Pattern Recognition Letters 000 (2017) 1–10

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A study of deep convolutional auto-encoders for anomaly detection in videos

Manassés Ribeiro[a,b], André Eugênio Lazzaretti[b], Heitor Silvério Lopes[b,*]

[a] Catarinense Federal Institute of Education, Science and Technology, Rod. SC 135 km 125, Videira (SC), 89560-000, Brazil
[b] Federal University of Technology – Paraná (UTFPR), Av Sete de Setembro 3165, Curitiba (PR), 80230-901, Brazil

ARTICLE INFO

Article history:
Available online xxx

Keywords:
Deep learning
Convolutional auto-encoder
Anomaly detection
Object recognition
Feature extraction

ABSTRACT

The detection of anomalous behaviors in automated video surveillance is a recurrent topic in recent computer vision research. Depending on the application field, anomalies can present different characteristics and challenges. Convolutional Neural Networks have achieved the state-of-the-art performance for object recognition in recent years, since they learn features automatically during the training process. From the anomaly detection perspective, the Convolutional Autoencoder (CAE) is an interesting choice, since it captures the 2D structure in image sequences during the learning process. This work uses a CAE in the anomaly detection context, by applying the reconstruction error of each frame as an anomaly score. By exploring the CAE architecture, we also propose a method for aggregating high-level spatial and temporal features with the input frames and investigate how they affect the CAE performance. An easy-to-use measure of video spatial complexity was devised and correlated with the classification performance of the CAE. The proposed methods were evaluated by means of several experiments with public-domain datasets. The promising results support further research in this area.

1. Introduction

The classification of human behavior in videos has been a subject of great interest in computer vision [1]. Particularly, in recent years, many efforts have been focused to detect anomalous (or abnormal) behaviors in automated video surveillance [2–10]. However, the definition of anomalous events in video surveillance is not only context-dependent but, also, dependent of human-defined semantics. As a matter of fact, there is no general rule for such a definition, except by the qualitative observation that anomalous events occur infrequently in comparison with normal events [11].

Commonly, novelty detection and anomaly detection are considered synonyms because, to date, there is no universally accepted definition for these terms [12]. In this sense, our anomaly detection model can be approached as an one-class classification problem, such that the normal class is assumed to be human-defined and with a large number of examples, while the other class corresponds to the anomaly class (i.e. samples that are not or rarely present in the normal class). For instance, in the analysis of crowded pedestrian walkways, anomaly behaviors could be the circulation of non pedestrians in the walkways (e.g. bikers), anomalous pedestrian motion and behavior patterns (e.g. wrong direction or walking on the grass), as well as abnormal objects in the scene (e.g. left baggage or thrown objects).

Even if the detection of abnormal behavior was restricted to pedestrian walkways, the corresponding anomalies might have quite different characteristics, requiring the extraction of particular features from the video frames to represent and automatically classify them. For instance, from the appearance point of view, a pedestrian walking in a wrong direction behaves similarly to a pedestrian walking in the right one. However, pedestrian *motion* patterns differ significantly regarding the direction, which may characterize an anomaly. On the other hand, unusual pedestrians, such as people in wheelchairs, present different *appearance* patterns, when compared to regular pedestrians walking on walkways, even though the motion patterns are similar. This classification is significantly more difficult in crowded scenes, as they present changes in the subject size, shape, boundaries, and occlusions.

A key issue for anomaly detection methods is the extraction of relevant features from the raw image, to enable a good classification of different types of anomalies. In the literature, the most common approach is to use spatial and temporal features to model activity patterns. Such features are based on standard computer vision techniques and other variants, such as Histogram of Oriented Gradients (HOG) [4], Histogram of Optical Flow (HOF) [13],

* Corresponding author.
E-mail addresses: manasses@ifc-videira.edu.br (M. Ribeiro), lazzaretti@utfpr.edu.br (A.E. Lazzaretti), hslopes@utfpr.edu.br, heitorslopes@gmail.com (H.S. Lopes).

http://dx.doi.org/10.1016/j.patrec.2017.07.016
0167-8655/

social force model [8], dense trajectories [14], and dynamic textures [7]. However, as pointed by Perlin and Lopes [15], those features, called hand-crafted descriptors, require that some *a priori* knowledge have to be incorporated during the training step. Such knowledge depends mostly on the surveillance target and it is difficult to define across different applications. As a result, some features may perform well in particular domains and drive classifiers to bad classification accuracy in others, even combining motion and appearance features [10].

Recently, Convolutional Neural Networks (CNNs) have achieved the state-of-the-art performance for object recognition [16,17]. A possible reason for such a high performance is that they can learn features automatically, and with superior discriminatory power for image representation, when compared to hand-crafted image descriptors [15,18]. However, CNNs are trained in a supervised way and they are not directly applicable to anomaly detection tasks, where only the normal class is known. To overcome this issue, Auto-Encoders (AEs) can be an interesting option for one-class classification problems, because it can be trained using only the normal class. The AE model was proposed by Rumelhart et al. [19] and, later, popularized by Vincent et al. [20] with the Stacked Denoising AEs (SDAE), as well as by Krizhevsky and Hinton [21]. AEs were initially used in the image retrieval context, but very recently their application for video anomaly detection has emerged [10,18]. However, AEs are not capable of capturing the 2D structure in image and video sequences, because the input data is a 1D vector. To cope with this issue, the Convolutional AE (CAE) architecture seems to be more appropriated [22].

In fact, CAEs for anomaly detection in video are still underexplored in the recent literature (see, for instance, [18]). In general, works that employ AEs cope with extracted features, such HOF and HOG, mentioned before, and a classifier. Our work proposes a different approach because we use not only the entire frames (and packages of frames and features), but also, the reconstruction errors to discriminate anomalies in videos of different levels of complexity.

Thus, the issue addressed in this work is the use of deep CAE in the anomaly detection context. The working hypothesis is that a CAE is able to learn normal events in videos, and, therefore, we hypothesize that the reconstruction error of a frame can be used for devising an anomaly score, thus allowing CAEs to be used for one-class classification tasks. As a matter of fact, humans are very competent to combine intuitively different features, such as motion and appearance features, in order to interpret the meaning of a video sequence. In this sense, this work also addresses the question: does fusing high-level information (e.g. the abovementioned features) with the input data increase the classification performance of a CAE? Finally, although video complexity is a difficult issue to be objectively evaluated, humans can successfully interpret videos within a large range of complexity. However, deep learning (DL) methods, such as a CAE, may have their performance influenced by the underlying spatial complexity of a video. Therefore, we also propose a measure of spatial video complexity and investigate the possible relationship between it and the performance of a CAE to detect anomalies in videos.

In short, with the focus on using CAE in the context of anomaly detection in videos, the main contributions of this work are:

1. To propose an anomaly score, derived from the CAE's reconstruction error and find out its possible relationship with normal and abnormal events in a video.
2. To propose a method for efficient aggregation of high-level features with the input frames and investigate how they affect the CAE's performance in detecting anomalies.
3. To devise an easy-to-use measure of spatial complexity of a video and correlate it with the classification performance of a CAE.

