Detecting Anomalous Events in Videos by Learning Deep Representations of Appearance and Motion

Dan Xu[a,*], Yan Yan[a], Elisa Ricci[b], Nicu Sebe[a]

[a] *Department of Computer Science, University of Trento, Trento, Italy*
[b] *Fondazione Bruno Kessler (FBK), Trento, Italy*

Abstract

Anomalous event detection is of utmost importance in intelligent video surveillance. Currently, most approaches for the automatic analysis of complex video scenes typically rely on hand-crafted appearance and motion features. However, adopting user defined representations is clearly suboptimal, as it is desirable to learn descriptors specific to the scene of interest. To cope with this need, in this paper we propose Appearance and Motion DeepNet (AMDN), a novel approach based on deep neural networks to automatically learn feature representations. To exploit the complementary information of both appearance and motion patterns, we introduce a novel double fusion framework, combining the benefits of traditional early fusion and late fusion strategies. Specifically, stacked denoising autoencoders are proposed to separately learn both appearance and motion features as well as a joint representation (*early fusion*). Then, based on the learned features, multiple one-class SVM models are used to predict the anomaly scores of each input. Finally, a novel *late fusion* strategy is proposed to combine the computed scores and detect abnormal events. The proposed ADMN is extensively evaluated on publicly available video surveillance datasets, showing competitive performance with respect to state of the art approaches.

Keywords: Video surveillance, abnormal event detection, unsupervised learning, stacked denoising auto-encoders, feature fusion

1. Introduction

In the last few years the massive deployment of distributed camera systems in public spaces has increased the need for advanced tools performing the automatic analysis of video surveillance streams. A fundamental challenge in intelligent video surveillance is to automatically detect anomalous events in complex and crowded scenes. This problem has attracted considerable attention in the computer vision research community [1, 2, 3, 4].

Early works in the literature are based on the analysis of individual moving objects in the scene [5, 6, 7, 8]. First, visual tracking is performed to compute the trajectories of the targets and a model is learned describing typical activities. Then, anomalous events are identified by looking at patterns which distinctly diverge from the model. However, these methods are not suitable for analyzing complex scenes, as the accuracy of visual tracking algorithms significantly degrades in case of several occluded targets. Therefore, more recently, unsupervised non-object centric approaches have gained popularity [2, 4, 9, 10, 11, 12, 13]. These methods address the anomaly detection task by analyzing the co-occurence of atomic spatio/temporal patterns and are based on hand-crafted features extracted from low-level appearance and motion cues. Commonly used low-level features include histogram of oriented gradients (HOG), 3D spatio-temporal gradient, histogram of optical flow (HOF). However, adopting generic user defined features is a clear limitation of these approaches and improved performance can be obtained by learning scene specific descriptors.

Recently, deep learning approaches have been successfully used to tackle various computer vision tasks, such as object classification [14], object detection [15] and activity recognition [16]. While these works mostly focus on supervised learning tasks and Convolutional Neural Networks, unsupervised approaches have also gained popularity. In particular, autoencoder networks [17] have been investigated to address fundamental tasks such as object tracking [18] and face align-

[*] Corresponding author: Dan Xu. Email address: dan.xu@unitn.it

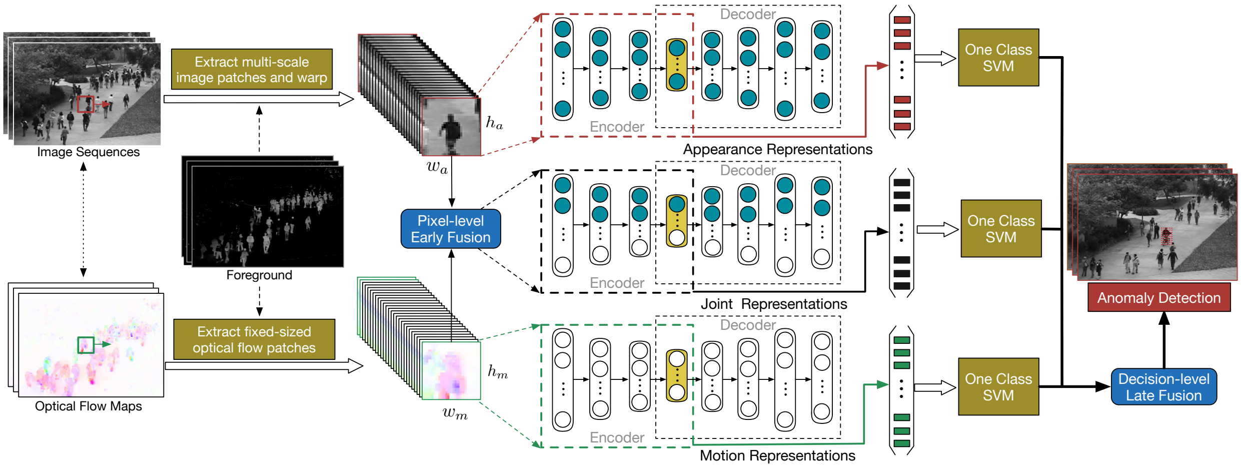


Figure 1: Overview of the proposed AMDN approach for abnormal video event detection.

ment [19]. In both scenarios, improved performance over traditional methods can be achieved, since using deep architectures rich and discriminative features can be learned via multi-layer nonlinear transformations.

Following this intuition, in this paper we propose a novel approach for detecting anomalous activities in complex video surveillance scenes. Opposite to previous works [11, 12, 2, 13] which rely on hand-crafted features to model spatio/temporal activity patterns, we propose to learn discriminative feature representations in a fully unsupervised manner adopting stacked denoising autoencoders (SDAE) [20]. Figure 1 shows an overview of the proposed method, named Appearance and Motion DeepNet (AMDN). Our AMDN is based on a novel *double fusion* scheme (integrating both traditional early fusion and late fusion strategies) for combining low-level features of appearance and motion. Specifically, in the first phase, still image patches and optical flow fields are provided as input to two separate SDAE networks, to learn appearance and motion features, respectively. A third SDAE is used to learn a joint representation of appearance and motion from the concatenation of image pixels and the corresponding optical flow (*early fusion*). In the second phase, multiple one-class SVM models, corresponding to the learned feature representations, are used to compute a set of anomaly scores. Then, a novel *late fusion* scheme is proposed to combine the computed scores for abnormal event prediction. The proposed AMDN is evaluated on three challenging video surveillance datasets and compared with several state of the art methods. Our experiments clearly demonstrate the effectiveness of the proposed double fusion framework as well as the importance of learning features with SDAEs.

To summarize, the main contributions of this work are threefold:

- To the best of our knowledge, this paper represents the first attempt to address the anomalous event detection task using deep learning architectures. In this way, discriminative feature representations can be automatically learned for the scene of interest, showing significant advantages over previous methods based on hand-crafted features.
- Our AMDN learns appearance and motion features as well as their correlations. Deep learning methods for combining multiple modalities have been investigated in previous works [21, 22]. However, none of these works consider the anomaly detection task.
- A double fusion scheme is proposed to combine appearance and motion features. The advantages of combining early and late fusion approaches have been demonstrated in previous works [23]. However in [23] the authors did not consider a deep learning framework, neither the problem of discovering unusual activities in video surveillance streams.

This paper extends our previous work in [24]. Specifically, with respect to [24] in this paper we added a section discussing related work on abnormal video event detection and on learning deep representations in unsupervised settings (Section 2). Moreover, in Section 3 we provide further insights on the proposed AMDN framework, enriching the descriptions of the main components of our systems (SDAEs - Section 3.2.2, one-class

SVM - Section 3.3.1) and introducing a novel late fusion strategy based on ℓ_p-norm (Section 3.3.2). Finally, we significantly expanded the experimental evaluation, adding results on a third publicly available dataset and performing a detailed analysis of the different components of our method (Section 4).

In the rest of this paper, we first review the related work in Section 2. Then, we introduce the proposed AMDN framework in Section 3, describing the AMDN structure in detail and the approach we used for training our SDAEs. Finally, the proposed method is evaluated extensively and the experimental results are presented in Section 4. Conclusions are drawn in Section 5.

2. Related Work

In this section we review previous works considering: (i) the addressed task, *i.e.* abnormal video event detection and (ii) deep learning approaches in unsupervised settings.

