Learning Spatio-Temporal Features for Detecting Anomalies in Videos using Convolutional Autoencoder

Manassés Ribeiro[1,2], Marcelo Romero [2], André E. Lazzaretti[2], Heitor S. Lopes[2]
[1]Catarinense Federal Institute of Education, Science and Technology, Videira, Brazil
[2] Federal University of Technology Paraná, Curitiba, Brazil
Emails: manasses.ribeiro@ifc.edu.br, nmarceloromero@gmail.com, {lazzaretti, hslopes}@utfpr.edu.br

***Abstract*—Automatic video surveillance systems are a recurrent topic in recent video analysis research. Anomaly detection is an interesting way for tackling this problem, because video analysis is tedious and exhaustive for humans. Depending on the application field, anomalies can present different characteristics and challenges for pattern representation, requiring the design of hand-crafted features (such as spatial and temporal information). Deep learning methods have achieved the state-of-the-art performance for many recognition problems in recent years and may be an interesting choice for learning features automatically, since it captures the 2D structure in image sequences during the learning process. The deep Convolutional Autoencoder (CAE) may be an interesting approach for anomaly detection since they can learn signatures automatically in an unsupervised way. This work purposes the use of a deep CAE in the anomaly detection context for learning spatio-temporal signatures from raw video frames. Similar our previous work, we use as anomaly score a successful strategy based on the reconstruction error of a package of frames. The proposed methods were evaluated by means of several experiments with public-domain datasets. The promising results support further research in this area.**

***Keywords*—Anomaly detection; One-class classification problems; Convolutional autoencoders; Deep learning methods**

I. Introduction

The task of video analysis is tedious and exhaustive for humans, and this process can be worsened by the inherent human limitations, such as distraction and tiredness. Despite human abilities to analyze and classify images and videos, those limitations may lead human observers to fail at detecting important unexpected events (i.e. anomalies). Hence, this problem has pushed the development of automatic video surveillance systems, since the necessary human endeavor for effectively performing the observation task is too hard.

Nowadays, automatic video surveillance is a topic of great importance that has been intensely studied in the recent literature [1], and automatically detecting anomalies in videos has been a subject of great interest for both academic and industry areas [2]. Indeed, detecting objects from previously unknown classes is an issue treated in different pattern recognition problems, especially because new classes, anomalous behaviors and concept drifts can frequently occur in real-world applications [3]. Notice that, in this work, we consider anomaly detection, novelty detection and One-Class Classification (OCC) as synonyms, since there is still no universally accepted definition for each these terms [4] and they are often treated as similar. In this text, we will use preferably the term "anomaly detection".

The anomaly detection problem has been tackled through different strategies throughout the years. Recently, several approaches have focused on pattern representation. The most common pattern representation methods are based on hand-crafted features, such as those based on optical flow, histogram methods, or their combination. On the other hand, Deep Learning (DL) methods, such as Convolutional Neural Networks (CNN) and Convolutional Autoencoder (CAE), have been also studied for feature learning. CNN is a supervised approach considered as the state-of-the-art for image and video classification problems [5], [6]. A CAE is an unsupervised approach based on Autoencoders (AEs) capable of capturing the 2D structure of image and video sequences [7]. Since training CAE does not require label information of input data, it can be useful to model anomaly detection problems.

DL approaches are useful for learning relevant signatures (features) capable of representing patterns to aid at retrieving information within large-volume of video data [8]. For this reason, it can be an alternative approach to tackle video anomaly detection problems [4].

In fact, the use of CAEs for feature learning in video is still under-explored in the recent literature (see, for instance, [9] and [10]). In these works, AEs are, in general, used for mixing hand-crafted features (extracted from frames or patches), and a classifier is used to discriminate anomalies. In our previous work [10] we proposed the use of a CAE for combining *appearance* and *motion* features extracted from video frames using the Canny edge detector and optical flow, respectively. The reconstruction error of the CAE was used as the classification score.

This method was successful and, in the current work, we propose a step further: an approach based on using packages of raw frames organized in time-slices instead of only *appearance* and *motion* features extracted from video frames. Hence, we aim at using a CAE to learn spatio-temporal signatures (features) from raw frames instead of using hand-crafted features that may not provide enough information for detecting anomalies. Notice that, compared to our previous work, we maintain our strategy based on using reconstruction errors to classify anomalies in videos. The working hypothesis is that a CAE is capable of learning relevant spatio-temporal signatures from normal events in videos, and that the reconstruction error of a package with video frames can be used for providing an anomaly score that can be used to discriminate normal from abnormal events in videos.

Accordingly, the problem addressed in this work consists in modelling a normal concept that can be used for discriminating spatio-temporal anomalies based on a set of videos containing events considered normal.

The remaining of this work is organized as follows. Section II presents the theoretical background and some related work. Section III describes the proposed method. Section IV presents the computational experiments, their results and a brief discussion. Finally, Section V presents the conclusions drawn from the development of this work, and future research directions are pointed out.