This paper is organized as follows. Section 2 presents some related works found in recent literature. Section 3 addresses the fundamental topics related to AEs and CAEs. Section 4 presents some topics about video spatial complexity measure. Section 5 addresses appearance and motion filters. Section 6 describes in detail the proposed methods. Section 7 presents how the experiments were done, their results and a short discussion. Finally, Section 8 reports the general conclusions drawn, and suggests future research directions.

2. Related work and contributions

Video anomaly detection methods can be categorized according to the surveillance target, type of sensors, feature extraction process, and modeling (learning) methods [1]. Regarding surveillance target, the anomaly detection can be performed on traffic, individuals, crowds, and single or multiple objects. As for the types of sensors, visible-spectrum cameras are the most frequently used. The limitation of this type of sensor is the field of view and resolution of the camera [23]. Methods for feature extraction are dependent on the surveillance target. There are two main groups: those which first perform target tracking by analyzing individual moving objects in the scene (extracting complex motion features), and those that extract features directly from the image at the pixel level [24].

Jiang et al. [11] proposed three different levels of spatio-temporal contexts to be extracted in order to perform the tracking process of all moving objects in the video. Brun et al. [25] proposed a different approach using a string kernel and tracking-based approach for evaluating the similarity between trajectories and define a novelty score for different zones in a scene. Similarly, Yang et al. [26] used a trajectory segmentation to perform the tracking process and a multi-instance learning to detect abnormal trajectories. In general, the main limitation of tracking methods in complex and crowded scenes is the presence of occluded objects, which degrade the anomaly detection performance [10].

On the other hand, features based on appearance and motion are more robust to occlusion problems in videos [7,27]. The most common features are built using 3D spatio-temporal gradients, HOG and HOF. In [8], a social force model was proposed in such a way that regions with anomalies are found in the abnormal frames by means of interaction forces and a bag of words approach. In [24], a combination of visual feature extraction and image segmentation is presented and the method works without the need of a training phase. In [28], histograms of oriented swarms is applied, together with HOG, to capture the dynamics of crowded environments. Such appearance and motion model increases the detection accuracy of local anomalies and have a lower computational cost, compared to other state-of-the-art methods. Other spatio-temporal statistical measures to characterize the overall behavior of the scene are presented in [4,6,27,29–31]. However, as discussed in Section 1, hand-crafted descriptors normally require that some *a priori* knowledge should be incorporated in the training step. In order to circumvent such limitation, in this work we propose a method for efficient aggregation of high-level features with the input frames by using a CAE and investigate how they affect the CAE in the anomaly detection process.

Regarding the learning methods, the most used approach is based on the one-class classifier, which has been extensively used for anomaly detection problems. For instance, the one-class Support Vector Machine (SVM) was used by Xu and Ricci [10]. Also, a similar one-class approach, named space-time Markov Random

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016

Field model, was devised by Kim and Grauman [5]. There are some approaches that include both the feature extraction and the learning method in a single step. In [32], a sparse combination is proposed, and it turns the original problem into a few costless small-scale least square optimization problem. Following a similar idea, a sparse reconstruction and a novel dictionary selection is presented in [33]. In [27], a probability model that takes the spatial and temporal contextual information into account is learned. The framework is unsupervised, without the need to label the training data to perform the anomaly detection task. It is also possible to include some *a priori* knowledge about the application. However, by using a different step for the classification process, it is needed to select the most appropriate classifier, in addition to specifying the descriptors in the feature extraction stage. These issues, by themselves, are hard to address for given applications. In this work, we promote the use of CAE in the context of anomaly detection. Differently from traditional methods, using a CAE, the feature extraction is integrated to the classification process. We propose the CAE's reconstruction error as the "anomaly" score so that one can discriminate between normal and abnormal events in a video.

DL methods have been investigated for computer vision problems, and they turn out to be very effective for visual recognition tasks. Several related works have appeared recently, and they are categorized according to the basic method that they are derived from, that is: CNN [17], AE [34], Restricted Boltzmann Machines [35], and Sparse Coding [36]. It is noticed that, in the context of anomaly detection, DL methods are still in early stages of development. In [37], an unsupervised Deep Belief Network (DBN) was trained to extract a set of features in a relatively low-dimensional space, and a one-class classifier (OCC) was trained with the features learned by the DBN. In general, OCCs can be inefficient for modeling decision surfaces in large and high-dimensional datasets. However, by combining the OCC with a DBN, it is possible to reduce redundant features and improve the performance for standard OCC datasets.

In [10], an appearance and motion SDAE was proposed to extract features of video surveillance datasets. Based on the features learned, multiple one-class SVM models were used to predict the anomaly scores and classify each frame. Despite the fact that AEs are very efficient methods for given applications, they cannot capture the 2D structure in image and video sequences, and the Convolutional AE (CAE) architecture can be more appropriated [22]. A similar procedure is presented in [18], where two AE (SDAE and CAE) were used to learn regular motion patterns from video sequences. The main advantage of this approach is the possibility of capturing regularities (degrees of normality) from multiple datasets jointly. Nevertheless, the anomalies may be characterized by motion and appearance features, thus requiring that the input of the CAE includes such sort of features. In this work, we propose the aggregation of high-level features, such as optical flow and edge filter, with the input frames, in order to allow the identification of different types of anomalies. At this point, it is important to emphasize that CAEs for anomaly detection in video are still underexplored in the recent literature [18].

An important issue for anomaly detection performance to be reviewed in this work is the spatial complexity of videos. Long ago, Cilibrasi and Vitányi [38] proposed the use of the Kolmogorov complexity to measure image complexity. Later, Yu and Winkler [39] showed that the Kolmogorov-based complexity of an image usually increases with decreasing resolution and that the spatial information is strongly correlated with compression-based complexity measures. From the above-mentioned works, we hypothesize that the video complexity can play an important role in the classification process, affecting its performance. This can be especially true for anomaly detection methods, where only one class is used during the training step. In this sense, a complexity measure may correlate with *a priori* performance limitation for a particular dataset. Therefore, we propose a measure of video complexity and investigate a relationship between it and the performance of a CAE to detect anomalies in videos.

3. Deep learning with auto-encoders

3.1. Auto-encoder

The AE was introduced by Rumelhart et al. [19] and is regarded as an unsupervised fully connected one-hidden-layer neural network to learn from unlabeled datasets. The idea is that the AE is trained to reconstruct the input pattern at the output of the network. An AE takes an input $\mathbf{x} \in \mathbb{R}^d$ and first maps it to the latent representation (hidden layer) $\mathbf{h} \in \mathbb{R}^{d'}$ using the mapping function $\mathbf{h} = f_\Theta = \sigma(\mathbf{W}\mathbf{x} + b)$ with parameters $\Theta = \{\mathbf{W}, b\}$. For reconstructing the input, a reverse mapping of $f : \mathbf{y} = f_{\Theta'}(h) = \sigma(bfW'\mathbf{h} + b')$ with $\Theta' = \{\mathbf{W}', b'\}$ is used. The parameters $\mathbf{W}$ learnt from the input layer to the hidden layer compose the encoder and the parameters $\mathbf{W}'$ learnt from the hidden layer to the output layer define the decoder. The decoder parameters are normally related to the parameters in the encoder by $\mathbf{W}' = \mathbf{W}^T$ [22].