2.1. Abnormal Event Detection in Videos

Existing techniques tackling the abnormal video event detection are extensively reviewed in [25]. These methods can be mostly partitioned in two categories depending on the types of event representations adopted, namely trajectory-based methods and non-object centric methods.

2.1.1. Trajectory-based Methods

Trajectories are widely used features for abnormal video event detection [26, 6], due to their ability of describing synthetically the dynamic information of foreground objects. Trajectory-based methods usually rely on two phases. First, visual tracking algorithms are used to estimate the motion of the objects and the people in the scene. Then, features representing the trajectories of the targets are employed to construct statistical models describing typical activities. In the second phase, the activities corresponding to trajectories which deviate significantly from the learned model are identified as anomalous [27, 28].

A pioneering work on this research line is [29], where object trajectories are modeled using probability density functions. In [30] Hu *et al.* developed a multiple objects tracking algorithm to collect trajectories, which which are then used to learn statistical distributions. Both spatial and temporal information are considered for anomaly detection. Markris and Ellis [31] proposed a Bayesian approach for detecting abnormal trajectories based on annotated scene semantics. Jiang *et al.* [32] introduced a dynamic hierarchical clustering framework for trajectory grouping using Hidden Markov Models (HMMs) to represent each group of trajectories.

In general, trajectory-based methods guarantee satisfactory performance when foreground objects are easy to detect and track, *e.g.* in indoor environments or when there are few targets in the scene. On the contrary, performance significantly degrades in unconstrained scenarios (*e.g.* in case of dense crowds), when the several occlusions make traditional tracking and detection approaches not reliable.

2.1.2. Non-object Centric Methods

This category of approaches address the anomaly detection task by learning representative activity patterns from behavior-related attributes of objects and people within spatial/temporal contexts. Commonly considered behavioral attributes include size, gradient, direction and speed of the targets in the scene, which are described with low-level representations such as HOG [33], 3D spatio-temporal gradient [34], HOF [35] and dense spatial-temporal interest points (Dense STIPs) [36].

Cong *et al.* [4] employed multi-scale histograms of optical flow and a sparse coding model and used the reconstruction error as a metric for outlier detection. Mehran *et al.* [9] proposed a “social force” model based on optical flow features to represent crowd activity patterns and identify anomalous activities. In [10] co-occurrence statistics of spatio-temporal events are adopted in combination with Markov Random Fields (MRFs) to discover unusual activities. Spatio-temporal MRFs are also employed in [37]. Multiple spatio-temporal filters at different scales and local feature descriptors are considered in [2]. Kratz *et al.* [38] introduced an HMMs-based approach for detecting abnormal events through analyzing the motion variation of local space-time volumes. Ricci *et al.*[13] proposed a convex hierarchical clustering approach to detect abnormal events in time and space at different scale.

The advantage of these methods over trajectory-based ones is that, working at pixel level or, more generally, on 2D cells/3D cubes, they are more robust in case of complex scenes. However, all these approaches rely on hand-crafted features which are difficult to define *a priori* due to the huge variations of anomalous behaviors. Our AMDN represents one of the first attempts in the computer vision community to overcome these issues by learning deep feature representations.

2.2. *Deep Learning Models in Unsupervised Setting*

Deep learning techniques have recently achieved remarkable success in the computer vision field, beating the state-of-the-art in various challenging tasks [14, 16, 18]. Unsupervised deep learning approaches have also received increasing popularity, as a large amount of annotated training data is usually relatively difficult to obtain in a variety of real-world applications. Commonly used unsupervised deep models include deep belief networks [39] and stacked autoencoders [40], which can be efficiently trained with layer-wise pretraining and fine-tuning [41]. These models have shown much more representative power than their shallow counterparts, resulting in improved performance on several tasks. For instance, Kan *et al.* [42] addressed the cross-pose face recognition problem using stacked autoencoders. In this way pose-invariant features are learned from faces with different poses, obtaining superior performance than previous methods. In [18], the authors proposed a deep denoising autoencoder network for robust visual tracking and learning effective target representations. To our knowledge, no existing works in the literature considered an unsupervised deep learning framework for learning multiple features to tackle the abnormal event detection problem.

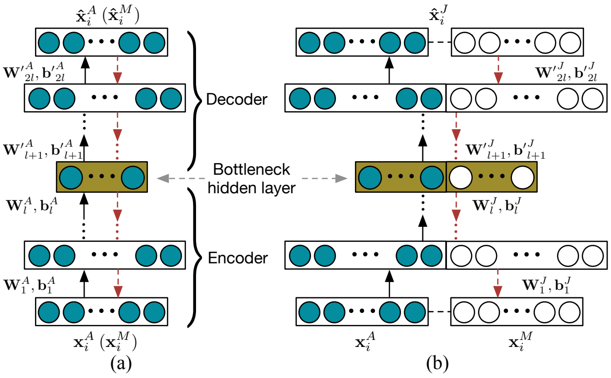
3. AMDN for Abnormal Event Detection

3.1. *Overview*

As discussed in Section 1, the proposed AMDN framework for detecting anomalous activities is based on two main steps (Fig.1). In the first phase, SDAEs are used to learn appearance and motion representations of visual data, as well as a joint representation capturing the correlation between appearance and motion features (Sec. 3.2). In the second phase (Sec. 3.3), three separate one-class SVMs [43] are learned based on the different types of feature representations. Once the one-class SVM models are trained, given a test sample corresponding to an image patch, three anomaly scores are computed and combined. The combination of the one-class SVM scores is obtained with a novel late fusion scheme. In the following we describe the proposed approach in details.

3.2. *SDAEs in AMDN*

In this subsection we first review some basic concepts about denoising autoencoders (DAEs) and then describe the details of the proposed approach for learning deep representations of appearance and motion.



$\hat{\mathbf{x}}_i^A$ $(\hat{\mathbf{x}}_i^M)$
$\mathbf{W'}_{2l}^A, \mathbf{b'}_{2l}^A$
Decoder
$\mathbf{W'}_{l+1}^A, \mathbf{b'}_{l+1}^A$
Bottleneck hidden layer
$\mathbf{W}_l^A, \mathbf{b}_l^A$
Encoder
$\mathbf{W}_1^A, \mathbf{b}_1^A$
$\mathbf{x}_i^A$ $(\mathbf{x}_i^M)$
(a)
$\hat{\mathbf{x}}_i^J$
$\mathbf{W'}_{2l}^J, \mathbf{b'}_{2l}^J$
$\mathbf{W'}_{l+1}^J, \mathbf{b'}_{l+1}^J$
$\mathbf{W}_l^J, \mathbf{b}_l^J$
$\mathbf{W}_1^J, \mathbf{b}_1^J$
$\mathbf{x}_i^A$
$\mathbf{x}_i^M$
(b)

Figure 2: (a) The structure of the (a) appearance and motion and (b) joint SDAEs for learning feature representations.

3.2.1. *Background on Denoising Autoencoders*

A denoising autoencoder [17] is a one-hidden-layer neural network which is trained to reconstruct a sample $\mathbf{x}_i$ from its (partially) corrupted version $\tilde{\mathbf{x}}_i$. Typical corrupted inputs are obtained by drawing samples from a conditional distribution $p(\mathbf{x}|\tilde{\mathbf{x}})$ (*e.g.* common choices for corrupting samples are additive Gaussian white noise or salt-pepper noise).

A DAE can be divided into two parts, *i.e.* encoder and decoder, connected by a single hidden layer. The two parts are used to learn two mapping functions, $f_e(\mathbf{W}, \mathbf{b})$ and $f_d(\mathbf{W}', \mathbf{b}')$, where $\mathbf{W}, \mathbf{b}$ denote the weights and the bias term of the encoder part, while $\mathbf{W}', \mathbf{b}'$ refer to the corresponding parameters of the decoder. For a corrupted input $\tilde{\mathbf{x}}_i$, a compressed hidden layer representation $\mathbf{h}_i$ can be obtained through $\mathbf{h}_i = f_e(\tilde{\mathbf{x}}_i \mid \mathbf{W}, \mathbf{b}) = \sigma(\mathbf{W}\tilde{\mathbf{x}}_i + \mathbf{b})$. Then, the decoder tries to recover the original input $\mathbf{x}_i$ from $\mathbf{h}_i$ computing $\hat{\mathbf{x}}_i = f_d(\mathbf{h}_i \mid \mathbf{W}', \mathbf{b}') = s(\mathbf{W}'\mathbf{h}_i + \mathbf{b}')$. The function $\sigma(\cdot)$ and $s(\cdot)$ are activation functions, which are typically nonlinear transformations such as the sigmoid. Using this encoder/decoder structure, the network can learn a more stable and robust feature representations of the input.