II. Theoretical Background and Related Work

A. Anomaly Detection

In the literature, anomaly detection has been proposed for a variety of reasons, including, for instance, detection of malicious activities (frauds [11], for instance), in bioinformatics [12] and weather forecast (catastrophe detection) [13]. Regarding the area of visual surveillance in videos, anomaly detection has attracted research due to the growing concern with safety in both public and private places [10], [9].

OCC is an important concept to approach problems of automatic video surveillance in crowded scenes, that is, environments with a large amount of objects and people moving. In general, an OCC (or novelty detector) can be defined as a classifier based on previously known patterns, which are arranged as one (or a set of) normal concept, allowing the identification of patterns that were not present in the original training dataset, normally defined as novelties [14]. Usually, to tacke the anomaly detection problem as an OCC problem, the normal class should be human-defined and it contains a large number of examples. On the other hand, the "abnormal" class comprises samples that are quite different of the normal samples or that are rarely present in the normal class.

Figure 1 presents an example of abnormal events in a hypothetical two-dimensional dataset. In this example, the data has two normal regions, N_1 and N_2, in which most the samples are present. Other samples that are distant from these regions, such as the objects o_1,o_2, and o_3, are considered anomalies.

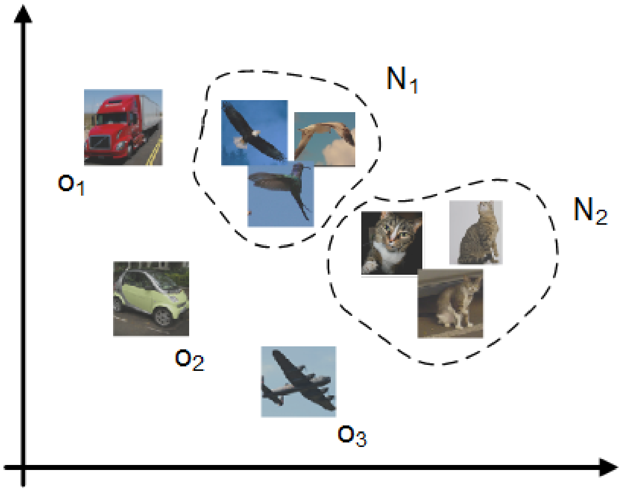

Fig. 1. Example of anomalies in a bi-dimensional didactic dataset.

There are three main approaches to use OCC for anomaly detection [14]:

1) Density methods: comprise models that estimate the probability density function of the input patterns;
2) Reconstruction methods: they use clustering to find out if a given input pattern is an anomaly or not, based on the distance from the unknown input to clusters previously defined in the training process;
3) Boundary methods: comprise methods that impose boundaries upon the training dataset, assuming an unknown distribution.

B. Pattern Representation

A key issue for anomaly detection methods is how a pattern is represented. Its choice is crucial and it may significantly affect the classification performance. The acquisition of relevant features from the raw data (i.e. images or videos) is important to enable a good classification performance for different types of anomalies. In an attempt to classify pattern representation approaches, Hu et al. [15] suggested that the methods for representing patterns can be grouped into four main categories: trajectory-based, spatio-temporal-interest-point-based, foreground-blob-based and volume-based. However, methods that can be grouped into one or more categories are frequently observed in the literature and, usually, they have in common the use of hand-crafted features.

Therefore, in the work [16], the author have organised the pattern representation methods for anomaly detection into three groups: statistical-based, spatial-temporal-based and those based on DL. For statistical-based approaches (i.e. methods based on statistical models postulated over the normal context, where the abnormal events are classified based on probabilities), there are, for instance: Gaussian mixture model [17] and hidden Markov model [18]. For pattern representation based on spatio-temporal, there are techniques such as histogram of oriented gradients [19], histogram of optical flow [20], textures of optical flow [21], tracking-based [22], and spatio-temporal texture [23]. In general, such features are based on standard computer vision methods and other variants, and spatial-temporal-based methods are the most common approach for pattern representation.

This work focus on pattern representation based on DL methods. DL are specially suited for learning data representations. They comprise both linear and non-linear transformations aiming at producing abstract, but useful representations [24]. DL algorithms are based on distributed representations and the idea is that the final information is generated from exhaustive iterations considering many units. By combining different layer compositions, it is possible to obtain different levels of abstraction [24].

DL methods have been intensively used for computer vision problems and, in special, for visual recognition tasks. Several related work can be found in recent years, and they are categorized according to the basic method used, that is: CNN [6], AE [25], and restricted Boltzmann machines [26]. As mentioned before, in the context of anomaly detection, DL methods are still in early stages of development.

