A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework

Weixin Luo*
ShanghaiTech University
luowx@shanghaitech.edu.cn

Wen Liu*
ShanghaiTech University
liuwen@shanghaitech.edu.cn

Shenghua Gao†
ShanghaiTech University
gaoshh@shanghaitech.edu.cn

Abstract

Motivated by the capability of sparse coding based anomaly detection, we propose a Temporally-coherent Sparse Coding (TSC) where we enforce similar neighbouring frames be encoded with similar reconstruction coefficients. Then we map the TSC with a special type of stacked Recurrent Neural Network (sRNN). By taking advantage of sRNN in learning all parameters simultaneously, the non-trivial hyper-parameter selection to TSC can be avoided, meanwhile with a shallow sRNN, the reconstruction coefficients can be inferred within a forward pass, which reduces the computational cost for learning sparse coefficients. The contributions of this paper are two-fold: i) We propose a TSC, which can be mapped to a sRNN which facilitates the parameter optimization and accelerates the anomaly prediction. ii) We build a very large dataset which is even larger than the summation of all existing dataset for anomaly detection in terms of both the volume of data and the diversity of scenes. Extensive experiments on both a toy dataset and real datasets demonstrate that our TSC based and sRNN based method consistently outperform existing methods, which validates the effectiveness of our method.

1. Introduction

Anomaly detection has been extensively studied in computer vision because of its potential applications in video surveillance, activity recognition and scene understanding, *etc*. An anomaly detection system would greatly reduce human labor and time. However, anomaly detection is still an extremely challenging task because of the unbounded property of anomaly. In real applications, on the one hand, compared with normal events, anomaly is rare and it is extremely expensive to collect abnormal events; On the other hand, it is infeasible to collect all possible abnormal events. Therefore for a typical anomaly detection dataset, only normal scenarios are given in a training set. To identify whether an abnormal event occurs, a common approach is to exploit regular patterns in terms of appearance and motion on the training set. Any pattern that does not agree with these regular ones would be classified as irregular ones.

Dictionary learning based approaches have demonstrated their success for anomaly detection [17][29]. In these approaches, learning a dictionary to encode all normal events on the training set and an abnormal event would result in a large reconstruction error. However, the optimization of sparse coefficients is extremely time consuming, which becomes the bottleneck of dictionary learning based anomaly detection approaches. Further, features govern the performance of anomaly detection, while dictionary learning based approaches are mainly based on hand-crafted features, which may not be optimal for video representation. Recently, in light of the great successes of deep learning in many computer vision tasks[15][22], it has been introduced to the anomaly detection. Specifically, an Auto-Encoder is learnt on the normal training data under an assumption that regular data can be reconstructed by themselves while irregular ones cannot [11]. However, such a solution is based on a 3D Convolutional Neural Network (ConvNet), while previous work has shown that extracting appearance and motion information separately with a two-stream network is a better solution for feature extraction in videos [7]. Further, such a solution either takes a video cube as its input, and regular/irregular frames in this cube may affect the classification of each other. To avoid this, video cubes have to be sampled by centering the cube over all frames, which is computationally expensive.

In this paper, we propose a sparse coding based approach for anomaly detection. More specifically, a dictionary is learnt to encode regular patterns in terms of appearance, and features corresponding to normal events be sparsely reconstructed by this dictionary with a small reconstruction error. Further, to improve the smoothness of prediction over neighboring frames, a temporally-coherent term is imposed. Then we arrive at a Temporally-coherent Sparse Coding (TSC) formulation. It is interesting that our TSC formu-

*The authors contributed equally and are listed in alphabetical order.
†Corresponding author.

lation can be interpreted as one special stacked Recurrent Neural Network (sRNN): the optimization of sparse coefficients to an Iterative Soft-thresholding Algorithm(ISTA) algorithm corresponds with to a stacked network, and the temporally-coherent term makes the reconstruction coefficients of current frame depend on that of previous frame. In order to directly optimize the reconstruction coefficients rather than elaborately choosing the hyper-parameters in TSC, we propose to optimize all parameters in sRNN simultaneously, which avoids the nontrivial hyper-parameter selection in TSC. In addition, sRNN is a feed-forward network that would greatly accelerate the anomaly prediction in testing phase.

It is desirable to learn an anomaly detection model which works well under multiple scenes with multiple view angles. However, almost all existing datasets are not suitable for such kind of evaluation because of the lack of scene diversity. In fact, almost all existing datasets only have contained videos captured by one fixed camera. In this paper, we build a new dataset, named ShanghaiTech Campus, for anomaly detection. Rather than deliberately designing some abnormal events in videos, we use multiple surveillance cameras with different view angles installed at different spots, to capture the real events happened in the living area of our university campus. To the best of our knowledge, it is the largest one in terms of the volume of frames, scene diversity and the change of view angles. Therefore this new dataset would greatly facilitate the anomaly detection in real scenarios.