Training an AE does not require label information of the input data. It uses the back propagation algorithm to minimize the reconstruction error e between each input $\mathbf{x}_i$ and the corresponding output $\mathbf{y}_i$, by adjusting the parameters of the encoder $\mathbf{W}$ and the decoder $\mathbf{W}'$, as shown in Eq. (1):

$$e(\mathbf{x}, \mathbf{y}) = \frac{1}{2N} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{y}_i\|_2^2. \tag{1}$$

The drawback of the above formulation is that, without additional constraints, the final mapping is the identity. There are different approaches proposed in the literature to circumvent such limitation. Denoising AEs (DAE) [34] are one of the most used models. A DAE reconstructs the input by using its partially corrupted version, where the corrupted version is obtained by adding some amount of noise, distributed according to the characteristics of the input vector. They can also be built using deep architectures to learn a compressed representation of the input, by limiting the number of hidden units [21].

3.2. Convolutional auto-encoder

The main limitation of AE and DAE is that they do not capture the 2D structure in image and video sequences [22]. 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. [22]. 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 CNN [21]. The loss function is similar to the AE, as presented in Eq. (2):

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

where λ is the regularization parameter for the regularization term $\|\mathbf{W}\|_2^2$, normally used during the training procedure of the CAE. As in CNNs, CAE architecture contains convolutional, deconvolutional, pooling, and unpooling layers, as presented along the subsequent sections.

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016

3.2.1. Convolutional and deconvolutional layer

The convolutional layer abstracts the information of a filter into a scalar value parameterizing 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 Eq. (3):

$$\mathbf{h}_k = \sigma(\mathbf{x} * \mathbf{W}_k + b_k), \tag{3}$$

where b is the bias, σ is an activation function (in this work, the hyperbolic tangent), and symbol $*$ corresponds to the 2D-convolution. The reconstruction is obtained using Eq. (4):

$$\mathbf{y} = \sigma\left(\sum_{k\in H} \mathbf{h}_k * \tilde{\mathbf{W}}_k + c\right), \tag{4}$$

where there is one bias c 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 $\mathbf{W}$.

The deconvolutional layer performs an inverse operation of the convolution layer with deconvolutions. The learned filters in the deconvolutional layers serve as the base to reconstruct the shape of the input, taking into account the required reshape of the output, as presented in [18]. Convolutional and deconvolutional layers can be stacked to build deep architectures for CAEs. The filters in the first layers of the convolution layer (and later layers in the deconvolution layers) extract low-level features, whilst later layers can extract high-level features of the input frames, which in this work, are basically motion and appearance frames.

3.2.2. Pooling and unpooling layer

Pooling layer was 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. It also reduces the spatial size of the representation, reducing the amount of parameters and computation along the network, by using operations such as the maximum value over non overlapping rectangular sub-regions (patches). On the other hand, unpooling layer performs the reverse operation of pooling and it reconstructs the original size of each rectangular sub-region.

4. Spatial video complexity measure

The spatial information is an useful estimator of spatial complexity in images. The Kolmogorov complexity is an objective spatial information measure with wide theoretical background that justifies its use as an spatial complexity estimator of an image [39]. Formally, the Kolmogorov complexity is the length of the shortest computer program p that produces the string s using a given description language L on an universal Turing machine U [40]. The Kolmogorov complexity $K(x)$ is defined as:

$$K(x) = \min_p\{|p| : U(p) = x\}, \tag{5}$$

where $|p|$ is the length of program p. Thereby, the Conditional Kolmogorov complexity $CK(x_0, x_1)$ can be used to determine length of shortest program that produces output x_1 from input x_0:

$$CK(x_0, x_1) = \min_p\{|p| : U(p|x_0) = x_1\}. \tag{6}$$

A normalized compression rate (NCR) based on the Kolmogorov complexity can be used to estimate the spatial complexity of an image or video. However, Kolmogorov complexity is not directly computable. Consequently, a NCR cannot be computed either, but it can be approximated using a real-world compressor [38,39]. In this work, we used the *bzip2* algorithm [41] as the real-world compressor. Therefore, $NCR(x)$ is defined as:

$$NCR(x) = \frac{s(C_0(x)) - s(C_{\max}(x))}{s(C_0(x))}, \tag{7}$$

where C is the real-world compressor, x is an uncompressed image, $s(\cdot)$ is the "size of" operator, $C_0(x)$ and $C_{\max}(x)$ represent, respectively, no-compression and maximum compression of x, attainable by C. Therefore, $NCR(x)$ is defined in the range $0 \le NCR(x) < 1$.

5. Appearance and motion filters

5.1. Canny edge detector

Edge detection algorithms include a variety of mathematical methods to detect discontinuities. Discontinuities are typically a set of points at which image brightness changes sharply and they are normally organized as edges. Edge detectors can capture important events and changes that represent objects.

The Canny edge detector [42] is a popular algorithm for this purpose. It is an optimal smoothing filter considering several criteria: detection, localization, and minimizing multiple responses to a single edge. It was shown that this filter can be approximated by first-order derivatives of Gaussians. This filter is accomplished in a multi-stage process. First, all image is smoothed by a smoothing filter, usually a Gaussian filter. After, a 2D first derivative operator is applied to the smoothed image, in order to highlight regions with first spatial derivatives. This step finds the magnitude and orientation of the gradient. Then, a process of non-maximal suppression is applied along the direction of gradient, which means that edge points are defined as points where the gradient magnitude assumes a local maximum in the gradient direction. Finally, an hysteresis thresholding procedure is applied to help noisy edges not to be broken up into multiple edge fragments.