C. Feature Learning with DL Methods

Focusing on the feature learning, recently, DL methods have achieved the state-of-the-art performance for computer vision problems [6]. A possible reason for such a high performance is that they can learn the feature extractor and the classifier simultaneously. Such characteristic can improve the inter-class separation, since both, classifier and feature extractor, are optimized to increase the overall accuracy. Thus, if there is a large amount of samples available for training, DL methods are capable of achieving superior discriminatory power for image representation when compared to hand-crafted image descriptor-based methods [27].

However, CNNs are suited for supervised classification problems and they are not directly applicable to deal with anomaly detection tasks, where only the normal class is previously known. To overcome this issue, AEs are an alternative for feature learning in OCC problems, since it can be trained using only the known class. Moreover, both the Reconstruction Error (RE) and the bottleneck (latent representation), can be used to provide classification scores. The AE model was proposed by Bourlard and Kamp [28] and, later, popularised by Vicent et al. [29] and by Krizhevsky et al. [30]. AEs were initially used in the image retrieval context but, very recently, their application for video anomaly detection has gained strength [16], [9].

The AE, that has been studied for decades (see [25], [9]), is a fully connected one-hidden-layer neural network devised to learn from unlabeled data. The idea is that the AE is trained to reconstruct the input pattern at the output of the network. Internally, an AE has a hidden layer $\mathbf{h}$ that compresses the input data to represent it in a latent representation space. The latent representation aims at exploiting closeness of input patterns, where a large number of inputs can be aggregated in a model to represent an underlying concept. Latent representation is useful to reduce the dimensionality and making it easier to understand the data [31].

However, AEs are not capable of capturing the 2D structure in image and video sequences, because the input data is a 1D vector. Such characteristic results in redundancy in the parameters of the network and removes the local information that can be extracted from the images, which is particularly relevant in the anomaly detection context, as anomalies are locally positioned in the scene. To cope with this issue, the CAE architecture was proposed by Masci et al. [7]. CAEs are similar to the ordinary AE, but the difference between them is the fact that in the CAE the weights are shared among all locations in the input, preserving the spatial locality, similar to a CNN [30]. The CAE is optimized using the cost function presented in Equation 1:

$$e(\mathbf{x}, \mathbf{y}, \Theta, \mathbf{W}) = \frac{1}{2N} \sum_{i} \left\| f\left(\mathbf{x}_i, \Theta\right) - \mathbf{y}_i \right\|_2^2 + \lambda \left\| \mathbf{W} \right\|_2^2, \quad (1)$$

where λ is the regularization parameter for the term $\|\mathbf{W}\|_2^2$, normally used during the training procedure of the CAE. CAE architecture contains convolutional and pooling layers, and it adds deconvolutional and unpooling layers for the decoding part of the architecture.

The convolutional layer abstracts the information of a filter into a scalar value parameterising the number of maps, the size of the maps and kernels' size. It connects multiple input activations within the fixed receptive field of a filter to a single activation output in the feature map. For the input $\mathbf{x}$, the hidden layer mapping (latent representation) of the $k-th$ feature map is given by Equation 2:

$$\mathbf{h}_k = \sigma\left(\mathbf{x} * \mathbf{W}_k + b_k\right), \quad (2)$$

where a single bias b is used for the whole map, σ is an activation function (for instance, the hyperbolic tangent), and the $*$ symbol corresponds to the 2D-convolution. Since one bias per pixel would take too many degrees of freedom and it is desired that each filter becomes specialized on features of the entire input, a single bias per latent map is used. The reconstruction is obtained using Equation 3:

$$\mathbf{y} = \sigma\left(\sum_{k \in H} \mathbf{h}_k * \tilde{\mathbf{W}}_k + c\right), \quad (3)$$

where c is a single bias per input channel, and H identifies the group of latent feature maps. $\tilde{\mathbf{W}}$ corresponds to the flip operation over both dimensions of the weights $\mathbf{W}$. In practice, the flip operation organizes the weights $\mathbf{W}$ in reverse order. The 2D convolution can be a *full convolution* or a *valid convolution*. In the first case, the convolutional of a $m \times n$ matrix with a $x \times n$ matrix will result in a $(m+n-1) \times (m+n-1)$ matrix, whilst in valid convolution, the resulting matrix will be $(m-n+1) \times (m-n+1)$.

The deconvolutional layer performs an inverse operation of the convolution layer with deconvolutions. The filters learned in the deconvolutional layers serve as the base to reconstruct the shape of the input, taking into account the required reshape of the output [9]. Convolutional and deconvolutional layers can be stacked to build deep architectures for CAEs. Figure 2 (top) shows an example of a CAE composed of convolutional and pooling layers in the left side (encoder), and deconvolutional and unpooling layers in the right side (decoder).

Pooling layers were originally intended for fully-supervised feed-forward architectures and it down-samples the latent representation by a constant factor. The idea of the pooling layer is to obtain translation-invariant representations, allowing more complex representations, when combined with convolutional layers. On the other hand, unpooling layers perform the reverse operation of pooling and it reconstructs the original size of each rectangular sub-region. Figure 2 (bottom) shows examples of pooling and unpooling layers.