Contributions: We summarized the contributions of our work as follow: i) We develop a TSC formulation for anomaly detection, which can be interpreted as a special sRNN. With the help of sRNN, the anomaly prediction in testing phase is greatly accelerated. ii) We build a new anomaly detection dataset, which contains more diverse scenes and pushes the study of anomaly detection towards the usage in real applications.

2. Related Work

Most previous approaches for anomaly detection are mainly comprised of two modules: i) Feature extraction; In this module, we can extract hand-crafted or learnt features on a training set. ii) Learn a model to characterize the distribution of normal events and classify the outliers of normal distribution as anomalies.

Feature extraction. Early work utilizes the low-level trajectory features to represent the regular patterns[24]. However, these methods are not robust in complex or crowded scenes. In order to solve this problem, spatial-temporal features, such as a histogram of oriented gradients (HOG)[20], the histogram of oriented flows (HOF)[6], are widely used. Based on these spatial-temporal features, Zhang *et al*. [28] model the normal patterns with a Markov random field (MRF). Adam at al. [1] fit the regular histograms of optical flow in local regions with an exponential distribution. To represent the local optical flow patterns, Kim and Grauman [14] utilize a mixture of probabilistic PCA models.

Model selection and anomaly prediction. Dictionary learning based approaches are widely used in anomaly detection[29][17][5][21]. A fundamental assumption of these methods is that any feature can be linearly represented as a linear combination of basis of a dictionary which encodes regular patterns on the training set. [29][17][5] use the reconstruction error to determine whether a frame is abnormal or not. Ren *et al*. [21] point out that reconstruction error, such as least square error, dose not take sparsity term into consideration, and in fact, it does help the anomaly detection accuracy. To avoid this, Ren *et al*. [21] propose two solutions, *i.e*. maximum coordinate (MC) and non-zero concentration(NC), to detect anomaly. However, sparse reconstruction based methods are usually time-consuming in the optimization of sparse coefficients. To solve this problem, Jia *et al*. [17] propose to discard the sparse constraint and learn multiple dictionaries to encode the patches at multiple scales, which inevitably brings additional costs in the training phase.

Deep learning based anomaly detection. Deep learning approaches have demonstrated its successes for image classification [15], object recognition [9], as well as anomaly detection [11][27]. In [11], Hasan *et al*. propose a 3D convolutional Auto-Encoder (Conv-AE) to model the regular frames, however, 3D convolution cannot characterize the spatial and temporal information very well, as shown in the activity recognition [13]. In light of the capability of convolutional neural networks (ConvNets) to learn spatial features and the strong capability of recurrent neural network (RNN) and long short term memory (LSTM) to model temporal patterns, [3] [19] make attempts to leverage a convolutional LSTMS Auto-Encoder (ConvLSTM-AE) to characterize both appearance and motion information. Although RNNs or LSTMs are powerful and effective for processing sequential data, they are actually "black box" whose internal structures are hard to be interpreted. Recently, Scott *et al*. [26] show that a special type of RNN actually enforces a sparse constraint on the features. Inspired by the work of sparse coding based anomaly detection and interpretable RNN, we propose a TSC and its sRNN counterpart for anomaly detection.

3. Our Approach

3.1. A Revisit of Sparse Coding Based Anomaly Detection

Sparse coding based anomaly detection aims to learn a dictionary to encode all normal events with small recon-

struction error [29][17]. Mathematically, denote a feature corresponding to a normal event as x_i, then it is desirable that x_i can be linearly reconstructed by a dictionary A with small reconstruction error ϵ_i, i.e., $x_i = A\alpha_i + \epsilon_i$. Under the assumption that $\epsilon_i \sim \mathcal{N}(0, \sigma^2 I)$, and $\alpha_i \sim$ Laplace$(0, 2\sigma^2/\lambda)$, we arrive at the following objective function:

$$\min_{A,x_i} \frac{1}{2}\|x_i - A\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1 \tag{1}$$

In this formulation, the first term corresponds to a reconstruction error, and it measures how well the feature can be reconstructed by the dictionary, and the second term corresponds to a sparsity term, and λ balances the sparsity and the reconstruction error. Larger λ corresponds to a even more sparse solution. To avoid trivial solutions of the problem, usually a L2 norm constraint is imposed on each column of A: $\|A(:,j)\| \leq 1$. By alternatively optimizing the dictionary and the sparse coefficients on the training set [29], the dictionary can be learnt and it encodes all normal patterns. In the testing phase, when a feature comes in, we first compute the sparse coefficients based on dictionary A. Then based on its reconstruction error, we can classify whether it belongs to normal or abnormal events.