5.2. Optical flow

The general idea of the optical flow is to represent some kind of displacement or velocity related to the distance that a pixel moves between two subsequent frames. Considering that a pixel located at (x, y) in the frame t with the intensity $I(x, y, t)$ moves by Δx, Δy and Δt in the subsequent frame, by the brightness constancy assumption [43], one can state that:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \tag{8}$$

According to [43], by using the first-order Taylor approximation, the right-side of Eq. (8) can be rewritten as:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t. \tag{9}$$

Eqs. (8) and (9) can be combined, resulting in the following general equation:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0. \tag{10}$$

By dividing both sides by Δt, one can obtain the optical flow equation in terms of velocities u and v, which define the optical flow:

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0. \tag{11}$$

The main drawback of the optical flow equation is the presence of two unknowns (u and v). There are different algorithms proposed in the literature to solve the equation by introducing additional conditions for estimating u and v. In this work, we use the

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016

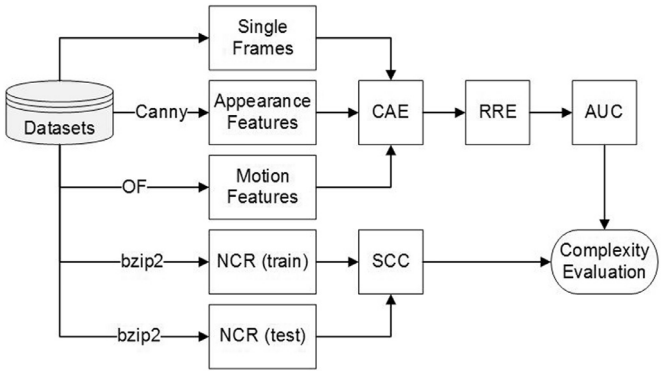

Fig. 1. Overview of the proposed approach.

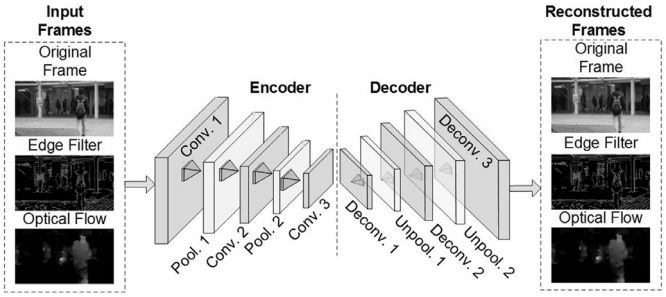

Fig. 2. Architecture of the CAE proposed in this work, based on [18].

method presented by Brox et al. [44] which, besides the brightness constancy assumption, introduces two other constraints: gradient constancy constraint and discontinuity-preserving spatio-temporal smoothness constraint. The method was implemented using a procedure based on the iterative reweighted least square method proposed by Liu [45].

6. Proposed methods

6.1. Overview

This work is intended to study the applicability of the reconstruction error of a CAE as an anomaly detector for videos. Also, we want to study the effect of appearance and motion filters and the influence of the video spatial complexity in the classification performance of video anomalies. Fig. 1 presents a high-level overview of the proposed approach, which will be detailed in the next Sections.

Firstly, appearance and motion features are extracted from frames of video datasets. Features and frames are combined to produce different scenarios (case studies) that characterize the input data to the CAE. A CAE is trained using normal events, and both normal and abnormal events are classified using the regularized reconstruction error (RRE). Next, the receiver operating characteristics (ROC) curve is plotted for several thresholds, and the area under the ROC curve (AUC) is used to measure the classification performance. The proposed complexity coefficient is calculated using the normalized compression rate (NCR) of the training and test set. Finally, the complexity evaluation is inferred by a qualitative analysis of the correlation between the spatial complexity coefficient (SCC) and AUC.

6.2. Data preparation

All the video datasets used in this work are publicly available. Each dataset is composed of a number of video clips previously labeled by their creator as normal, or with anomalies (at some point of the video). The frames are sub-sampled from video clips using a sliding window and then, both appearance and motion features are extracted. To extract the appearance features we use the Canny edge detector, described in Section 5.1, applied to each frame t. To extract motion features, we use a state-of-the-art optical flow method, described in Section 5.2, to pairs of subsequent frames t and $t-1$. Consequently, the first frame of the videos are ignored. Next, using the appearance and motion features extracted, four case studies were created by grouping the data into packages combining frame, Canny and optical flow features. The four case studies are: (1) only frames (FR case), (2) frame and edge (FR+ED case), (3) frame and optical flow (FR+OF case), and (4) frame, edge and optical flow (FR+ED+OF case).

Table 1
Dimensions of each layer of the CAE.

Layer	Dimensions
Input and Conv. 1	$256 \times 57 \times 37$
Pool. 1	$256 \times 28 \times 18$
Conv. 2	$128 \times 28 \times 18$
Pool. 2	$128 \times 14 \times 9$
Conv. 3 and Deconv. 1	$64 \times 14 \times 9$
Unpool. 1	$128 \times 14 \times 9$
Deconv. 2	$128 \times 28 \times 18$
Unpool. 2	$256 \times 28 \times 18$
Deconv. 3 and Output	$256 \times 57 \times 37$

6.3. CAE architecture

A fully convolutional auto-encoder (CAE) is used to learn different ways of fusing appearance and motion features extracted from video sequences. The hypothesis is that the CAE can model the complex combination of appearance and motion, thus learning relevant signatures of normal videos with low reconstruction error. Conversely, high reconstruction errors are expected for abnormal events (not present in the training set).

The CAE has a deep architecture organized in different encoder and decoder layers. The encoder consists of convolutional layers, whilst the decoder is based on deconvolutional layers – that are the reverse of the encoder with no tied weights. Our architecture is similar to the model recently proposed by Hasan et al. [18]. It is composed of three convolutional layers and two pooling layers on the encoder side, and the same reversed structure in the decoder side. The CAE is trained to learn the signature of normal events, considering the optimization presented in Eq. (3). The architecture of our CAE-based approach is shown in Fig. 2.

In the first convolutional layer, the CAE architecture is composed of 256 filters with stride 4. It produces 256 feature maps with resolution of 57×37 pixels. Next comes the first pooling layer that produces 256 feature maps with resolution of 28×18 pixels. All pooling layers have a 2×2 kernel, performing sub-sampling by the max-pooling method. 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 final deconvolutional layer is the reconstructed version of the input. Table 1 summarizes the details of each layer of the CAE.

The inputs to the CAE are cuboids $\mathbf{X}$ extracted from video clips. The cuboid is a 3D-structure with different number of channels, varying between one (single frame) and three (package of a frame and appearance and motion features). Each channel is a 2-D array with resolution of 235×155 pixels. The cuboids are built according to the case studies described in Section 7.

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016

6.3.1. CAE training

Training a CAE does not require the label information of the input data. However, for the purpose of this work, we used an indirect labeling, since all training instances belong to the group of videos without anomalies.

The training method uses the back propagation algorithm to minimize the reconstruction error e shown in Eq. (2). However, as mentioned before, in our approach the input data is a cuboid $\mathbf{X}$. Therefore, the reconstruction error is evaluated over all dimensions. To optimize the loss function, we use stochastic gradient descent with the adaptive sub-gradient method AdaGrad [46]. It computes a dimension-wise learning rate that adapts the rate of gradients by a function of all previous updates in each dimension. AdaGrad is widely used due to its theoretical guarantee of convergence and empirical success. The weights were initialized using the Xavier algorithm [47] that automatically determines the scale of initialization based on the number of input and output neurons. It keeps the signal in a reasonable range of values through many layers.