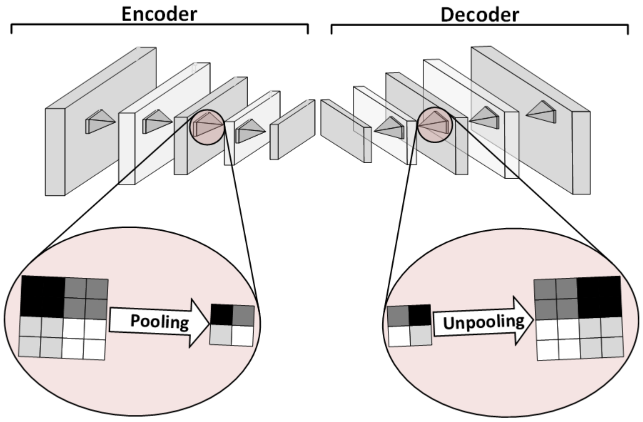

Fig. 2. Example of convolutional, pooling, deconvolutional and unpooling layers. (top) shows an example of a CAE composed of convolutional and pooling layers (encoder side), whilst in the decoder side there are deconvolutional and unpooling layers. (bottom) shows examples of both pooling and unpooling layers.

Finally, the backpropagation algorithm, introduced in [32], is used for training a CAE. Backpropagation is based on a *chain rule* for computing the derivatives. The idea behind the chain rule is to allow the information of the cost function to flow backwards through the network in order to compute the gradient and ultimately update its weights.

III. Proposed Method

In this work, the use of deep CAE for feature learning is used to allow the identification of different types of anomalies. Differently from traditional methods, the feature learning is integrated to the classification process using a CAE. In order to circumvent the limitation of hand-crafted descriptors used in previous works [10], [33], which normally requires that some *a priori* knowledge is incorporated, an approach for automatically learning spatio-temporal signatures (features) is proposed.

For this purpose, raw frames are used as input to the CAE (in time-slices) in order to investigate whether the CAE is capable of automatically learning signatures in videos using only raw input frames. Next, the CAE's Reconstruction Error (RE) is proposed as the "anomaly" score in order to discriminate between normal and abnormal events in a video.

The objective is to obtain a model capable of learning, by itself, spatio-temporal signatures, instead of using hand-crafted feature extractors. The working hypothesis is that the CAE can learn changes among video frames, allowing to capture relevant spatio-temporal signatures. Therefore, low REs is expected for known patterns, whilst high REs are expected for anomalies. Then, the spatio-temporal features learned by a trained CAE may be used to classify anomalies.

The proposed method for learning spatio-temporal features has four main steps. First, a data preparation step is performed. Input data is organised in cuboids for the training step. Cuboids are 3D-dimensional structures where frames representing the spatial dimension are packaged sequentially, thus joining the temporal with the spatial dimensions. A CAE is then trained with examples of the normal class. The model is optimized using the RE between the input cuboid and its reconstructed output. Once the training phase is complete, both normal events and anomalies are classified using the Normalized Reconstruction Error (NRE) (see Subsection III-C). Finally, the the classification performance is measured. Figure 3 presents an overview of the proposed approach.

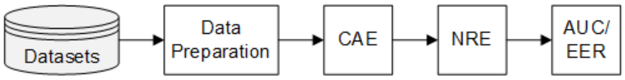

Fig. 3. Overview of the approach for learning spatio-temporal signatures.

Regarding data preparation process, the frames are extracted and sub-sampled from video clips using a fixed size window that slides along the video clip, thus reducing the observed region. After, the frames are grouped into cuboids $\mathbf{X}$ with three frames, where the observed frame f is in the center of cuboid composed of the frames within the sliding window.

Since the variability between two subsequent frames is quite small, frames are separated by a gap, so as to increase the variability. Thus, the cuboid for a certain window does not contain all of its frames. Some frames between the initial and the central frame are discarded. The same occurs for the frames between the central and the final frame. For instance, for sliding window of size $n = 5$, one frame between the initial and the central frame is skipped, and another one between the central and the final frame, i.e. the cuboid is composed only of the first, third and fifth frames, ignoring the second and fourth frames. Consequently, the number of frames ignored from a video clip depends upon the window size. An example of the sliding window approach is shown in Figure 4.

Finally, five case studies are created by grouping data into packages with discretised frames according to the window sizes $n = 3, 5, 7, 9$, whose values were selected empirically. For all case studies, the stride is 1, which represents the amount of frames skipped by the window after every slide along the video clip. For the test dataset, cuboids are labeled considering the label of the central frame f.

A. Model Architecture

The proposed architecture is similar to the model proposed in [10]. It is composed of three convolutional layers and two pooling layers in the encoder side, with the same reversed structure in the decoder side. The CAE is trained to learn the signature of normal events, considering the optimization metric presented in Equation 1.

