVM, GMM with higher AUC and lower EER scores. Specially, our methods can produce higher AUC scores at dual pixel-level which shows better quality in localizing anomaly regions. Additionally, S-RBM achieves fairly comparable results with the ConvAE. It is noteworthy that the ConvAE is a 12-layer deep architecture consisting of sophisticated connections between its convolutional and pooling layers. On the other hand, our RBM anomaly detector has only two layers, but obtains a respectable performance. We believe that our proposed framework is a promising system to detect abnormalities in video surveillance applications.

Table 1. Anomaly detection results (AUC and EER) at frame-level, pixel-level and dual pixel-level ($\alpha = 5\%$) on 3 datasets. Higher AUC and lower EER indicate better performance. Meanwhile, high dual-pixel values point out more accurate localization. We do not report EER for dual-pixel level because this number do not always exist. Best scores are in bold. Note that the frame-level results of ConvAE are taken from [4], but the pixel-level and dual-pixel level results are not available.

	Ped1					Ped2					Avenue				
	Frame		Pixel		Dual	Frame		Pixel		Dual	Frame		Pixel		Dual
	AUC	EER	AUC	EER	AUC	AUC	EER	AUC	EER	AUC	AUC	EER	AUC	EER	AUC
PCA	60.28	43.18	25.39	39.56	8.76	73.98	29.20	55.83	24.88	44.24	74.64	30.04	52.90	37.73	43.74
OC-SVM	59.06	42.97	21.78	37.47	11.72	61.01	44.43	26.27	26.47	19.23	71.66	33.87	33.16	47.55	33.15
GMM	60.33	38.88	36.64	35.07	13.60	75.20	30.95	51.93	18.46	40.33	67.27	35.84	43.06	43.13	41.64
ConvAE	**81.00**	**27.90**	-	-	-	**90.00**	21.70	-	-	-	70.20	**25.10**	-	-	-
RBM	64.83	37.94	41.87	36.54	16.06	76.70	28.56	59.95	19.75	46.13	74.88	32.49	43.72	43.83	41.57
S-RBM	70.25	35.40	**48.87**	**33.31**	**22.07**	86.43	**16.47**	**72.05**	**15.32**	**66.14**	**78.76**	27.21	**56.08**	**34.40**	**53.40**

4 Conclusion

We have presented a unified energy-based framework for video anomaly detection. Our method is based on RBMs to capture data regularity, and hence can distinguish and localize the irregular events. Our system is trained directly on the image pixels in a completely unsupervised manner. For video streaming, we further introduce a streaming version of our method that can incrementally update the parameters when new video frames arrive. Experimental results on several benchmark datasets show that the proposed method outperforms typical unsupervised baselines and achieves competitive performance compared with state-of-the-art method for anomaly detection.

Finally we note that our proposed approach is designed so that multiple RBMs are trained to capture different image statistics localized at different regions. Thus it is immediately amendable to a distributed and parallel implementation for a scalable system. Furthermore, as RBM belongs to a wider class of deep generative models, our framework is readily generalized to a more powerful deep unsupervised abnormality detection framework.

References

1. Amer, M., Goldstein, M., Abdennadher, S.: Enhancing one-class support vector machines for unsupervised anomaly detection. In: SIGKDD, pp. 8–15 (2013)
2. Basharat, A., Gritai, A., Shah, M.: Learning object motion patterns for anomaly detection and improved object detection. In: CVPR (2008)
3. Freund, Y., Haussler, D.: Unsupervised learning of distributions on binary vectors using two layer networks. Technical report, Santa Cruz, CA, USA (1994)
4. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: CVPR 2016 (2016)
5. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Comput. **14**(8), 1771–1800 (2002)
6. Hu, Y., Zhang, Y., Davis, L.S.: Unsupervised abnormal crowd activity detection using semiparametric scan statistic. In: CVPRW, pp. 767–774 (2013)
7. Li, W.X., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. PAMI **36**(1), 18–32 (2014)
8. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: ICCV (2013)
9. Lu, T., Wu, L., Ma, X., Shivakumara, P., Tan, C.L.: Anomaly detection through spatio-temporal context modeling in crowded scenes. In: ICPR (2014)
10. Nguyen, V., Phung, D., Pham, D.S., Venkatesh, S.: Bayesian nonparametric approaches to abnormality detection in video surveillance. Ann. Data Sci. (AoDS) **2**(1), 21–41 (2015)
11. Oluwatoyin, P.P., Wang, K.: Video-based abnormal human behavior recognition - a review. IEEE Trans. Syst. Man Cybern. 865–878 (2012)
12. Sabokrou, M., Fathy, M., Hosseini, M.: Real-time anomalous behavior detection and localization in crowded scenes. In: CVPRW (2015)
13. Saha, B., Pham, D.S., Lazarescu, M., Venkatesh, S.: Effective anomaly detection in sensor networks data streams. In: ICDM, pp. 722–727 (2009)

14. Smolensky, P.: Information processing in dynamical systems: foundations of harmony theory. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 194–281. MIT Press, Cambridge (1986)
15. Sodemann, A.A., Ross, M.P., Borghetti, B.J.: A review of anomaly detection in automated surveillance. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **42**(6), 1257–1272 (2012)
16. Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N.: Learning deep representations of appearance and motion for anomalous event detection. In: BMVC (2015)
17. Zhang, Y., Lu, H., Zhang, L., Ruan, X.: Combining motion and appearance cues for anomaly detection. Pattern Recogn. **51**, 443–452 (2016)
18. Zhao, B., Fei-Fei, L., Xing, E.P.: Online detection of unusual events in videos via dynamic sparse coding. In: CVPR, Washington, DC, USA, pp. 3313–3320 (2011)