6.4. Regularization of reconstruction errors

Similar to [18], we compute the Reconstruction Error (RE) of pixels intensity value I at location (x, y) in frame t of the video sequence, as shown in Eq. (12):

$$RE(x, y, t) = \| I(\mathbf{X}, t) - f(I(\mathbf{X}, t)) \|_2, \tag{12}$$

where f is the model learnt by the CAE and $\mathbf{X}$ is the input cuboid. Given the RE of all pixels of a frame t, we compute the Frame Reconstruction Error (FRE) by summing all the pixel-wise errors, see Eq. (13):

$$FRE(t) = \sum_{(x,y)} RE(x, y, t). \tag{13}$$

After, we smooth the FRE of the frames using a moving average filter, according to Eq. (14), where N is the number of samples of the moving average.

$$S_{FRE}(t) = \frac{1}{N} \sum_{j=0}^{N-1} FRE(t + j). \tag{14}$$

Finally, the Regularized Reconstruction Error (RRE) of a frame is computed by Eq. (15):

$$RRE(t) = \frac{S_{FRE}(t) - \min(S_{FRE})}{\max(S_{FRE}) - \min(S_{FRE})}, \tag{15}$$

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.

6.5. Spatial complexity analysis

In our approach, we propose to use Kolmogorov complexity to estimate the spatial complexity of the video datasets (see Section 4). Considering that Kolmogorov complexity is not directly computable, we used the bzip2 algorithm [41] as real-world compressor, so as to provide an approximation to $NCR(x)$, according to Eq. (7). Both train and test video subsets of all datasets were submitted to the compression algorithm. Then, the spatial complexity coefficient (SCC) was computed according to Eq. (16), as follows:

$$SCC(train, test) = \frac{1}{NCR(train) \times NCR(test)}, \tag{16}$$

where $train$ and $test$ are the corresponding datasets. Supposing that the value computed for $SCC(train, test)$ of a given dataset is larger than that computed for another dataset, this indicates that the former has a higher spatial complexity than the latter.

6.6. Evaluation measures

In this work, AUC is used as the measure of the classification performance. The AUC demonstrates a comparison that is independent of the threshold and provides a direct analysis of the mapping performed by the classifier. This facilitates the comparison with other studies in the literature, that also use AUC [10,18,48,49]. The computation of the ROC curve is done by using the RRE defined by Eq. (15), considering the true positive rate (TPR) and false positive rate (FPR). From AUC, one can assess the Equal Error Rate (EER) that 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.

It is noteworthy that, for one-class classification problems, the automatic and complete parameter selection is still an open problem in the literature [50,51]. The main limitation is the presence of only one class during the training procedure (normal class). Such limitation implies that only true positive and false positive rates can be estimated, which compromises cross-validation based methods. Accordingly, in order to provide an indication of the most appropriate threshold to discriminate between normal and abnormal frames, different thresholds were compared, considering the distribution of the RRE from normal examples as the reference to select the threshold that results in the closest performance compared to the EER. The experiments include thresholds based on the average RRE ($\bar{A}_{RRE}$) and k standard deviations (σ_{RRE}) from normal examples. In our case, $k = \{1, 2, 3\}$, therefore, the following thresholds were considered: $\theta_1 = \bar{A}_{RRE}$, $\theta_2 = \bar{A}_{RRE} + \sigma_{RRE}$, $\theta_3 = \bar{A}_{RRE} + 2\sigma_{RRE}$, and $\theta_4 = \bar{A}_{RRE} + 3\sigma_{RRE}$.

Finally, the complexity evaluation is accomplished by analyzing the correlation between AUC and SCC and establishing its relationship with the classification performance.

7. Experiments and results

The CAE model proposed in this work was trained using a version of Caffe modified by Hasan et al. [18]. Quantitative and qualitative evaluations were done using multiple video datasets. Caffe is an open source DL framework developed by the Berkeley Vision and Learning Center (BVLC) created by Jia et al. [52]. All experiments were run in a dedicated GPU server with Intel i7-5820K CPU running at 3.3 GHz, with 32GB of RAM, and equipped with a Nvidia K40 GPU accelerator, running Ubuntu 14.04.3 LTS.

In this work, the three different datasets used are traditional benchmarks for anomaly detection problems: UCSD pedestrian dataset [49] (including the two subsets, Ped1 and Ped2), and Avenue Dataset [32]. Both are composed of a collection of videos with frames labeled as “normal” or “anomaly”. According to this binary classification, the ground-truth, annotated by a human expert, corresponds to the supposedly anomalous frames. The main features from each dataset are presented below:

- UCSD pedestrian: This video dataset was acquired by monitoring a pedestrian walkway using a stationary camera. The normal frames contain only pedestrians, whilst anomalies include bicycles, vehicles, skateboarders, and wheelchairs passing throughout the pedestrians. The subset Ped1 is composed of about 5500 normal and 3400 anomalous frames, each with 238×158 pixels of resolution. The subset Ped2 is composed of about 346 normal and 1652 anomalous frames of resolution 360×240 pixels.
- Avenue: This dataset contains 16 video clips for training and 21 for testing, comprising 15,328 and 15,324 frames, respectively. These videos were captured at the Chinese University of Hong Kong campus avenue. In the normal frames there are