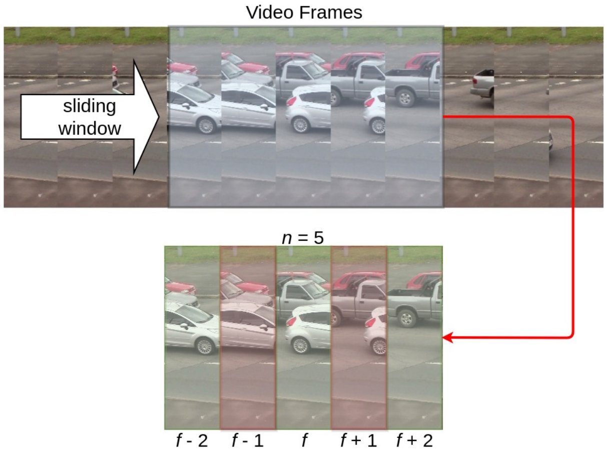

Fig. 4. Example of sliding windows approach for selecting frames to compose the cuboid X. In this example, the frames $f-2$, f and $f+2$ will be selected to compose the 3D cuboid, whilst both frames $f-1$ and $f+1$ will be discarded.

The first convolutional layer of the architecture is composed of 256 filters with stride 4. It produces 256 feature maps with resolution of 57×37 pixels that are fed to the first pooling layer, which produces 256 feature maps with resolution of 28×18 pixels. All pooling layers of the CAE are composed of 2×2 kernels, performing sub-sampling through the max-pooling operation. The second and third convolutional layers have 128 and 64 filters, respectively. The last encoder layer produces 64 features maps of 14×9 pixels. The decoder reconstructs the input by deconvolving and unpooling the input in reverse order. The output of the final layer of the CAE is the reconstructed version of the original input. Table I summarizes the details of each layer of the CAE.

TABLE I
DIMENSIONS OF EACH LAYER OF THE CAE.

Layer	Dimensions	Filter size
Input and Conv. 1	256×57×37	11×11
Pool. 1	256×28×18	2×2
Conv. 2	128×28×18	5×5
Pool. 2	128×14×9	2×2
Conv. 3 and Deconv. 1	64×14×9	3×3
Unpool. 1	128×14×9	2×2
Deconv. 2	128×28×18	5×5
Unpool. 2	256×28×18	2×2
Deconv. 3 and Output	256×57×37	11×11

The inputs of the CAE are cuboids $\mathbf{X}$ extracted from video clips. Each cuboid is a 3D-structure with three channels (see previous Section), where each input channel is a 2-D array with resolution of 235×155 pixels in gray scale.

B. Model Training

A CAE is an unsupervised learning method, which does not require class labeling of the input data. However, an indirect labelling is used in this work, since all training instances belong to videos without anomalies (that is, the normal class).

We use the backpropagation algorithm to minimize the cost function e, which is optimized using Stochastic Gradient Descent (SGD) with the adaptive sub-gradient method AdaGrad. The weights of the network are initialized using the Xavier algorithm [34]. Since the input data is a cuboid $\mathbf{X}$, the RE is evaluated over all dimensions.

C. Normalization of Reconstruction Errors

Similar to the work [10], the RE of pixels intensity value I at location (x, y) in frame t of the video sequence is computed, as shown in Equation 4:

$$RE(x,y,t) = \parallel I(\mathbf{X},t) - f(I(\mathbf{X},t)) \parallel_2^2, \quad (4)$$

where f is the model learned by the CAE and $\mathbf{X}$ is the input cuboid. Given the RE of all pixels of a frame t, the Frame Reconstruction Error (FRE) is computed by summing all the pixel-wise errors, see Equation 5:

$$FRE(t) = \sum_{(x,y)} RE(x,y,t). \quad (5)$$

After, the FRE of the frames is smoothed using a moving average filter, according to the Equation 6, where N is the number of samples (window size) of the moving average.

$$S_{FRE}(t) = \frac{1}{N} \sum_{j=0}^{N-1} FRE(t+j). \quad (6)$$

Finally, the Normalized Reconstruction Error (NRE) of a frame is computed by Equation 7:

$$NRE(t) = \frac{S_{FRE}(t) - \min(S_{FRE})}{\max(S_{FRE}) - \min(S_{FRE})}, \quad (7)$$

where the $\min(S_{FRE})$ and $\max(S_{FRE})$ are, respectively, the minimum and maximum values in the smoothed S_{FRE} found along all frames of the dataset.

D. Classification and Evaluation

In this work, the are under the ROC curve (AUC) is used to measure the classification performance. The AUC demonstrates a comparison that is independent from the threshold and provides a direct analysis of the mapping performed by the classifier. This method enables the comparison with other studies in the literature that also use the AUC, such as [9], [10]. The computation of the ROC curve is done by using the NRE, considering the True Positive Rate (TPR) and the False Positive Rate (False Positive Rate). From the AUC, it is possible to assess the EER, which indicates the point of the ROC curve where the false acceptance rate is equal to the false rejection rate, i.e., the best average performance. The lower the EER value, the higher the accuracy of the classifier.