3.2. Temporally-coherent Sparse Coding (TSC) for Anomaly Detection

One advantage of sparse coding based anomaly detection is that it learns a dictionary to encode all normal events with small reconstruction errors, and an abnormal event would be associated with a large reconstruction error. However, it does not consider the temporal coherence between neighboring frames within normal/abnormal events. However, as shown previous works [17][18], in sparse coding, similar features may be encoded as dissimilar sparse codes, i.e., the locality information is lost. To preserve the similarity between the neighboring frames, motivated by the work [29], we propose a Temporally-coherent Sparse Coding (TSC). Specifically, if two neighboring frames are similar, it is desirable that their sparse coefficients are similar too. To achieve this goal, we use the similarity between neighboring frames to weight the distance between their sparse coefficients. Specifically, we denote x_{t-1} and x_t as features corresponding the $(t-1)$-th frame and t-th frame respectively, and denote the similarity between them as $S_{t-1,t} = \exp(-\frac{\|x_t - x_{t-1}\|_2^2}{\delta^2})$, where $\delta^2 = 100$ in our experiments. It is worth mentioning that since $S_{t-1,t}$ would be multiplied by λ_2, therefore, we can set δ to any value and tune λ_2 accordingly. Then we use $S_{t-1,t}$ to weight $\|\alpha_t - \alpha_{t-1}\|_2^2$ and substitutes temporally coherent constraint into the objective function of sparse coding, we arrive at the objective function of TSC:

$$\begin{aligned} &\min_{A,\alpha_t} \sum_{t=1}^{T} \|x_t - A\alpha_t\|_2^2 + \lambda_1\|\alpha_t\|_1 + \lambda_2 S_{t,t-1}\|\alpha_t - \alpha_{t-1}\|_2^2 \\ &\text{s.t.} \quad \|A(:,i)\| \leq 1 \end{aligned} \tag{2}$$

This objective 2 is not convex. Following the classical optimization strategy in sparse coding [2][16], we can alternatively update A and α_t ($t = \{1, \ldots, T\}$).

Optimization of A. When all α_t ($t = \{1, \ldots, T\}$) are fixed, the objective function corresponding to A can be written as follows:

$$\begin{aligned} &\min_{A} \sum_{t=1}^{T} \|x_t - A\alpha_t\|_2^2 \\ &\text{s.t.} \ \|A(:,i)\| \leq 1 \end{aligned} \tag{3}$$

Then, we use a projected gradient descent algorithm to optimize A.

Optimization of α_t. When A is fixed, we arrive at the following objective function w.r.t. reconstruction coefficients of all features:

$$\min_{\alpha_t} \sum_{t=1}^{T} \|x_t - A\alpha_t\|_2^2 + \lambda_1\|\alpha_t\|_1 + \lambda_2 S_{t,t-1}\|\alpha_t - \alpha_{t-1}\|_2^2 \tag{4}$$

After that, we update α_t ($t = \{1, \ldots, T\}$) with a Sequential Iterative Soft-Thresholding Algorithm(SISTA) [26] whose main steps are algorithm 1. In this algorithm, $\text{soft}_b(x) = \max(x - b, 0) = \text{ReLU}(x - b)$, K corresponds to the steps of ISTA algorithm. γ is a hyper parameter.

Algorithm 1 Sequential iterative soft-thresholding algorithm.

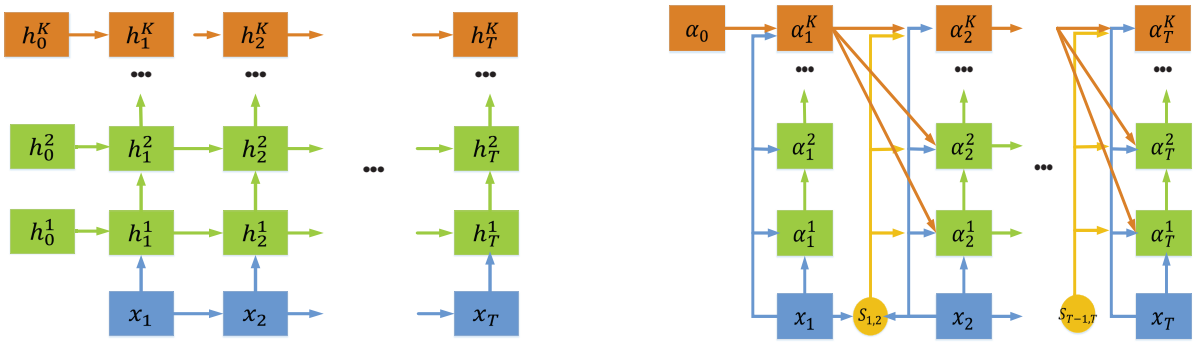
Input: extracted feature $x_{1:T}$, hyper-parameter $\lambda_1, \lambda_2, \gamma$, initial $\hat{\alpha}_0$, the steps of ISTA K

for $t = 1$ to T **do**
  $\hat{\alpha}_t^0 = \alpha_{t-1}$
  for $k = 1$ to K **do**
    $z = [I - \frac{1}{\gamma}(A^T A + S_{t-1,t}\lambda_2 I)]\hat{\alpha}_t^{k-1} + \frac{1}{\gamma}A^T x_t$
    $\hat{\alpha}_t^{(k)} = \text{soft}_{\lambda_1/\gamma}(z + \frac{S_{t-1,t}\lambda_2}{\gamma}\alpha_{t-1})$
  end for
  $\alpha_t = \hat{\alpha}_t^K$
end for
return $\alpha_{1:T}$;