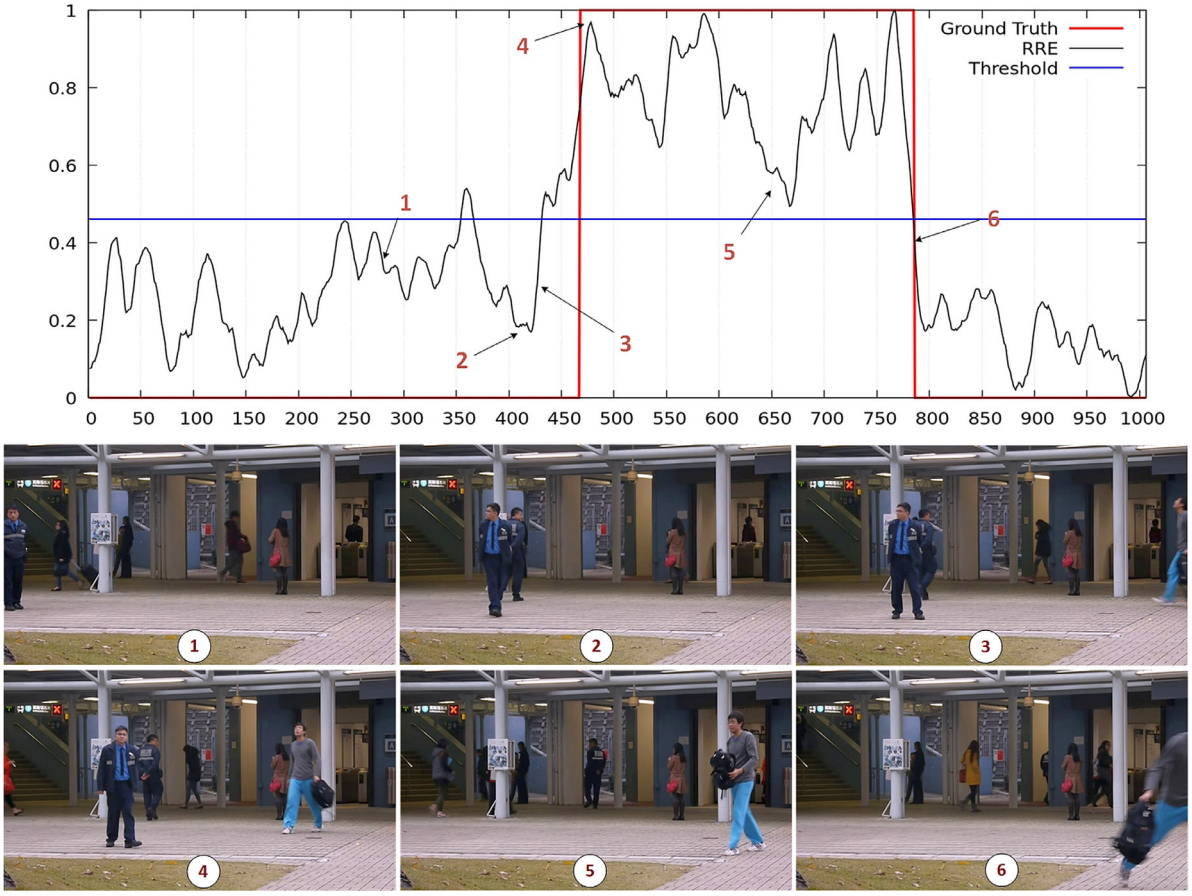

Fig. 3. (top) RRE plotted along the frames of a video clip. (bottom) Screen shots of the remarkable moments explained in the text. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

people walking by towards different directions, whilst abnormal frames include people running, throwing objects, and loitering. The resolution of these frames is 640×360 pixels.

7.1. Experiment #1: Reconstruction error for detecting anomalies

The objective of this experiment is to verify in what extent a CAE, trained with the minimization of the reconstruction error (given by the loss function of Eq. (2)) over a set of "normal" frames, is capable of devising anomalous frames, unseen in the training step.

Fig. 3(top) shows the plot of the RRE (black line), computed with Eq. (15), for a specific video of the Avenue dataset. The ground-truth, annotated by the creator of the dataset, and the threshold for anomaly detection are also plotted (red and blue line, respectively). Basically, this video shows people walking by in the background and surveillant policemen. In a given moment a men suddenly enters in the scene and throws a backpack a number of times (this is the anomaly), and then leaves.

Fig. 3(botton) shows six remarkable frames in which the behavior is commented below, so it is possible to correlate with the RRE plot:

1. In the beginning there is only people walking in the background. A policemen slowly enters in the scene around frame 200. It is followed by another policemen, who appears around frame 325. There are some fluctuations in the RRE under the threshold, thus indicating normal behavior, except by a short moment when the second policemen turns around and stands ahead the camera (frame 375).
2. One of the policeman becomes occluded by the other and, around frame 425 the first policeman goes to the background. The RRE decreases, somewhat following the movement of the scene.
3. Around frame 425 a man suddenly enters in the scene from the right side. There is a rapid increase in the RRE, detecting some unusual movement.
4. The man throws up a backpack around frame 470. Consequently, RRE crosses the anomaly threshold, indicating that such behavior was not present in training set. The man continues the threatening behavior repeating the action a number of times, and RRE oscillates accordingly.
5. Around frame 650 the man stops for a while before throwing again the backpack. The lack of abrupt movement makes RRE oscillate downwards, but still above the threshold.
6. Around frame 775 the man vanishes from the scene. In the remaining of the video there is only people walking in the background. As a consequence, RRE falls down and remains below the threshold.

Finally, it is important to highlight that, in average, the best results were obtained by using the threshold θ_2, that is, $\bar{A}_{RRE} + \sigma_{RRE}$ (see Section 6.6). With this threshold, the classifier performance corresponds to the EER performance in the ROC curve. This result can be used as an initial threshold estimation for other datasets in future and related works.

7.2. Experiment #2: effect of appearance and motion filters

The working hypothesis in this experiment is that if one enrich the input data of the CAE with high-level information, the discrim-

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016

Table 2
AUC/EER results for the four case studies and state-of-the-art over all three video datasets.

Datasets	FR		FR+ED		FR+ED+OF		FR+OF		State-of-the-art		
	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	Reference
UCSD Ped2	0.814	0.260	**0.847**	0.245	0.821	0.269	0.584	0.420	0.908	0.170	Xu and Ricci [10]
Avenue	0.738	0.328	**0.772**	0.270	0.724	0.310	0.620	0.418	0.702	0.251	Hasan et al. [18]
UCSD Ped1	0.535	0.480	0.569	0.495	**0.585**	0.431	0.545	0.475	0.927	0.160	Saligrama and Chen [48]
Weighted Mean	0.725	0.337	**0.759**	0.290	0.710	0.323	0.608	0.424	0.747	0.233	

ination between normal and abnormal frames can be increased. In this case, the high-level information was extracted from the frames themselves by means of filters (explained in Section 5) and, later, packed together with the raw data. A CAE with the architecture proposed in Section 6.3 was trained using four different combinations of data to the input layer, such that the input data aggregates video frames and both, appearance and motion features.

In this experiment we use single frames (FR case) as the baseline, and a combination of FR with motion and appearance filters was compared. First, we combined the original FR with appearance features (FR+ED case). Second, we combined the original frame FR with motion features (FR+OF case). Finally, we combined FR with both, appearance and motion features (FR+ED+OF case). Results for the four cases are summarized in Table 2, showing the AUC and EER. Also, for the sake of comparison, we included the best values obtained by the state-of-the-art methods. Numbers highlighted in bold are the overall best result obtained for each dataset. Since the number of frames of the test sets are different from dataset to dataset, we also show the mean values for each case, weighted by the number of frames, thus reflecting the global performance of classifiers.

By inspection of Table 2, one can verify that when appearance features are added, the AUC tends to be better when compared with the baseline. On the other hand, when the motion features are added the AUC tends to decrease. This study suggests that, for the above-mentioned datasets, adding appearance features to the raw input data is more relevant than to add motion features. The inclusion of motion features led to worse results possibly due to the video dynamics. In general, all videos of the datasets have similar motion patterns. People and other objects move most of the time in similar speed and direction. Then, supposedly, it is more difficult identify relevant motion patterns characterizing anomalies. On the other hand, in this context, appearance features are more relevant for identifying anomalies.

Table 2 allows a broad comparison between the state-of-the-art results to ours, for the four cases. For the UCSD Ped2 dataset, our result (FR+ED case) is close to that obtained by Xu and Ricci [10]. For the Avenue dataset, our result (FR+ED case) was better than those achieved by Hasan et al. [18]. On other hand, for the UCSD Ped1 dataset, our result is much worse than that presented by Saligrama and Chen [48]. Notice that all the above-mentioned state-of-the-art approaches are very elaborate, including many schemes such as data-augmentation, division of frames into patches to reduce iterations between objects, large CAE architectures, etc. However, no method achieve the best results for all datasets.