IV. Computational Experiments and Results

This Section presents the experiments conducted to verify the hypotheses that a CAE is capable of learning relevant spatio-temporal signatures for discriminating anomalies.

Regarding feature learning, the aim is to evaluate the CAE capability to perform such task. For this purpose, a CAE was used for automatically learning spatio-temporal features from raw frames without requiring previously knowledge, and its results were compared to with those provided by the approach based on hand-crafted feature extraction proposed in our previous work [10]. Also, the RE was used for detecting anomalies.

The experiments were carried using four benchmark video datasets frequently used for anomaly detection problems: UCSD pedestrian dataset (including the two subsets — Ped1 and Ped2) [35], Avenue [36], and UMN [37]. The datasets are composed of a collection of videos with frames manually labeled by a human expert as "normal" or "abnormal".

Notice that color video datasets were converted to gray scale for this work. Moreover, frames larger than 235×155 were resized to that size using the linear interpolation method, whilst images smaller than 235×155 were kept in their original size. For both gray scale converting and image resizing, the OpenCV library[1] was used.

The CAE model proposed for this experiment was trained using a version of the Caffe [2] framework modified by Hasan et al. [9]. All experiments were run in a dedicated GPU server with an Intel i7-5820K CPU running at 3.3 GHz, with 32 GB of RAM and equipped with a Nvidia Titan-Xp GPU, running on Ubuntu.

A. Automatic Learning of Spatio-Temporal Features

The working hypothesis of this experiment is that the CAE is able to automatically learn spatio-temporal signatures and also to discriminate anomalies that were unseen during the training step. A CAE was trained using the architecture proposed in Subsection III-A. Four different combinations of the data were used, which were prepared according to the size of a sliding window (temporality) n. The case studies are 3F, 5F, 7F and 9F, defined with respect to sliding window sizes $n = 3$, $n = 5$, $n = 7$ and $n = 9$.

The experiment (FR+ED) use the original frames (FR) combined with appearance features (ED) extracted using Canny edge detector (presented in our previous work [10]). It was used as baseline in order to compare with different combinations of temporal frames (case studies). Results are summarized in Table II, showing AUC and EER. Numbers highlighted in bold are the overall best result obtained for each dataset.

Table II shows that CAE seems to be capable of learning spatio-temporal signatures (appearance and motion features), as expected. Results from this experiment are comparable to the best baseline results using FR+ED (obtained in our previous work [10]) and state-of-the-art (shown in State-of-the-art column of Table II), overcoming the result for the Ped2 dataset and achieving very close results for both Avenue and Ped1 datasets. This study suggests that, for the above-mentioned datasets, the CAE can learn relevant spatio-temporal signatures from raw input data to discriminate anomalies. We consider that it is an important contribution, since the CAE avoids the need of selecting hand-crafted feature extractors, since CAE can be trained from raw input data. Although for the UMN dataset the AUC improved as the sliding window (temporality) increased, the best overall result was worse than the baseline. This issue could be related to the CAE limitation of learning temporal information, since it was not designed to perform such a task. Since anomalies in the UMN dataset are mostly characterized by motion patterns, results suggest that a larger sliding window would be necessary to capture more relevant spatio-temporal signatures. Overall, it seems to be more difficult for the CAE to learn relevant motion patterns signatures characterizing anomalies, whilst it seems to be easier for it to learn signatures of appearance features.

Figure 5 shows the NREs obtained using Equation 7, plotted along the frames for a specific video of the Avenue dataset. The blue line shows the plot of the NRE (FR+ED) for the best overall result of the CAE using a combination of raw data with appearance features obtained in our previous work [10]. The green line shows the plot of the NRE for the best result when using spatio-temporal signatures learned by the CAE. The ground-truth, annotated by the creator of the dataset and the threshold for anomaly detection are also plotted (red and black line, respectively).

Figure 5 shows that the NRE spatio-temporal (ST) follows the NRE (FR+ED), but with less fluctuations and following a more smooth tendency along the frames. This smoothed tendency resembles a moving average effect because in this approach the frames are arranged into groups (cuboids). Moreover, a strongly correlation between ST NRE and FR+ED NRE is verified, in which the Spearman correlation was computed near to 0.910. These results suggest that, for the previously mentioned datasets, the initial hypothesis for this experiments is confirmed.

V. Conclusions

Due to the growing concern with public security worldwide, automatic video surveillance has become a topic of great interest. In turn, anomaly detection in videos can be considered a hard task, since anomalies are highly dependent on human concepts, and the volume of data has never been so big. Aiming at tackling this issue, this work presented an approach to learn relevant spatio-temporal signatures for detecting anomalies in videos using a deep CAE.