3.3. Interpreting TSC with a Stacked RNN (sRNN)

A traditional RNN is based on an assumption that $h_t = f(x_t, h_{t-1})$, which introduces a recurrent structure. Many previous works [10][4] show that by stacking multiple RNNs on top of each other, the performance of classification or regression can be further boosted. We denote x_t as an input at time t and denote h_t^k as an output of hidden nodes in the k-th layer at time t. σ_b is the nonlinear activation function parameterized by b. In this paper, we choose $\sigma_b(x) = \text{soft}_b(x)$. Mathematically, the stacked RNN (sRNN) can be written as follows [26]:

$$h_t^{(k)} = \begin{cases} \sigma_b(W^{(1)} h_{t-1}^{(1)} + V x_t), & k = 1, \\ \sigma_b(W^{(k)} h_{t-1}^{(k)} + U^{(k)} h_t^{(k-1)}), & k > 1. \end{cases} \tag{5}$$



(a) Vanilla stacked RNN [26]

(b) Stacked RNN couterpart of TSC

Figure 1. The blue boxes represent the input x_t of stacked RNNs. The green and orange boxes represent coding vectors α_t^k. The yellow circles are similarities between neighboring frames.

The first layer accepts the last moment output at the same layer h_{t-1}^1 and the current moment input x_t as its inputs. Similarly, the rest of the stacked layers accept the last moment output h_{t-1}^k at the same layer and the previous layer output h_t^{k-1} at the same moment as their inputs.

By comparing the optimization procedure in Algorithm 1 with stacked RNN, we can see that Equation (2) can be interpreted with sRNN: The K steps in Sequential Iterative Soft-Thresholding Algorithm correspond to the number of layers in sRNN. Compared the proposed sRNN to classical RNN [10], the difference between them is that x_t is fed into to all sRNN layers in our sRNN, while vanilla RNN only takes x_t as its input in the first layer. Further, $S_{t,t-1}$ takes x_t and x_{t-1} as inputs, which means that h_t^k also depends on the input of last moment x_{t-1}. And $S_{t,t-1}$ is the input of each hidden state h_t^k. We illustrate the stacked RNN in our problem in Figure 1.

More specifically, the mapping from the variables in TSC to variables in sRNN in Equation (5) is:

$$
\begin{aligned}
W^{(1)} &= I - \frac{\lambda_2}{\gamma} A^T A \\
W^{(k)} &= \frac{S_{t-1,t}\lambda_2}{\gamma} I, k > 1 \\
U^{(k)} &= I - \frac{1}{\gamma}(A^T A + S_{t-1,t}\lambda_2 I), k > 1 \\
V^{(k)} &= \frac{1}{\gamma} A^T, k = 1, ..., K \\
b &= \lambda_1/\gamma \\
h_t^{(k)} &= \alpha_t^k
\end{aligned}
$$

3.4. Learning Parameters with Our sRNN

If the number of layers in stacked RNN (K) is very high, our network is identical with the TSC, which guarantees that all α_t's are sparse. Nevertheless, we also need to choose proper hyper-parameters in TSC to guarantee its good performance for anomaly detection. However, such hyper-parameter selection is nontrivial. Rather than optimizing the objective in TSC with the fixed hyper-parameters, we propose to optimize all parameters in sRNN simultaneously. Specifically, we optimize parameters in our sRNN by using an Auto-Encoder way, *i.e.*, we use the last layer output (h_t^K) of sRNN to reconstruct the input x_t with the mapping function parameterized by Z, i.e., $x_t = Zh_t^K$. We denote the parameters in sRNN as $\theta = \{A, \lambda_1, \lambda_2, Z, \alpha_0, \gamma\}$. Then we can optimize all parameters as follows:

$$
\min_{\theta} \sum_{t=1}^{T} \|x_t - Zh_t^K\|_F^2 + \beta\|\theta\|_F^2 \tag{6}
$$

To solve the Equation (6), we use a min-batch based Stochastic Gradient Descent (SGD) algorithm. Specifically, we use the RMSPROP [23] based SGD method, and the weight for weight decay term $\beta = 0.005$. Further, a larger K will inevitably introduces more computational cost. Therefore, rather than using a very large K, we use a small one (K=3). As shown in experiments section, such a shallow architecture achieves much better performance than all existing methods. Our sRNN has two advantages: i) we can learn all parameters in sRNN rather than choosing the hyper-parameters in TSC; ii) the architecture of our sRNN is not that deep. In the testing phase, we can get $\alpha_t = h_t^K$ in one forward pass, which greatly accelerates anomaly detection.