Table 3 shows another interesting way to present the results of experiments, at the event level, which is the best classification performance, considering both normal and abnormal classes. The Table presents the confusion matrix obtained with the EER threshold. From the Table, one can easily compute the true positive rate (TPR) and the true negative rate (TNR) for each dataset: 0.756 and 0.755, respectively, for the UCSD Ped 2 dataset; 0.730 for both, TPR and TNR of the Avenue dataset; and 0.569 and 0.568 for UCSD Ped1 dataset. For all cases, it is clear the balance between TPR and TNR.

Table 3
Confusion matrix for the EER result of all datasets. "N" represents the normal class and "A" represents the abnormal class.

	UCSD Ped2			Avenue			UCSD Ped1	
	Predicted			Predicted			Predicted	
	N	A		N	A		N	A
N	273	401	**N**	8468	1000	**N**	431	531
A	88	1236	**A**	3128	2707	**A**	327	701

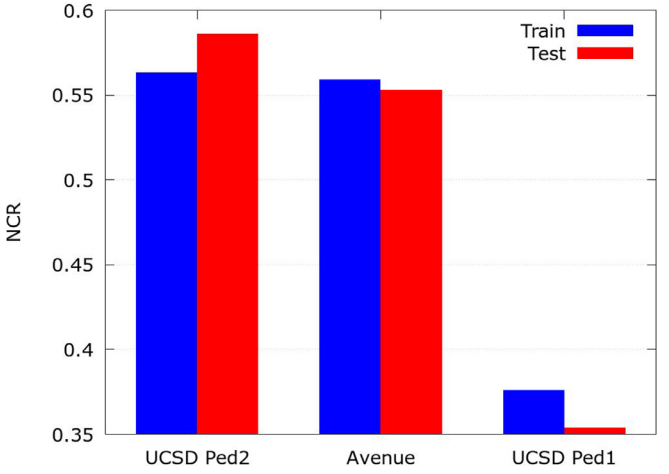

Fig. 4. Normalized compression ratio results for train an test datasets. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

Thus, by this way is possible to verify the performance of classifiers regarding correct and incorrect hits, taking into account that all datasets have unbalanced classes.

7.3. Experiment #3: effect of video spatial complexity in the classification performance

In this experiment, we investigate the possible relationship between the video spatial complexity with the classification performance of the CAE. It is supposed that, the more spatially complex the video, the harder is for the CAE to detect abnormalities. A method proposed to access such complexity was proposed in Section 6.5.

According to the previous experiment, the use of appearance features has improved results regarding the baseline. Therefore, henceforth we used the FR+ED approach for the next experiments.

The first step is to compute the NCR (Eq. (7)) for all video datasets, considering both train and test sets separately. Results are shown in Fig. 4. For the UCSD Ped2 dataset the *NCR* of the train and test sets were 0.563 and 0.586, respectively. For the Avenue dataset the *NCR* of the train and test sets were 0.559 and 0.553, respectively. Finally, for the UCSD Ped1 dataset the *NCR* of the train and test sets were 0.376 and 0.354, respectively.

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016

Table 4
AUC and SCC for all datasets.

	SCC	AUC
UCSD Ped1	3.028	0.895
Avenue	3.233	0.754
UCSD Ped2	7.513	0.547

Next, the SCC (Eq. (16)) is computed for all datasets. And, then, the spatial complexity is qualitatively estimated by analysing the behavior of AUC and SCC.

Fig. 4 shows that the UCSD Ped2 was the most compressible dataset, followed by Avenue, and then, by UCSD Ped1. The more compressible, the less complex, as seen in Table 4. In this Table, one can observe that, as the complexity of the datasets decrease, the classification performance of the CAE increase, suggesting a negative correlation between SCC and AUC. It is also remarkable the differences in compressibility (given by NCR) between the train and test sets, for all datasets. When the test set is more compressible (that is, less spatially complex) than the train set, the classification performance is higher than when the test set is less compressible (more spatially complex) than the train set.

8. Conclusions

In this work we proposed a CAE architecture to learn normal behavior signatures and, then, use the model for anomaly detection. Combinations of original frames and appearance and motion features were used as input data to the CAE. The RRE was used to measure the "anomaly level" of frames, and the classification performance was evaluated by using AUC at different thresholds. The Kolmogorov complexity was proposed as a measure of video spatial complexity. Experiments to validate the methods were performed using publicly available video datasets.

In the first experiment, a CAE was trained with a video dataset so as to devise a normal behavior signature. We showed that the use of the RRE as a "normality measure" allowed the discrimination of anomalies. The oscillations of RRE along with frames of a video seem to follow some ongoing events and, by using an adequate threshold, we showed that it is possible to distinguish between normal behavior and anomalies with reasonable accuracy for some video datasets. Our experiments suggested the use of $\bar{A}_{RRE} + \sigma_{RRE}$ as a "normality threshold". However, this cannot be generalized to other datasets.

The second experiment investigated how the aggregation of high-level information to raw data can improve classification performance of the CAE. The proposed method allowed the fusion of multiple channels of information, in our case, appearance and motion features. Results indicated that some features were useful, while others were not. Therefore, as a general conclusion, aggregating high-level information can be valuable, provided the user can devise which kind of filters can better capture the nature of the anomalies intended to be detected.

The third experiment tested a simple method to estimate the spatial complexity of videos by using a real-world compressor. Results suggested a negative correlation between the complexity and the classification performance of the CAE. Notwithstanding, it was not yet possible to state if the presence of anomalies in a video increases or not its spatial complexity, since it can be affected by other factors, such as poor correlation between pixels of an image. This investigation is left for future work.

The detection of abnormal behavior in videos, specially in surveillance videos, is a subject of growing research. 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. Overall, the work suggest that there is much yet to be done towards a more general and formal definition of normality/anomaly, so as to support researchers to devise efficient computational methods to mimetize the semantic interpretation of visual scenes by humans.

Acknowledgments

Author M. Ribeiro would like to thank the Catarinense Federal Institute and CAPES for the scholarship; author H.S.Lopes would like to thank to CNPq for the research grant number 440977/2015-0. All authors would like to thank to NVIDIA for the donation of a GPU for this work.