In our experiments, raw frames were used as input to the CAE using a spatio-temporal approach, to investigate whether the CAE was able to learn spatio-temporal signatures automatically. Results suggested that the proposed method can learn those features with overall results similar to the baseline of our previous work. Particularly to the UMN dataset, our results were worse than the baseline, possibly because this dataset is characterized mostly by motion features. It seems that the CAE do not work well when the dataset is composed mostly of movement. For this case, a much larger sliding window may lead to better results. It is worth to mention that

[1] https://opencv.org/

[2] http://caffe.berkeleyvision.org/

TABLE II
AUC/EER RESULTS FOR THE FOUR CASE STUDIES OF AUTOMATIC LEARNING OF SPATIO-TEMPORAL FEATURES.

Datasets	**FR+ED**		**3F**		**5F**		**7F**		**9F**		**State-of-the-art**	
	AUC	**EER**	**AUC**	**EER**	**AUC**	**EER**	**AUC**	**EER**	**AUC**	**EER**	**AUC**	**EER**
Avenue	**0.772**	0.270	0.767	0.303	0.769	0.309	0.770	0.306	0.771	0.306	0.702	0.251 [9]
UCSD Ped1	0.585	0.431	0.584	0.462	**0.585**	0.463	0.576	0.465	0.582	0.467	0.927	0.160 [38]
UCSD Ped2	0.847	0.245	0.843	0.252	0.856	0.244	0.863	0.243	**0.875**	0.233	0.908	0.170 [39]
UMN	**0.944**	0.106	0.715	0.458	0.746	0.446	0.778	0.369	0.778	0.346	0.960	– [37]

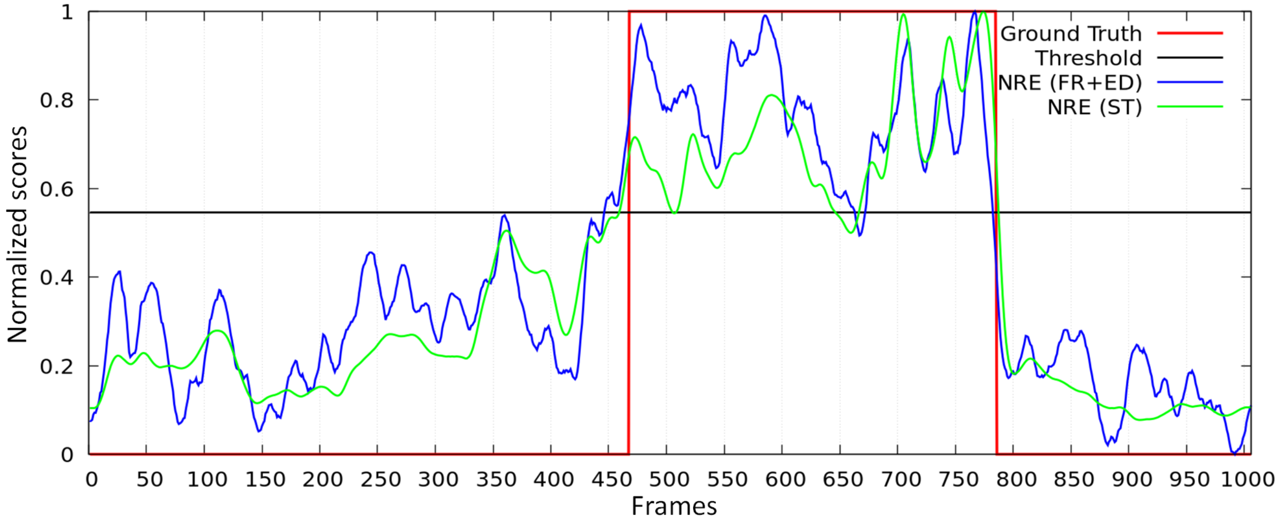

Fig. 5. Spatio-temporal NRE plotted along the frames of a Avenue dataset video clip.

the NRE had less fluctuations and followed a more smooth tendency along the frames using the spatio-temporal approach.

Results indicated that automatically learning features with a CAE can be an interesting alternative that can be extended to other datasets. However, in order to learn useful motion features, a large temporal window may be necessary. Hence, it was not yet possible to state whether a CAE is capable of learning useful motion features automatically.

Finally, results obtained so far encourage future work towards more experiments with other real-world datasets so as to test and improve the methods here proposed. Indeed, results unveiled interesting open issues to be explored in the future. In a broader sense, results achieved in this work showed that the computational approaches proposed are very promising for the research area related to anomaly detection in videos. Therefore, future work could focus on further analysing the temporal information, by using of recurrent networks, for instance, to improve the classification performance.

Acknowledgment

Author M. Ribeiro would like to thank the Catarinense Federal Institute and IFC/CAPES/Prodoutoral for the scholarship; H.S.Lopes thanks to CNPq for the research grants no. 311778/2016-0 and 423872/2016-8, and Fundação Araucária for the financial support by means of PRONEX 042/2018. All authors thanks NVIDIA for the donation of Titan-Xp GPU boards used in this work.