At training time, given a training set $\mathcal{T} = \{\mathbf{x}_i\}_{i=1}^N$, a DAE learns its parameters $(\mathbf{W}, \mathbf{W}', \mathbf{b}, \mathbf{b}')$ by solving the following optimization problem:

$$\min_{\mathbf{W},\mathbf{W}',\mathbf{b},\mathbf{b}'} \sum_{i=1}^{N} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2^2 + \lambda(||\mathbf{W}||_F^2 + ||\mathbf{W}'||_F^2) \quad (1)$$

where $|| \cdot ||_F$ denotes the Frobenius norm. The first term represents the average reconstruction error, while the weight penalty term is introduced for regularization. The parameter λ balances the importance of the two terms. Typically, sparsity constraints are imposed on the output of the hidden units to discover meaningful representations from the data [18]. If we let μ_j be the target sparsity level and $\hat{\mu}_j = \frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_i^j$ be the average activation values all over all training samples for the j-th unit, an extra penalty term based on cross-entropy, $\varphi(\boldsymbol{\mu}||\hat{\boldsymbol{\mu}}) = -\sum_{j=1}^{H}[\mu_j \log(\hat{\mu}_j) + (1-\mu_j)\log(1-\hat{\mu}_j)]$, can be added to (2) to learn a sparse representation. Here, H is the number of hidden units. The optimization problem (2) has a non-convex objective function and gradient descent can be used to compute a local optima.

3.2.2. SDAEs Structure and Training

Structure. The proposed approach for detecting anomalous event rely on three SDAE networks (Fig.1) associated to different types of low-level inputs. These SDAE are used to learn appearance and motion features as well as a joint representation of them. The basic structures of the proposed SDAE networks is illustrated in Fig. 2 (a) and (b). For the encoder part, we use an over-complete set of filters in the first layer to capture a representative information from the data. Then, the number of neurons is reduced by half in the next layer until reaching the "bottleneck" hidden layer. The decoder part has a symmetric structure with respect to the encoder part.

Specifically, the first SDAE learns mid-level *appearance* representations from the original image pixels. To capture rich appearance attributes, a multi-scale sliding-window approach with a stride d is used to extract dense image patches, which are then warped into equal size $w_a \times h_a \times c_a$, where w_a, h_a are the width and height of each patch and c_a is the number of the channels ($c_a = 1$ for gray images). The warped patches $\mathbf{x}_i^A \in \boldsymbol{R}^{w_a \times h_a \times c_a}$ are used for training. All the patches are linearly normalized into a range [0, 1]. We stack 4 encoding layers with $v_a \times w_a \times h_a \times c_a$ neurons in the first layer, where $v_a > 1$ is an amplification factor for constructing an over-complete set of filters. The use of over-complete representations in combination with sparsity terms have been shown to be effective in learning meaningful compressed representations in previous works [20, 44].

The second SDAE is used to learn the *motion* features. We compute dense optical flow and we use a sliding window approach with windows of fixed size $w_m \times h_m \times c_m$ ($c_m = 2$ for optical flow magnitude along x and y axes) for motion representation learning. Similar to the appearance feature pipeline, the patches $\mathbf{x}_i^M \in \boldsymbol{R}^{w_m \times h_m \times c_m}$ are normalized into [0,1] within each channel and 4 encoding layers are used. The number of neurons of the first layer is set to $v_m \times w_m \times h_m \times c_m$, $v_m > 1$.

While the first two SDAEs learn appearance and motion features separately, to take into account the *correlations between motion and appearance* we propose to couple these two pipelines to learn a joint representation (Fig. 2 (b)). The network training data $\mathbf{x}_i^J \in \boldsymbol{R}^{w_j \times h_j \times (c_a + c_m)}$ are obtained through a pixel-level early fusion of the gray image patches and the corresponding optical flow patches.

Training. The proposed SDAEs are trained separately. We rely on the typical learning scheme based on two steps: pretraining and fine-tuning. The network parameters are initialized through pretraining all layers, and then fine-tuning is used to adjust parameters over the whole network.

Given a training set $\mathcal{T}^k = \{\mathbf{x}_i^k\}_{i=1}^{N^k}$, $k \in \{A, M, J\}$ corresponding to appearance, motion and joint representation, the layer-wise pretraining learns the parameters of each SDAE minimizing the reconstruction loss regularized by a sparsity-inducing term, *i.e.*:

$$J = \sum_{i=1}^{N_k} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||_2^2 + \lambda(||\mathbf{W}||_F^2 + ||\mathbf{W}'||_F^2) + \gamma\varphi(\boldsymbol{\mu}||\hat{\boldsymbol{\mu}}). \quad (2)$$

In each layer, the input is corrupted to learn the mapping function, which is then used to produce the representation for the next layer with uncorrupted inputs. Fine-tuning consider all the layers of each SDAE as a single model. The backpropagation algorithm can be used to fine-tune the network.

The following objective function is used for fine-tuning:

$$J' = \sum_{i=1}^{N^k} ||\mathbf{x}_i^k - \hat{\mathbf{x}}_i^k||_2^2 + \lambda_F \sum_{j=1}^{L}(||\mathbf{W}_j^k||_F^2 + ||\mathbf{W}'^k_j||_F^2) \quad (3)$$

where λ_F is a user defined parameter and $2L + 1$ is the number of layers in the SDAEs. Similar to previous works [41], here we remove the sparsity regularization because the pre-trained weights will serve as regularization to the network. To speed up the convergence during training, stochastic gradient descent is employed and the training set is divided into mini-batches.

Once training is complete, the appearance, motion and joint feature representations can be computed to perform video anomaly detection.

In this work, we choose the output of the "bottleneck" hidden layer to obtain a more compact representation. Let $\mathbf{x}_i^k$ be the i-th input data sample, and $\sigma_l^k(\mathbf{W}_l^k, \mathbf{b}_l^k)$ be the mapping function of the l-th hidden layer of the k-th SDAE pipeline. The learned features, $\mathbf{s}_i^k$, can be extracted through a forward pass computing, *i.e.*

$$\mathbf{s}_i^k = \sigma_L(\sigma_{L-1}(\cdots\sigma_1(\mathbf{W}_1^k\mathbf{x}_i^k + \mathbf{b}_1^k))), \quad (4)$$

where the L-th hidden layer is the "bottleneck" hidden layer.

3.3. Abnormal Event Detection with Deep Representations

In this work the video anomaly detection problem is formulated as a patch-based binary categorization problem, *i.e.* given a test frame we adopt a sliding window approach and classify each patch as corresponding to a normal or an abnormal event. Specifically, given the t-th test patch, we compute the associated deep features representations $\mathbf{s}_t^k$, $k \in \{A, M, J\}$. Then, we rely on three one-class SVM models to calculate a set of anomaly scores $A(\mathbf{s}_t^k)$ (Subsection 3.3.1). Finally, the scores are linearly combined to obtain the global anomaly score $\mathcal{A}(\mathbf{s}_t^k) = \sum_{k\in\{A,M,J\}} \alpha^k A(\mathbf{s}_t^k)$ (Subsection 3.3.2). In the following we describe these phases in details.