3.5. Multiple Patches Sampled at Multiple Scales

Sampling multiple patches at multiple scales has been shown to be a very effective way for improving the anomaly detection [17]. We also use the same strategy. Specifically, in our work, we use the spatial ConvNet pretrained on the UCF101 dataset to extract spatial features for each frame,

and the size of output feature map is $7\times7\times2048$ *. Then we gradually partition the feature map into increasingly finer regions: 1×1, 2×2, and 4×4. We use the max pooling over each subregion. So the feature dimension of all subregions are the same, i.e., 2048. Rather than learning multiple dictionaries for features at different scales [17], which brings additional computational cost, features at all scales share the same dictionary in our method. For the features at multiple scales, we only enforce a temporal coherent constraint for features at the same scale and same spatial location.

3.6. Anomaly Detection on Testing Set

In training phase, we can learn the dictionary A which well encodes the normal events. In testing phase, we feed the feature of each patch corresponding to time t into our special sRNN, and with one forward pass, we can get the α_t. So we can calculate the reconstruction error corresponding to patch x_t: $l(t) = \|x_t - A\alpha_t\|_2^2$. Then, we pick the maximum reconstruction error among all patches within this frame as the frame level reconstruction error. Following the work [11][3], after calculating all frame level reconstruction errors for all testing videos, we normalize the errors to range $[0, 1]$, and calculate regularity score for each frame based on the following equation:

$$s(t) = 1 - \frac{l(t) - \min_t l(t)}{\max_t l(t) - \min_t l(t)} \tag{7}$$

Smaller $s(t)$ means the t-th frame is more likely corresponding to an abnormal event.

4. ShanghaiTech Campus Dataset

It is desirable that the anomaly detection model trained can be directly applied in multiple scenes with multiple view angles. However, almost all existing datasets only contain videos captured with one fixed angle camera, and it lacks diversity of scenes and view angles. To increase scene diversity, we build a new anomaly detection dataset. To the best of our knowledge, it is the biggest one for anomaly detection, and it is even bigger than the summarization of all existing datasets in terms of the volume of data and the diversity of scenes. Further we introduce the anomalies caused by sudden motion in this dataset, such as chasing and brawling in our dataset, which are not included in existing datasets. These characteristics make our dataset more suitable in real scenarios. To better understand the differences between our dataset and existing anomaly detection datasets, we briefly summarize all anomaly detection datasets as follows:

- CUHK Avenue [17] dataset contains 16 training videos and 21 testing videos with a total of 47 abnormal events, including throwing objects, loitering and running. The size of people may change because of the camera position and angle.

- UCSD Pedestrian 1 (Ped1) [18] dataset includes 34 training videos and 36 testing videos with 40 irregular events. All of these abnormal cases are about vehicles such as bicycles and cars. Pedestrian 2 (Ped2) [18] dataset contains 16 training videos and 12 testing videos with 12 abnormal events. The definition of anomaly for Ped2 is the same with Ped1.

- Subway [1] dataset are 2 hours long in total. There are two categories, i.e. Entrance and Exit. Unusual events contain walking in wrong directions and loitering. More importantly, this dataset is recorded in indoor environment while above ones are recorded in outdoor environment.

- Our new proposed dataset has 13 scenes with complex light conditions and camera angles. It contains 130 abnormal events and over 270, 000 training frames. Moreover, pixel level ground truth of abnormal events is also annotated in our dataset.

We show some samples of our dataset to Figure 2 and some statistics of different datasets to Table 1.

5. Experiments

In this paper, we first empirically evaluate our proposed method under a controlled setting on a synthesized dataset, then we compare our method with other state-of-the-art methods on real anomaly detection datasets. Different parameters in TSC and sRNN are also empirically evaluated. All codes and dataset will be released *.

5.1. Experimental Setup

Parameters. In our experiments, for real anomaly detection dataset, the dimensionality of feature is 2048, and we fix the size of dictionary A to 2048×2048. For TSC, we set λ_1, λ_2, γ to 0.2, 2.0, and 1, respectively. For sRNN, we set $K = 3$. We set the length of each video clip T to 10 frames.

Measurements. We can predict whether abnormal event occurs based on $s(t)$. One can set a threshold and if the score of the frame is smaller than the threshold, the frame can be categorized to an abnormal case. Obviously a higher threshold may cause a higher false negative ratio, while a lower one may lead to more false alarms. By changing the

*Similar to the work in Conv-AE [11], we also find that the motion feature, such as optical flow or CNN feature extracted from optical flow [12] does not help the anomaly prediction.

*https://github.com/StevenLiuWen/sRNN_TSC_Anomaly_Detection

Figure 2. Some samples from our new proposed dataset and other datasets. The first row represents some samples from the UCSD Ped1, UCSD Ped2, CUHK Avenue and Subway Entrance and Subway Exit datasets, respectively. The second row represents normal scenes from our proposed dataset (ShanghaiTech Campus). The third row stands for the related abnormal cases where green dots are abnormal regions.