References

[1] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 367–386, 2015.

[2] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 6, pp. 1257–1272, 2012.

[3] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, no. Supplement C, pp. 134 – 147, 2017, online Real-Time Learning Strategies for Data Streams.

[4] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215 – 249, 2014.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2012, vol. 25, pp. 1097–1105.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2015, pp. 1–9.

[7] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks and Machine Learning*, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds., vol. I. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 52–59.

[8] X. Zhu, J. Liu, J. Wang, C. Li, and H. Lu, "Sparse representation for robust abnormality detection in crowded scenes," *Pattern Recognition*, vol. 47, no. 5, pp. 1791–1799, 2014.

[9] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2016, pp. 733–742.

[10] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognition Letters*, vol. 105, pp. 13 – 22, 2018.

[11] J. Akhilomen, "Data mining application for cyber credit-card fraud detection system," in *Conference on Advances in Data Mining. Applications and Theoretical Aspects*, ser. Lecture Notes in Computer Science, P. Perner, Ed. Heidelberg, Germany: Springer, 2013, vol. 7987, pp. 218–228.

[12] N. I. George, J. F. Bowyer, N. M. Crabtree, and C.-W. Chang, "An iterative leave-one-out approach to outlier detection in RNA-seq data," *PLoS ONE*, vol. 10, no. 6, 06 2015.

[13] M. Ohba, S. Kadokura, Y. Yoshida, D. Nohara, and Y. Toyoda, "Anomalous weather patterns in relation to heavy precipitation events in Japan during the baiu season," *Journal of Hydrometeorology*, vol. 16, no. 2, pp. 688–701, 2015.

[14] D. M. J. Tax, "One-class classification," Ph.D. dissertation, Technische Universiteit Delft, 2001.

[15] X. Hu, S. Hu, J. Xie, and S. Zheng, "Robust and efficient anomaly detection using heterogeneous representations," *Journal of Electronic Imaging*, vol. 24, no. 3, p. 033021, 2015.

[16] M. Ribeiro, "Deep learning methods for detecting anomalies in videos: theoretical and methodological contributions," Ph.D. dissertation, Graduate Program in Electrical and Computer Engineering, Federal University of Technology - Paraná, Curitiba, PR, Brazil, Mar 2018.

[17] J. Yu, "A nonlinear kernel Gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes," *Chemical Engineering Science*, vol. 68, no. 1, pp. 506 – 519, 2012.

[18] E. Dorj and E. Altangerel, "Anomaly detection approach using hidden Markov model," in *International Forum on Strategic Technology*, vol. 2. Piscataway, NJ: IEEE, 2013, pp. 141–144.

[19] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5288–5301, 2015.

[20] T. Wang and H. Snoussi, "Detection of abnormal visual events via global optical flow orientation histogram," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 6, pp. 988–998, 2014.

[21] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Textures of optical flow for real-time anomaly detection in crowds," in *International Conference on Advanced Video and Signal-Based Surveillance*. Piscataway, NJ: IEEE, Aug 2011, pp. 230–235.

[22] S. Xie and Y. Guan, "Motion instability based unsupervised online abnormal behaviors detection," *Multimedia Tools and Applications*, vol. 75, no. 12, pp. 7423–7444, 2015.

[29] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[23] J. Wang and Z. Xu, "Crowd anomaly detection for automated video surveillance," in *International Conference on Imaging for Crime Detection and Prevention*. Piscataway, NJ: IEEE, 2015.

[24] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[25] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103.

[26] R. Salakhutdinov and G. E. Hinton, "Deep Boltzmann machines," *Machine Learning Research*, vol. 5, pp. 448–455, 2009.

[27] H. A. Perlin and H. S. Lopes, "Extracting human attributes using a convolutional neural network approach," *Pattern Recognition Letters*, vol. 68, no. 2, pp. 250–259, 2015.

[28] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, no. 4, pp. 291–294, 1988.

[30] A. Krizhevsky and G. E. Hinton, "Using very deep autoencoders for content-based image retrieval," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Bruges, Belgium: i6doc.com, 2011, pp. 489–494.

[31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[33] H. A. Perlin and H. S. Lopes, "Extracting human attributes using a convolutional neural network approach," *Pattern Recognition Letters*, vol. 68, no. 2, pp. 250–259, 2015.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256.

[35] V. Mahadevan, W.-X. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2010, pp. 1975–1981.

[36] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *International Conference on Computer Vision*. Piscataway, NJ: IEEE, 2013, pp. 2720–2727.

[37] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Conference on Computer Vision and Pattern Recognition*, no. 2. Piscataway, NJ: IEEE, 2009, pp. 935–942.

[38] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2012, pp. 2112–2119.

[39] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," in *British Machine Vision Conference*, M. W. J. Xianghua Xie and G. K. L. Tam, Eds. Swansea, UK: BMVA Press, 2015, pp. 8.1–8.12.