3.3.1. One-class SVM Modeling

One-class SVM is a widely used algorithm for outlier detection, where the main idea is to learn a hypersphere in the feature space and map most of the training data into it. The outliers of the data distribution correspond to point lying outside the hypersphere. While other approaches can be considered to compute anomaly scores, we consider one-class SVMs as it has been shown to be effective in previous works on abnormal event detection from surveillance videos using hand-crafted features [45]. Formally, given a set of training samples $\mathcal{S} = \{\mathbf{s}_i^k\}_{i=1}^{N^k}$, the underlying problem of one-class SVM can be formulated as the following quadratic program:

$$\begin{aligned} \min_{\boldsymbol{\theta},\rho} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 + \frac{1}{\mu N^k}\sum_{i=1}^{N^k} \xi_i - \rho \\ \text{s.t.} \quad & \boldsymbol{\theta}^T \phi(\mathbf{s}_i^k) \geq \rho - \xi_i,\ \xi_i \geq 0. \end{aligned} \tag{5}$$

where $\boldsymbol{\theta}$ is the learned weight vector, ρ is the offset, $\phi(\cdot)$ is a feature projection function which maps the feature vector $\mathbf{s}_i^k$ into a higher dimensional feature space. The user defined parameter $\mu \in (0, 1]$ regulates the expected fraction of outliers distributed outside the hypersphere. Introducing a nonlinear mapping, the projection function $\phi(\cdot)$ can be defined implicitly by introducing an associated kernel function $k(\mathbf{s}_i^k, \mathbf{s}_j^k) = \phi(\mathbf{s}_i^k)^T\phi(\mathbf{s}_j^k)$ and (5) can be solved in the corresponding dual form [43]. In our experiments we consider a radial basis function kernel, $k(\mathbf{s}_i^k, \mathbf{s}_j^k) = e^{\frac{-\|\mathbf{s}_i^k - \mathbf{s}_j^k\|^2}{2\sigma^2}}$. Given the optimal $\boldsymbol{\theta}$ and ρ obtained by solving (5), an outlier score for a test sample $\mathbf{s}_t^k$ of the k-th SDAE pipeline can be estimated by computing:

$$A(\mathbf{s}_t^k) = \rho - \boldsymbol{\theta}^T\phi(\mathbf{s}_t^k) \tag{6}$$

3.3.2. Flexible ℓ_p-norm Late Fusion for Anomaly Detection

In this subsection, we propose a flexible unsupervised late fusion scheme to automatically learn the weights $\boldsymbol{\alpha} = [\alpha^A, \alpha^M, \alpha^J]$. These parameters are used to compute the anomaly score $\mathcal{A}(\mathbf{s}_t^k) = \sum_{k\in\{A,M,J\}} \alpha^k A(\mathbf{s}_t^k)$. At test time for each patch t an abnormal activity is identified by computing $\mathcal{A}(\mathbf{s}_t^k)$ and comparing it with a threshold η, *i.e.* $\mathcal{A}(\mathbf{s}_t^k) > \eta$ denotes an anomalous event. The weights α^k are meant to reflect the importance of different feature representations, corresponding to different one-class SVM models. While many choices are possible to learn α^k, in this paper we propose to solve the following optimization problem:

$$\min_{\mathbf{P}^k\in\mathcal{P},\alpha^k\geq 0} \quad -\sum_k \alpha^k \mathrm{tr}\Big(\mathbf{P}^k\mathbf{S}^k\big(\mathbf{P}^k\mathbf{S}^k\big)^T\Big) + \lambda_s \|\boldsymbol{\alpha}\|_p^p \tag{7}$$

where λ_s is a regularization parameter and $\mathcal{P} = \{\mathbf{P} : \mathbf{P}\mathbf{P}^T = \mathbf{I}\}$. Similarly to Principal Component Analysis, the matrix $\mathbf{P}^k \in \boldsymbol{R}^{m\times M}$, $m << M$, maps the samples $\mathbf{s}_i^k \in \boldsymbol{R}^M$ associated to the k-th modality into a new subspace in order to maximize the variance of the first m-components, subject to orthogonality constraints. The matrix $\mathbf{P}^k\mathbf{S}^k\left(\mathbf{P}^k\mathbf{S}^k\right)^T$ represents the covariance of k-th feature type in the new subspace and measures the spread of the projected samples for each modality. Setting the weights α^k by solving the optimization problem (7) we favor feature types associated with data sets with smaller variance: our intuition is that scattered data sets correspond to noisy features which must be deemphasized.

In the proposed optimization problem (7) we also introduce a ℓ_p-norm term, which, compared with traditional ℓ_2-norm and ℓ_1-norm terms, guarantees an enhanced flexibility, by allowing to tune for p [46, 47]. Intuitively, ℓ_1-norm imposes sparsity on the learned weights, while ℓ_2 norms produces an "averaging" effect. Setting *a priori* one of the two may be suboptimal in term of performance. Moreover, the complexity of solving the problem (7) with ℓ_p-norm is the same as for ℓ_2-norm [48]. Therefore, in our experiments, we tune the parameter p with the interval of 0.1 from [1.1, 1.2, ... ,2.5]. We also set the parameter $m = 100$, as it empirically provides the best performance. The projection matrices $\mathbf{P}_k$ are introduced for learning the feature weights as explained above and they are not used in the test phase. The same value $m = 100$ is chosen for the three feature types. The proposed optimization problem (7) is a convex problem and if $p > 1$ an alternating minimization algorithm can be used to solve it with respect to $\mathbf{P}^k$ and α respectively [48]. Finally,

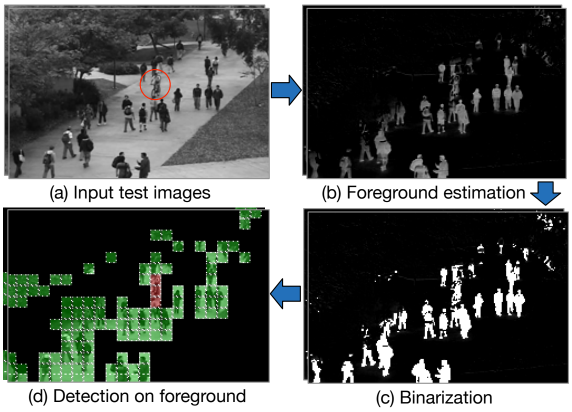


Figure 3: Illustration of the proposed foreground detection scheme using background subtraction for improving computational efficiency in the test phase. Green patches are selected foreground regions, red ones correspond to anomalies.

it is worth noting that, while the proposed late fusion scheme is applied to AMDN considering three underlying SDAEs, our strategy is general and can be used also in case of a different number of models.

4. Experiments

In this section we evaluate the performance of the proposed AMDN framework for abnormal event detection on three challenging video surveillance datasets. Specifically, we consider the UCSD pedestrian anomaly dataset (Ped1 and Ped2) [3], the Subway dataset [49] and the Train dataset [34].

4.1. Datasets

The **UCSD pedestrian** dataset [3] includes two subsets: Ped1 and Ped2[1]. The video sequences depict different crowded scenes and anomalies include bicycles, vehicles, skateboarders and wheelchairs. In some frames the anomalies occur at multiple locations. Ped1 has 34 training and 16 test image sequences with about 3,400 anomalous and 5,500 normal frames, and the image resolution is 238 × 158 pixels. Ped2 has 16 training and 12 test image sequences with about 1,652 anomalous and 346 normal frames. The image resolution is 360 × 240 pixels.

The **Subway** dataset [49] is collected using CCTV cameras and consists of two video streams corresponding to two different subway station scenarios (an entrance and an exit gate). The length of the videos is 96 min and 43 min, respectively. In the entrance subset, there are 66 abnormal events including people moving in a wrong direction, unusual gesture interactions between people and sudden stopping or running. In the exit subset 19 abnormal events are included, such as people moving in a wrong direction and loitering near the exit gate. The image resolution is 512 × 384 pixels.

The **Train** dataset [34] depicts moving people inside a train[2]. The dataset consists of 19218 frames, and the anomalous events are mainly due to unusual movements of people on the train. This is a challenging abnormal event detection dataset due to dynamic illumination changes and camera shake problems.

4.2. Implementation details

The proposed method is mainly implemented in Matlab and C++ based on the Caffe framework [50]. The code for optical flow calculation is written in C++ and wrapped with Matlab mex for computational efficiency [51]. For one-class SVMs, we use the LIBSVM library (version 3.2) [52]. The experiments are carried out on a PC with a middle-level graphics card (NVIDIA Quadro K4000) and a multi-core 2.1 GHz CPU with 32 GB memory.