Table 1. Comparision of our dataset with other released datasets.

Dataset	#Frames					#Abnormal Events	#Scenes
	Total	Training	Testing	Regularity	Irregularity		
Our Dataset	**317,398**	**274,515**	**42,883**	**300,308**	**17,090**	**130**	**13**
CUHK Avenue	30,652	15,328	15,324	26,832	3,820	47	1
UCSD Ped2	4,560	2,550	2,010	2,924	1,636	12	1
UCSD Ped1	14,000	6,800	7,200	9,995	4,005	40	1
Subway Entrance	136,524	20,000	116,524	134,124	2,400	66	1
Subway Exit	72,401	7,500	64,901	71,681	720	19	1

threshold gradually, we can arrive at a ROC curve. The Area Under Curve(AUC) is a commonly used measurement for detecting irregularity [18]. In this paper, we use frame-level AUC to evaluate the performance of different methods.

5.2. Evaluate with a Synthesized Dataset

To evaluate the performance of our method for the anomaly caused by a sudden change of appearance, we deploy experiments on a synthesized Moving-MNIST dataset. Specifically, we randomly choose two digits from the MNIST dataset, and put them in the center of a black image whose size is 225×225 pixels. Then in the next 19 frames, the digits randomly moving horizontally or vertically. In this way, we can get a sequence with 20 frames. In our experiments, we synthesize 10,000 sequences for training data and train the network. For each testing sequence, 5 consecutive frames are randomly occluded by randomly inserting

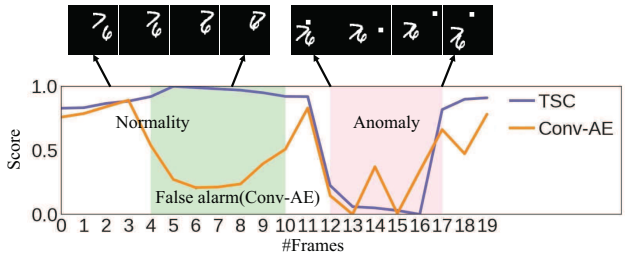


Figure 3. A sample on the Moving-MNIST dataset.

a 3×3 white box. We generate 3,000 sequences in total as testing data. Then we use the intensity of the images as features and normalized it with L2 normalization.

Our model achieves 86.52% and Conv-AE achieves 74.3%, which illustrates that our method significantly outperforms Conv-AE in terms of AUC. We also show the curve of $s(t)$ w.r.t. different events in Figure 3. We can see that our method achieves a more smooth prediction.

5.3. Evaluation with Real Anomaly Datasets

We also evaluate both TSC based and sRNN based methods on real datasets. Specifically, we conduct experiments on our own dataset as well as two recently most used datasets, including CUHK Avenue [17] and UCSD Ped2 [18] *.

We list the performance of different methods on these datasets in Table 2. It clearly shows that both our methods outperform all existing methods, including ConvAE [11] and Del *et al*. [8] which are recently proposed methods which achieve state-of-the-art performance for anomaly detection. Specifically, since our dataset contains multiple scenes, which makes our dataset more realistic and challenging. On this dataset, our TSC based method outperforms Conv-AE by about 6% in terms of AUC. On the Ped2 dataset, the improvement of our TSC outperforms Conv-AE by more than 10%. The improvement is obvious. Further, by comparing TSC and sRNN, we can see that performance of sRNN is even better than that of TSC. However, for sRNN, we don't need to elaborately choose hyper-parameters, and the testing phase is more sufficient than TSC.

Table 2. AUC of different methods on the Avenue, Ped2 and our dataset (ShanghaiTech Campus).

	Avenue	Ped2	Our dataset
MPPCA [18]	N/A	69.3%	N/A
MPPC+SFA [18]	N/A	61.3%	N/A
HOFME [25]	N/A	87.5%	N/A
Conv-AE [11]	74.5%	81.1%	60.85%
Del *et al.* [8]	78.3%	N/A	N/A
TSC	80.56%	91.03%	67.94%
sRNN	**81.71%**	**92.21%**	**68.00%**

We also compare our sRNN with LSTM based Auto-Encoder where the same features are used, and ConvLSTM based Auto-Encoder which extracts features with ConvLSTM from raw pixels. The AUC of these methods on Avenue and Ped2 is listed in Table 3: We can see that our sRNN also outperforms these baselines.

Table 3. AUC of two baselines on the Avenue and Ped2.

	Avenue	Ped2
LSTM-AE	75.33%	83.62%
ConvLSTM-AE	77.00%	88.10%
sRNN	**81.71%**	**92.21%**

Finally, we show the change of score ($s(t)$), similarities between neighboring frames ($S_{t,t-1}$), distances between sparse codes of neighboring frames ($\|\alpha_t - \alpha_{t-1}\|$) for some normal and abnormal events on the Ped2 dataset in Figure 3. We can see that a smooth similarity and distance can be found for the frames within the normal or abnormal events, which agrees with the motivation of our TSC.