References

[1] A.A. Sodemann, M.P. Ross, B.J. Borghetti, A review of anomaly detection in automated surveillance, IEEE Trans. Syst., Man Cybern. Part C 42 (6) (2012) 1257–1272.
[2] M. Bertini, A. Del Bimbo, L. Seidenari, Multi-scale and real-time non-parametric approach for anomaly detection and localization, Comput. Vis. Image Understanding 116 (3) (2012) 320–329.
[3] C. Chen, Y. Shao, X. Bi, Detection of anomalous crowd behavior based on the acceleration feature, IEEE Sens. J. 15 (12) (2015) 7252–7261.
[4] K.-W. Cheng, Y.-T. Chen, W.-H. Fang, Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation, IEEE Trans. Image Process. 24 (12) (2015) 5288–5301.
[5] J. Kim, K. Grauman, Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates, in: Proceddings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2921–2928.
[6] N. Li, X. Wu, D. Xu, H. Guo, W. Feng, Spatio-temporal context analysis within video volumes for anomalous-event detection and localization, Neurocomputing 155 (2015) 309–319.
[7] W. Li, V. Mahadevan, N. Vasconcelos, Anomaly detection and localization in crowded scenes, IEEE Trans. Pattern Anal. Mach. Intell. 36 (1) (2014) 18–32.
[8] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2, 2009, pp. 935–942.
[9] Y. Yuan, J. Fang, Q. Wang, Online anomaly detection in crowd scenes via structure analysis, IEEE Trans. Cybern. 45 (3) (2015) 562–575.
[10] D. Xu, E. Ricci, Learning deep representations of appearance and motion for anomalous event detection, in: Proceedings of British Machine Vision Conference, 2015, pp. 1–12.
[11] F. Jiang, J. Yuan, S.A. Tsaftaris, A.K. Katsaggelos, Anomalous video event detection using spatiotemporal context, Comput. Vis. Image Understanding 115 (3) (2011) 323–333.
[12] M.A. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Process. 99 (2014) 215–249.
[13] T. Wang, H. Snoussi, Detection of abnormal visual events via global optical flow orientation histogram, IEEE Trans. Inf. Foren. Secur. 9 (6) (2014) 988–998.
[14] H. Wang, A. Kläser, C. Schmid, C.L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vis. 103 (1) (2013) 60–79.
[15] H.A. Perlin, H.S. Lopes, Extracting human attributes using a convolutional neural network approach, Pattern Recognit. Lett. 68 (2) (2015) 250–259.
[16] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, 25, 2012, pp. 1097–1105.
[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015.
[18] M. Hasan, J. Choi, J. Neumann, A.K. Roy-Chowdhury, L.S. Davis, Learning temporal regularity in video sequences, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 733–742.
[19] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back--propagating errors, Nature 323 (1986) 533–536.
[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408.
[21] A. Krizhevsky, G.E. Hinton, Using very deep autoencoders for content-based image retrieval, in: Proceedings of 19th European Symposium on Artificial Neural Networks, 2011.
[22] J. Masci, U. Meier, D. Cireşan, J. Schmidhuber, Stacked convolutional auto-encoders for hierarchical feature extraction, in: Proceedingd of 21th International Conference on Artificial Neural Networks, I, 2011, pp. 52–59.
[23] A. Adam, E. Rivlin, I. Shimshoni, D. Reinitz, Robust real-time unusual event detection using multiple fixed-location monitors, IEEE Trans. Pattern Anal. Mach. Intell. 30 (3) (2008) 555–560.
[24] A. Pennisi, D.D. Bloisi, L. Iocchi, Online real-time crowd behavior detection in video sequences, Comput. Vis. Image Understanding 144 (2016) 166–176.

[25] L. Brun, A. Saggese, M. Vento, Dynamic scene understanding for behavior analysis based on string kernels, IEEE Trans. Circ. Syst. Video Technol. 24 (10) (2014) 1669–1681.
[26] W. Yang, Y. Gao, L. Cao, TRASMIL: a local anomaly detection framework based on trajectory segmentation and multi-instance learning, Comput. Vis. Image Understanding 117 (10) (2013) 1273–1286.
[27] T. Xiao, C. Zhang, H. Zha, Learning to detect anomalies in surveillance video, IEEE Signal Process. Lett. 22 (9) (2015) 1477–1481.
[28] V. Kaltsa, A. Briassouli, I. Kompatsiaris, L.J. Hadjileontiadis, M.G. Strintzis, Swarm intelligence for detecting interesting events in crowded environments, IEEE Trans Image Process. 24 (7) (2015) 2153–2166.
[29] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1446–1453.
[30] Y. Cong, J. Yuan, Y. Tang, Video anomaly search in crowded scenes via spatio-temporal motion context, IEEE Trans. Inf. Foren. Secur. 8 (10) (2013) 1590–1599.
[31] J. Wang, Z. Xu, Spatio-temporal texture modelling for real-time crowd anomaly detection, Comput. Vis. Image Understanding 144 (2016) 177–187.
[32] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 FPS in MATLAB, in: Proceedings of International Conference on Computer Vision, 2013, pp. 2720–2727.
[33] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: Proceedings IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3449–3456.
[34] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceddings of 25th International Conference on Machine Learning, 2008, pp. 1096–1103.
[35] R. Salakhutdinov, G.E. Hinton, Deep Boltzmann machines, 2009.
[36] S. Gao, I.W.H. Tsang, L.T. Chia, P. Zhao, Local features are not lonely – Laplacian sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3555–3561.
[37] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recognit. 58 (2016) 121–134.
[38] R. Cilibrasi, P.M.B. Vitányi, Clustering by compression, IEEE Trans. Inf. Theory 51 (2005) 1523–1545.
[39] H. Yu, S. Winkler, Image complexity and spatial information, in: Proceedings of 5th International Workshop on Quality of Multimedia Experience, 2013, pp. 12–17.
[40] O. Watanabe, Kolmogorov Complexity and Computational Complexity, 1st, Springer-Verlag, Berlin, Heidelberg, 2012.
[41] J. Seward, Bzip2 C library, 2017.
[42] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach.Intell. 8 (6) (1986) 679–698.
[43] R. Szeliski, Computer Vision: Algorithms and Applications, 1, Springer-Verlag, New York, 2010.
[44] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: Proceedings of European Conference on Computer Vision, in: Lecture Notes in Computer Science, 3024, Springer, 2004, pp. 25–36.
[45] C. Liu, Beyond Pixels: Exploring New Representations and Applications for Motion Analysis, Massachusetts Institute of Technology, 1999. Ph.D. thesis.
[46] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (2011) 2121–2159.
[47] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of 13th International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[48] V. Saligrama, Z. Chen, Video anomaly detection based on local statistical aggregates, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2112–2119.
[49] V. Mahadevan, W.-X. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1975–1981.
[50] B. Mack, R. Roscher, B. Waske, Can i trust my one-class classification? Remote Sens. 6 (9) (2014) 8779–8802.
[51] Y. Xiao, H. Wang, W. Xu, Parameter selection of gaussian kernel for one-class SVM, IEEE Trans. Cybern. 45 (5) (2015) 927–939.
[52] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of 22nd ACM International Conference on Multimedia, 2014, pp. 675–678.

Please cite this article as: M. Ribeiro et al., A study of deep convolutional auto-encoders for anomaly detection in videos, Pattern Recognition Letters (2017), http://dx.doi.org/10.1016/j.patrec.2017.07.016