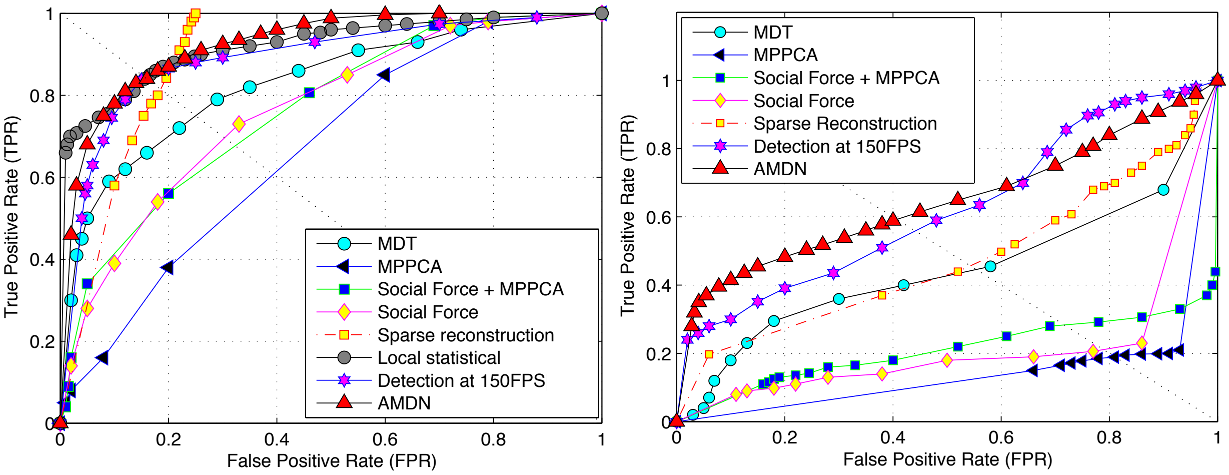
To improve the computational speed of our framework in the test phase, in this paper we introduce a foreground detection approach based on background subtraction. This is motivated by the fact that abnormal events are typically found in correspondence of moving pixels. An illustration of the proposed foreground detection scheme is provided in Fig. 3. For an input test image, the probability map of the foreground pixels is estimated with a background subtraction algorithm and binarized. The foreground regions are detected by identifying the patches which contains more than a certain number of foreground pixels (10% of the patch size in our test). We use the ViBe [53] method to perform background subtraction. ViBe has a low computational complexity and can obtain near real-time performance (almost 16 frames/second with a resolution of 360×240 in our Matlab environment).

4.3. Evaluation on the UCSD pedestrian dataset

In the first series of experiments we evaluate the performance of the proposed method on the UCSD dataset. For learning appearance features, patches are extracted using a sliding window approach at three different scales, *i.e.* 15 × 15, 18 × 18 and 20 × 20 pixels.

[1] `http://www.svcl.ucsd.edu/projects/anomaly/dataset.html`

[2] `http://vision.eecs.yorku.ca/research/anomalous-behaviour-data/`



(a) Frame-level ROC curve

(b) Pixel-level ROC curve

Figure 4: UCSD dataset (Ped1 sequence): comparison of different methods.

Table 2: UCSD dataset: comparison of different feature fusion schemes in terms of EER and AUC.

Method	Ped1(frame)		Ped1(pixel)		Ped2	
	EER	AUC	EER	AUC	EER	AUC
Joint representation (early fusion)	22.0%	84.9%	47.1%	57.8%	24.0%	81.5%
Fusion of appearance and motion pipelines (late fusion)	18.0%	89.1%	43.6%	62.1%	19.0%	87.3%
AMDN (double fusion)	16.0%	92.1%	40.1%	67.2%	17.0%	90.8%

Table 1: UCSD dataset: comparison (AUC) with the state of the art methods.

Method	Ped1(frame)	Ped1(pixel)	Ped2
MPPCA [37]	59.0%	20.5%	69.3%
Social force [9]	67.5%	19.7%	55.6%
Social force+MPPCA [3]	66.8%	21.3%	61.3%
Sparse reconstruction [4]	–	45.3%	–
Mixture dynamic texture [3]	81.8%	44.1%	82.9%
Local Statistical Aggregates [2]	92.7%	–	–
Detection at 150 FPS [1]	91.8%	63.8%	–
AMDN	**92.1%**	**67.2%**	**90.8%**

This generates more than 50 million image patches, 10 million of which are randomly sampled and warped into the same size ($w_a \times h_a = 15 \times 15$ pixels) for training. For learning the motion representation, the patch size is fixed to $w_m \times h_m = 15 \times 15$ pixels, and 6 million training patches are randomly sampled. In the test phase, we use a sliding widow approach with stride $d = 15$ and consider patches with size 15×15. The number of neurons of the first layer of the appearance and motion network is both set to 1024, while for the joint pipeline is 2048. Then, for the appearance and motion SDAE the structure of the encoder part can be simply defined as $1024 \rightarrow 512 \rightarrow 256 \rightarrow 128$. The decoder part has a symmetric structure. Similarly, for the joint SDAE the structure of the encoder part is $2048 \rightarrow 1024 \rightarrow 512 \rightarrow 256$. For the pretraining of the SDAEs, the corrupted inputs are produced by adding a Gaussian noise with variance 0.0003. The network fine-tuning is based on stochastic gradient descent with the momentum parameter set to 0.9. We set the parameters $\lambda = 0.01$, $\lambda_F = 0.0001$ and the mini-batch size $N_b = 256$. For one-class SVMs, the parameter μ is tuned with cross validation.

To perform a quantitative evaluation, we use both a frame-level ground truth and a pixel-level ground truth. The frame-level ground truth indicates whether one or more anomalies occur in a test frame. The pixel-level ground truth is used to assess the anomaly localization performance. If the detected anomaly region overlaps more than 40% with the annotated region, it is considered a true detection. We carry out a frame-level evaluation on both Ped1 and Ped2. Ped1 also provides 10 test image sequences with pixel-level ground truth. The pixel-level evaluation is performed on these sequences.

The proposed approach is compared with several state of the art methods. Specifically, we consider the Mixture of Probabilistic Principal Component Analyz-

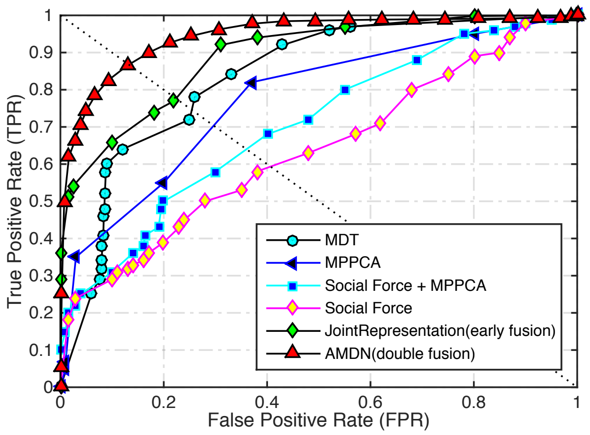


Figure 5: UCSD dataset (Ped2 sequence): comparison of frame-level performance (ROC curve) with different methods.

ers (MPPCA) approach in [37], the social force model in [9] and its extension in [3], the sparse reconstruction method in [4], mixture of dynamic texture (MDT) [3], Local Statistical Aggregates [2] and detection at 150 FPS [1].

Table 1 and Fig.6 show a quantitative comparison of different methods respectively in terms of Area Under Curve (AUC) and Equal Error Rate (EER). Figure 5 and Fig. 4 (a) and (b) report the associated ROC curves. The ROC curves are produced by varying the threshold parameter η. The performance of the baseline methods are taken from the original papers (when available). From the frame-level evaluation, it is evident that our method outperforms most previous approaches and that its performance are competitive with the best two baselines [2] and [1]. Moreover, considering pixel-level evaluation, *i.e.* accuracy in anomaly localization, our method outperforms all the competing approaches.

Table 2 demonstrates the advantages of the proposed double fusion strategy, comparing our AMDN with early fusion and late fusion approaches. Specifically, for early fusion we only consider the learned joint appearance/motion representation and a single one-class SVM. For late fusion we use the two separate appearance and motion pipelines and the proposed fusion scheme but we discard the joint representation pipeline. We observe that the late fusion strategy outperforms the early fusion and that the combination of the two schemes lead to a clear advantage. Finally, we also report some examples of anomalous events detected with our method on the UCSD dataset in Fig. 10.

4.4. Evaluation on the Subway dataset

To train the AMDN network, we follow previous works [37, 1] and use the first 15 minutes of the video sequences. The rest of the videos is used for testing. The frames are resized to a pixel resolution of 320 × 240 for computational efficiency. In these experiments the patch size for learning both appearance and motion features is 15 × 15 pixels. For SDAEs we use the same network configuration and training parameters of the experiments on the UCSD dataset. As baseline methods we consider recent approaches adopting the same dataset, including Spatio-Temporal Composition (STC) [33], MPPCA [37], Spatio-Temporal Oriented Energy (STOE) [34], Dynamic Sparse Coding (DSC) [54], Sparse Reconstruction [4] and Local Optical Flow [49].