5.4. The Effect of Different Hyper-Parameters

Weight of Sparsity Term(λ_1). λ_1 in Equation (2) controls the sparsity of α_t. As shown in Algorithm 1, α_t^k is optimized based on a soft-thresholding operator. The bigger λ_1 is, the more sparse α_t will be. We fix λ_2 and dictionary size to 2.0 and 2048×2048, respectively, and change λ_1 to observe how this parameter affect AUC on ped2 and Avenue. As shown in figure 5(a), bigger λ_1 does not always improve the AUC for both datasets.

Weight of Temporally-coherent Term (λ_2). λ_2 in Equation (2) controls the smoothness of the sparse codes between neighboring frames. Figure 5(b) demonstrates that a small λ_2 would harm the AUC for both the Avenue and Ped2 datasets.

Dictionary Size. We show the change of the TSC performance w.r.t. the change of dictionary size on the Avenue dataset in Figure 5(c). We can see that a larger dictionary can not always improve AUC and the optimal dictionary size varies from different datasets.

Number of Layers in Stacked RNN. The optimization of SISTA algorithm requires a very large K to achieve a sparse solution with a small reconstruction error. Fewer iterative steps may harm the optimization of TSC. Larger K means a deeper sRNN is needed for its TSC counterpart. However, a very deep sRNN may lead to gradient vanishing or explosion, which is harder to optimize. To validate how K affect the performance of our sRNN, we set it to 3 and 30, respectively. The sparsity (percentage of zero entries) of 3 layers based sRNN and that of 30 layers based sRNN is 80.0% and 92.7%, respectively, while the AUC for 3 layers based sRNN and 30 layers based sRNN is 91.03%

*The reason for not using the subway dataset because different ground truth is annotated in different work [17][11], and different ground truth favors the performance evaluation of different methods. As for UCSD pedestrain datasets, Ped1 is more frequently used for pixel-wise anomaly detection [27] and our work focuses on frame-level prediction, so we only conduct experiments on Ped2.

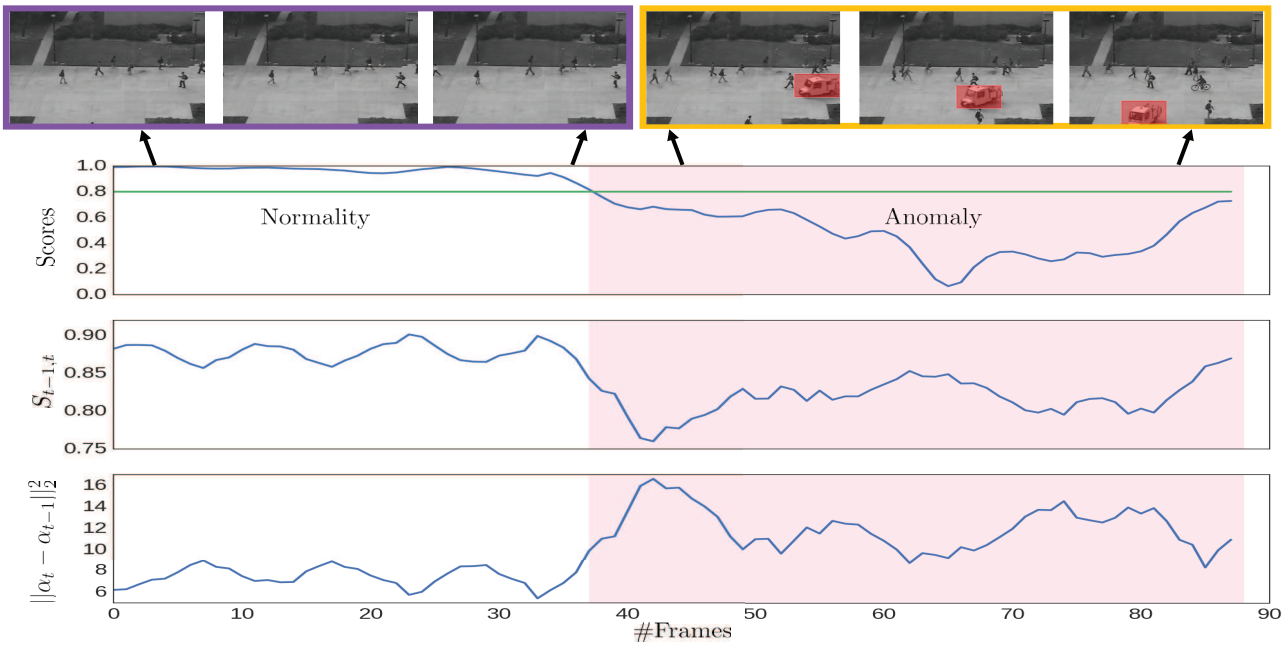
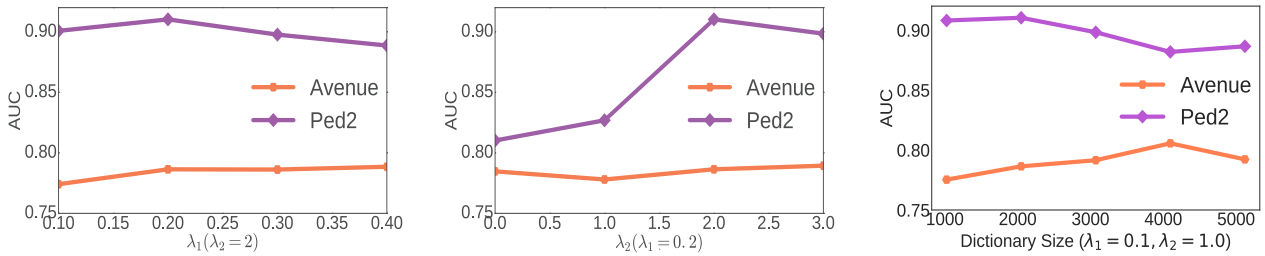