Table 3 provides the results of the comparison of AMDN with other baseline methods. AMDN obtains

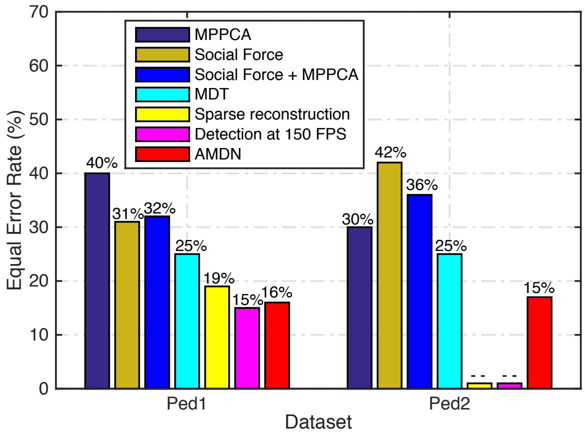
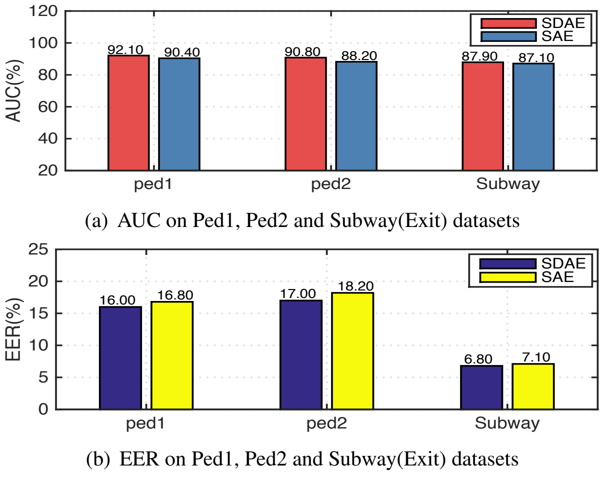


Figure 6: UCSD dataset: comparison of frame-level performance (Equal Error Rate) with different methods. Note that for the sparse reconstruction method [4] and the detection at 150 fps [1] performance on Ped2 dataset are not available.

Table 3: Comparison of different methods on the Subway dataset. In the third column, the first number denotes the detected anomalous events, while the second is the actual number of anomalous events.

Method	Dataset	Abnormal events	False alarm
STC [33]	Entrance	60/66	4
	Exit	19/19	2
MPPCA [37]	Entrance	57/66	6
	Exit	19/19	3
DSC [54]	Entrance	60/66	5
	Exit	–	–
Sparse reconstruction [4]	Entrance	27/31	4
	Exit	19/19	3
Local optical flow [49]	Entrance	27/31	4
	Exit	19/19	3
Joint representation (early fusion)	Entrance	56/66	8
	Exit	15/19	4
Fusion of appearance & motion (late fusion)	Entrance	58/66	6
	Exit	17/19	2
AMDN (double fusion)	Entrance	61/66	4
	Exit	19/19	1



(a) AUC on Ped1, Ped2 and Subway(Exit) datasets

(b) EER on Ped1, Ped2 and Subway(Exit) datasets

Figure 7: Performance comparison of SDAE and SAE when embedded in our framework.

the best detection performance from both the perspectives of abnormal events detection (61/66) and false alarm (1). The effectiveness of the proposed double fusion scheme is also verified in Table 3. Our AMDN outperforms both early fusion and late fusion techniques. Figure 11 shows some examples of the detected abnormal events, such as people entering through the exit gate, people entering without payment and people exiting from the entrance gate.

4.5. Evaluation on the Train dataset

In the third series of experiments we consider the Train dataset. The frames of the dataset are resized to 280 × 380 pixels for computational efficiency. For the AMDN parameters, we use the same experimental setting of the UCSD dataset experiments, except from the parameters λ_F and N_b which are set to 0.00001 and 100, respectively.

We compare the proposed approach with several methods in the literature considering the same datasets, including Spatio-Temporal Composition [33], Spatio-Temporal Oriented Energies [34], Local Optical Flow [49], Behavior Templates [55] and Mixture of Gaussian. From the precision/recall curve shown in Fig. 12 (c), it is clear that our method outperforms all the baselines.

4.6. Analysis of AMDN

In this section, we further analyze the proposed method to underline the importance of its main components. Fist of all, in order to demonstrate the effectiveness of the adopted SDAEs, we replace the stacked denoising autoencoder structure with a regular stacked autoencoder (SAE) into AMDN. Figure 7 shows the results of our comparison. Using SDAE we obtain a slight improvement in terms of AUC and EER on three datasets. We believe that the corrupted training data used with SDAE helps to learn more effective feature representations, as the corruption increases the variability of training data. This in line with what observed in previous works on deep architectures.

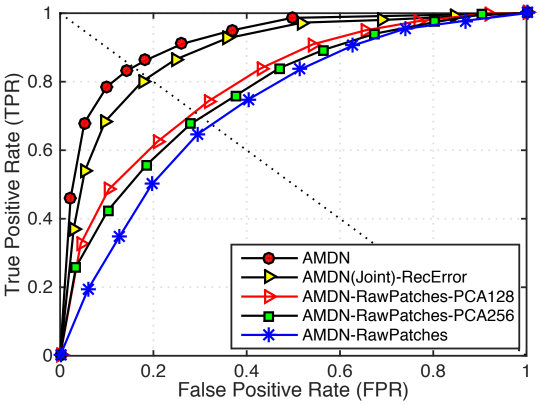


Figure 8: Comparison of different feature representations in AMDN.

To further demonstrate the validity of the learned deep representations, we evaluate the performance of AMDN by using different settings, including (i) AMDN-RawPatches: removing SDAEs from our framework and directly input patches to one-class SVMs; (ii) AMDN-RawPatches-PCA256: the same setting as (i) but using PCA for reducing the dimension of the patches to 256; and (iii) AMDN-RawPatches-PCA128: the same setting as (i) but using PCA for reducing the dimension of the patches to 128. Figure 8 compares the performance of AMDN with the different settings on the UCSD ped1 dataset. It is clear that the performance of AMDN-RawPatches is the worst among all the baselines. By using PCA on raw patches the performance is slightly improved, while AMDN is significantly better than all the baseline methods, thus demonstrating the effectiveness of the learned deep representations. We also report the TPR and FPR computed considering as anomaly score the reconstruction error of the SDAE model which uses the joint motion and appearance representation (denoted as AMDM (Joint) RecError). As shown in the figure, the proposed approach outperforms this model. We believe that it is probably because the simple reconstruction error is sensitive for identifying the anomaly regions. These results thus confirm the benefit of adopting one-class SVMs for anomaly detection in combination with a late fusion

Table 4: Comparison of different methods in terms of computational time during test (seconds per frame).

Method	Platform	CPU	GPU	Memory	Running time			
					UCSD-Ped1	UCSD-Ped2	Subway (exit)	Train
Mixture dynamic texture [3]	-	3.0 GHz	-	2.0 GB	25	-	-	-
Sparse reconstruction [4]	MATLAB	2.6 GHz	-	2.0 GB	3.8	-	4.6	-
Detection at 150 FPS [1]	MATLAB	3.4 GHz	-	8.0 GB	0.00697	-	0.00641	-
AMDN without foreground detection	MATLAB	2.1 GHz	Nvidia Quadro K4000	32 GB	9.4	13.5	12.8	14.2
AMDN with foreground detection					5.2	7.5	6.3	8.8

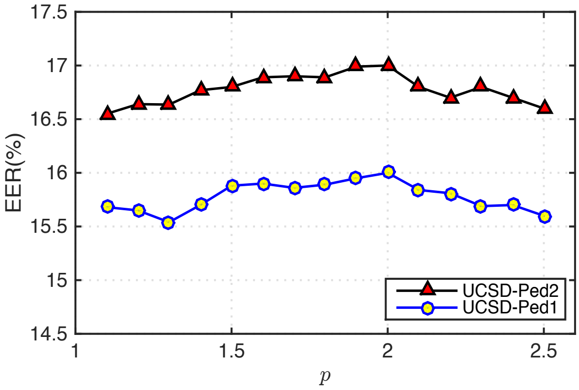


Figure 9: Performance at varying p.

scheme.