Figure 4. Scores, similarities and distances between neighboring frames for a video sample on Ped2. We can see that the similarities between neighboring frames can be kept for normal events. We highlight the abnormal events with red boxes. (Best viewed in color)



(a) AUC versus λ_1 (b) AUC versus λ_2 (c) AUC versus *dictionary size*

Figure 5. The change of AUC w.r.t. λ_1, λ_2 and the size of dictionary. (Best viewed in color)

and 88.10%, respectively. This experiment shows that sparsity does not necessarily lead better performance in sRNN. In our experiments, we set $K = 3$ for all datasets. Such a shallow architecture also accelerates the inference of $\alpha_t(h_t)$ in testing phase.

5.5. Running Time

Our sRNN model takes about one hour to train on Avenue for 10,000 steps.It takes about 0.02s for anomaly detection of a frame. While the prediction of a frame for a TSC based method is about 10 times slower than that of sRNN if we set $K = 30$ in SISTA algorithm. Therefore, our sRNN is an effective and efficient solution for anomaly detection.

6. Conclusion

In this paper, we propose a TCS framework for anomaly detection which preserves the similarities between the frames within the normal/abnormal events. Our TCS can be interpreted with a special sRNN. By optimizing all parameters in sRNN simultaneously, we can avoid the nontrivial parameter selection, and reduce the computational cost for inferring the reconstruction coefficients in testing phase. Considering the fact that most anomaly detection datasets only contain one scene with the same view angle, we build a datasets which is the most challenging one in terms of data volume and scene diversity. Extensive experiments on both a synthesized dataset and real datasets validate the effectiveness of our TCS and sRNN methods.

Acknowledgements. This work was supported by NSFC (No. 61502304).

References

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *TPAMI*, 30(3):555–560, 2008.

[2] M. Aharon, M. Elad, and A. Bruckstein. rmk-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[3] Y. S. Chong and Y. H. Tay. Abnormal event detection in videos using spatiotemporal autoencoder. *arXiv preprint arXiv:1701.01546*, 2017.

[4] J. Chung, C. Gülçehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. In *ICML*, pages 2067–2075, 2015.

[5] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011.

[6] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *European conference on computer vision*, pages 428–441. Springer, 2006.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. *arXiv preprint arXiv:1604.06573*, 2016.

[8] A. D. Giorno, J. A. Bagnell, and M. Hebert. A discriminative framework for anomaly detection in large videos. In *ECCV*, pages 334–349. Springer, 2016.

[9] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[10] A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[11] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *CVPR*, 2016.

[12] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[13] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*.

[14] J. Kim and K. Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2921–2928. IEEE, 2009.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[16] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801, 2007.

[17] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2720–2727, 2013.

[18] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, volume 249, page 250, 2010.

[19] J. R. Medel and A. Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[20] N. Navneet and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[21] H. Ren, H. Pan, S. I. Olsen, and T. B. Moeslund. A comprehensive study of sparse codes on abnormality detection. *arXiv preprint arXiv:1603.04026*, 2016.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[23] T. Tieleman and G. E. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning*, 2012.

[24] F. Tung, J. S. Zelek, and D. A. Clausi. Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. *Image and Vision Computing*, 29(4):230–240, 2011.

[25] T. Wang and H. Snoussi. Histograms of optical flow orientation for abnormal events detection. In *Performance Evaluation of Tracking and Surveillance (PETS), 2013 IEEE International Workshop on*, pages 45–52. IEEE, 2013.

[26] S. Wisdom, T. Powers, J. Pitton, and L. Atlas. Interpretable recurrent neural networks using sequential sparse recovery. *arXiv preprint arXiv:1611.07252*, 2016.

[27] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

[28] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *CVPR*, volume 1, pages 611–618. IEEE, 2005.

[29] B. Zhao, F. Li, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR*, pages 3313–3320. IEEE, 2011.