Table 5 compares AMDN with another method based on deep representations [56]. As shown in the table, the proposed framework outperforms [56] in the Ped1, Ped2 and Subway exit datasets. The approach in [56] is a frame-based deep learning method, which uses a convolution-deconvolution autoencoder network to learn a representation of the whole frame. We believe that our approach is more successful for the anomaly detection task, as it operates on a patch-level basis.

Finally, to evaluate the influence of the ℓ_p-norm in the flexible late fusion scheme, we perform a sensitivity analysis of the parameter p. Figure 9 shows the EER at varying p on UCSD Ped1 and Ped2 datasets. It is immediate to observe that in this case when p is set to 2, AMDN yields the best performance. Moreover, changing the parameter p around 2 only have a small effect on the final performance.

4.7. Computational Cost Analysis

To evaluate the performance of our approach in terms of computational cost, we conduct an empirical analysis on the considered datasets. As discussed above, different datasets have different image resolutions which affect the time required to process each frame. Specifically, the resolutions are 238×158, 360×240, 320×240 and 280 × 380 pixels, for the UCSD Ped1, UCSD Ped2, Subway and Train datasets, respectively.

Table 5: Performance comparison of different deep learning based methods in terms of EER and AUC.

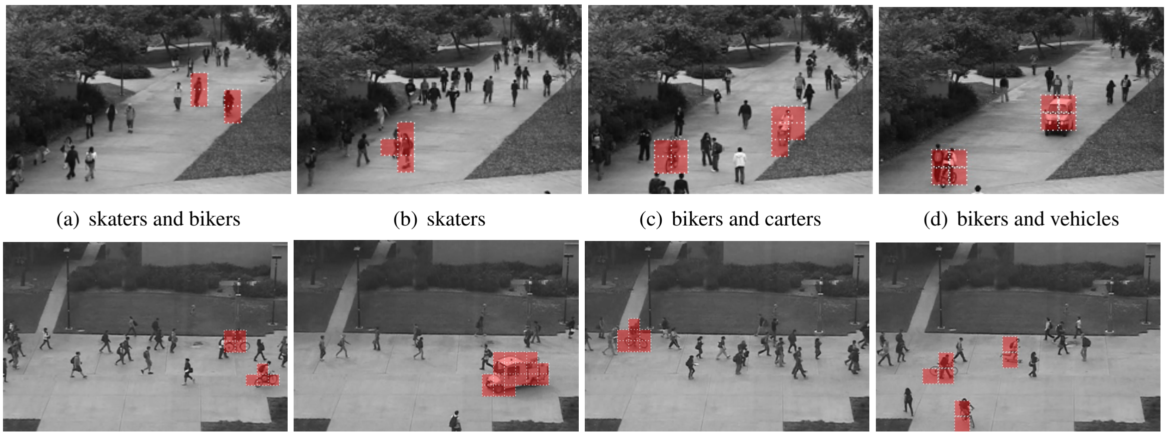
Method	Ped1		Ped2		Subway (exit)	
	EER	AUC	EER	AUC	EER	AUC
Conv-AE [56]	27.9%	81.0%	21.7%	90.0%	9.9%	80.7%
AMDN	16.0%	92.1%	17.0%	90.8%	6.8%	87.9%

Table 6: Impact of foreground detection (FD) on performance.

Method	Ped1		Ped2		Subway (exit)	
	EER	AUC	EER	AUC	EER	AUC
AMDN without FD	16.5%	91.4%	16.7%	89.6%	6.6%	88.2%
AMDN with FD	16.0%	92.1%	17.0%	90.8%	6.8%	87.9%

During the training phase, the learning of the proposed AMDN takes about 9, 4, 2.5 and 3.5 hours on the UCSD Ped1, UCSD Ped2, Subway and Train datasets, respectively. Table 4 shows the average running time of each frame during the test phase. Processing one frame to detect anomalous events takes about 9 − 14 seconds when the cell size is 15 × 15 pixels and no foreground detection is applied. Adopting the proposed foreground detection scheme (Subsection 4.2), a significant improvement in terms of computational speed is achieved (average improvement ∼ 44.3%). For sake of completeness, we also analyze the impact of the foreground detection approach on the performance in terms of EER and AUC. As shown in Table 6, no significant variations are observed. This is probably due to the fact that most false positive detections correspond to the foreground region.

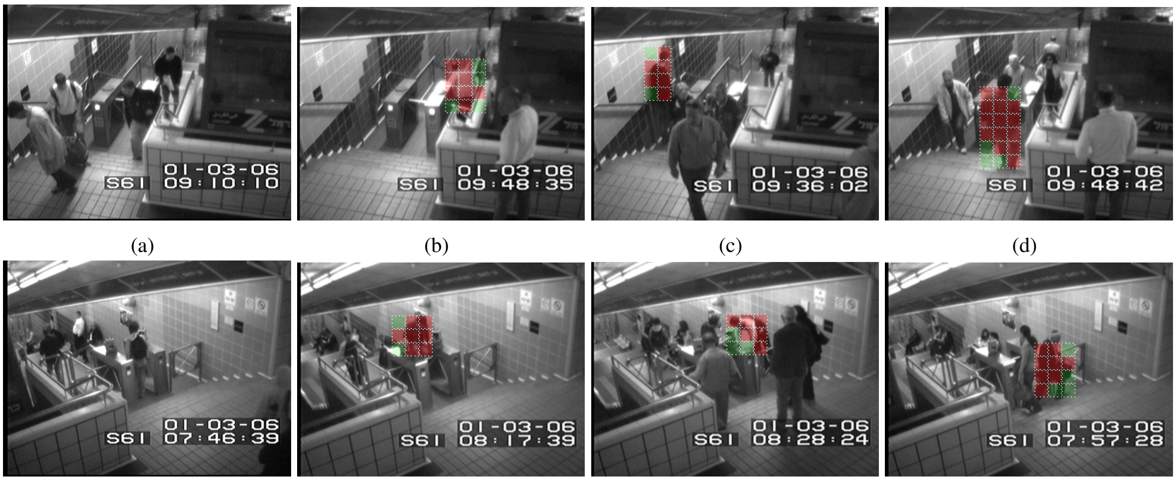
Finally, in Table 4 we also provide a comparison with some previous methods in terms of computational cost during test time. Since the original implementations of baseline approaches are not publicly available, we report the running times taken from [1] specifying the working environment. The methods in [1] and [3] are advantageous with respect to our approach in terms of detection speed. This is somehow expected as, similarly to other applications, the gain in terms of accuracy obtained with deep architectures comes at a price of an increased computational cost. Future works will be devoted to address this issue.



(a) skaters and bikers (b) skaters (c) bikers and carters (d) bikers and vehicles

(e) bikers (f) vehicles (g) bikers (h) bikers and skaters

Figure 10: Examples of anomaly detection results on Ped1 (top) and Ped2 (bottom) sequences.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 11: Examples of anomaly detection results on the Subway exit (top) and entrance (bottom) datasets. The regions with abnormal events are marked with red color. (a) and (e) show examples of normal frames of the exit and entrance scenarios. The detected anomalies in the examples include: people entering through the exit gate shown in (b), (c) and (d); people entering without payment shown in (f) and (g); people exiting through the entrance gate shown in (h).

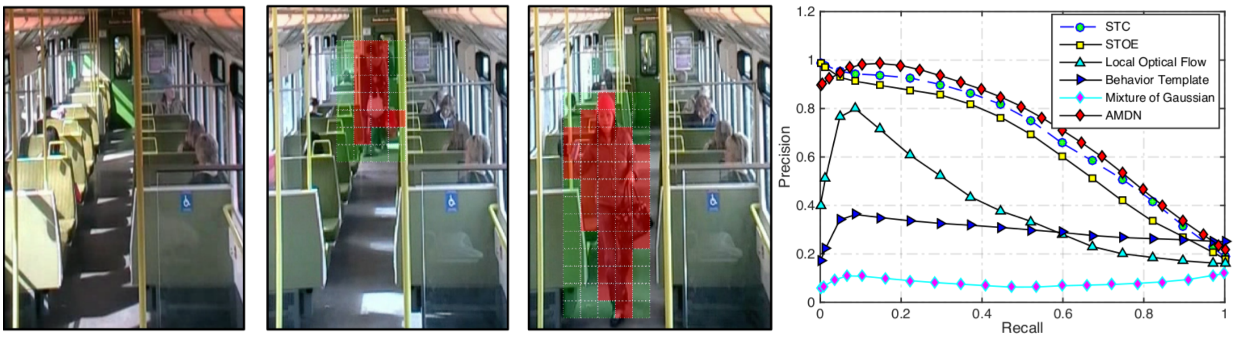


Figure 12: Results of anomaly detection on the Train dataset: (left) a frame depicting typical normal activities; (center) examples of detected anomalies; (right) precision/recall curve.

5. Conclusions

This paper introduced a novel unsupervised learning approach for video anomaly detection based on deep learning architectures. The proposed AMDN method is based on multiple SDAEs for learning both appearance and motion representations of activities in a video scene. A double fusion scheme is designed to combine the learned feature representations. We carried out an extensive experimental evaluation, considering three challenging publicly available video anomaly detection datasets (UCSD, Subway and Train), and we demonstrated the effectiveness and robustness of the proposed approach, showing competitive performance with respect to existing methods. The fundamental advantages of our approach are that it does not rely on any prior knowledge for designing features (the input of our framework are raw pixels) and does not require any object-level analysis (*e.g.* object detection or tracking). From the experimental results, it is obvious that our learned deep features are more powerful than traditional hand-crafted descriptors for representing the dynamic of the video scenes.

Currently, the computational overhead of AMDN in the test phase is too high for real-time processing. Strategies to reduce the computational cost will be studied in future works. Further research directions will include investigating other deep network architectures as well as alternative approaches for fusing data of multiple modalities in the context of SDAEs. Moreoever, an interesting follow-up of this work will be to extend ADMN in order to include contextual information. In fact, while at the present anomalous behaviours are detected by considering patches in isolation, it will be beneficial to look at co-occurrence of multiple patterns to spot additional unusual events.

Acknowledgments

This work was partially supported by the MIUR Cluster project Active Aging at Home, the EC H2020 project ACANTO. The authors also would like to thank NVIDIA for GPU donation.

References

[1] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 fps in matlab, in: ICCV, 2013.

[2] V. Saligrama, Z. Chen, Video anomaly detection based on local statistical aggregates, in: CVPR, 2012.

[3] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: CVPR, 2010.

[4] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: CVPR, 2011.

[5] Z. Fu, W. Hu, T. Tan, Similarity based vehicle trajectory clustering and anomaly detection, in: ICIP, 2005.

[6] C. Piciarelli, C. Micheloni, G. Foresti, Trajectory-based anomalous event detection, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1544–1554.

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

[8] X. Wang, K. Tieu, E. Grimson, Learning semantic scene models by trajectory analysis, in: ECCV, 2006.

[9] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: CVPR, 2009.

[10] Y. Benezeth, P. Jodoin, V. Saligrama, C. Rosenberger, Abnormal events detection based on spatio-temporal co-occurences, in: CVPR, 2009.

[11] T. Hospedales, S. Gong, T. Xiang, Video behaviour mining using a dynamic topic model, IJCV 98 (3) (2012) 303–323.

[12] V. Reddy, C. Sanderson, B. Lovell, Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture, in: CVPRW, 2011.

[13] E. Ricci, G. Zen, N. Sebe, S. Messelodi, A prototype learning framework using emd: Application to complex scenes analysis, IEEE TPAMI 35 (3) (2013) 513–526.

[14] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012.

[15] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: CVPR, 2014.

[16] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: NIPS, 2014.

[17] P. Vincent, H. Larochelle, Y. Bengio, P. Manzagol, Extracting and composing robust features with denoising autoencoders, in: ICML, 2008.

[18] N. Wang, D. Yeung, Learning a deep compact image representation for visual tracking, in: NIPS, 2013.

[19] J. Zhang, S. Shan, M. Kan, X. Chen, Coarse-to-fine autoencoder networks (cfan) for real-time face alignment, in: ECCV, 2014.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, JMLR 11 (2010) 3371–3408.

[21] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, Multimodal deep learning, in: ICML, 2011.

[22] N. Srivastava, R. R. Salakhutdinov, Multimodal learning with deep boltzmann machines, in: NIPS, 2012.

[23] Z. Lan, L. Bao, S. Yu, W. Liu, A. G. Hauptmann, Double fusion for multimedia event detection, in: Advances in Multimedia Modeling, 2012.

[24] D. Xu, E. Ricci, Y. Yan, J. Song, N. Sebe, Learning deep representations of appearance and motion for anomalous event detection, in: BMVC, 2015.

[25] O. P. Popoola, K. Wang, Video-based abnormal human behavior recognitiona review, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (6) (2012) 865–878.

[26] G. Medioni, I. Cohen, F. Brémond, S. Hongeng, R. Nevatia, Event detection and analysis from video streams, IEEE TPAMI 23 (8) (2001) 873–889.

[27] B. Morris, M. Trivedi, Learning, modeling, and classification of vehicle track patterns from live video, IEEE Transactions on Intelligent Transportation Systems 9 (3) (2008) 425–437.

[28] C. Piciarelli, G. L. Foresti, On-line trajectory clustering for anomalous events detection, Pattern Recognition Letters 27 (15)

(2006) 1835–1842.

[29] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, IVC 14 (8) (1996) 609–615.

[30] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, S. Maybank, A system for learning statistical motion patterns, IEEE Transactions on PAMI 28 (9) (2006) 1450–1464.

[31] D. Makris, T. Ellis, Learning semantic scene models from observing activity in visual surveillance, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35 (3) (2005) 397–408.

[32] F. Jiang, Y. Wu, A. K. Katsaggelos, A dynamic hierarchical clustering method for trajectory-based unusual video event detection., IEEE TIP 18 (4) (2009) 907–913.

[33] M. Roshtkhari, M. Levine, Online dominant and anomalous behavior detection in videos, in: CVPR, 2013.

[34] R. Zaharescu, A.and Wildes, Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing, in: ECCV, 2010.

[35] T. Wang, H. Snoussi, Histograms of optical flow orientation for visual abnormal events detection, in: AVSS, 2012.

[36] K.-W. Cheng, Y.-T. Chen, W.-H. Fang, Video anomaly detection and localization using hierarchical feature representation and gaussian process regression, in: CVPR, 2015.

[37] J. Kim, K. Grauman, Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates, in: CVPR, 2009.

[38] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, in: CVPR, 2009.

[39] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation 18 (7) (2006) 1527–1554.

[40] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al., Greedy layer-wise training of deep networks, in: NIPS, 2007.

[41] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, JMLR 11 (2010) 625–660.

[42] M. Kan, S. Shan, H. Chang, X. Chen, Stacked progressive auto-encoders (spae) for face recognition across poses, in: CVPR, 2014.

[43] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471.

[44] C. Poultney, S. Chopra, Y. L. Cun, et al., Efficient learning of sparse representations with an energy-based model, in: NIPS, 2006.

[45] C. Liu, G. Wang, W. Ning, X. Lin, L. Li, Z. Liu, Anomaly detection in surveillance video using motion direction statistics, in: ICIP, 2010.

[46] F. Orabona, L. Jie, B. Caputo, Multi kernel learning with online-batch optimization, JMLR 13 (1) (2012) 227–253.

[47] P. Jawanpuria, M. Varma, S. Nath, On p-norm path following in multiple kernel learning for non-linear feature selection, in: ICML, 2014.

[48] F. Nie, H. Wang, H. Huang, C. Ding, Joint schatten p-norm and p-norm robust matrix completion for missing value recovery, in: Knowledge Information System, 2013.

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

[50] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: ACM MM, 2014.

[51] C. Liu, Beyond pixels: exploring new representations and applications for motion analysis, Ph.D. thesis, Massachusetts Institute of Technology (2009).

[52] C. Chang, C. Lin, Libsvm: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27.

[53] O. Barnich, M. Van Droogenbroeck, Vibe: A universal background subtraction algorithm for video sequences, IEEE TIP 20 (6) (2011) 1709–1724.

[54] B. Zhao, L. Fei-Fei, E. P. Xing, Online detection of unusual events in videos via dynamic sparse coding, in: CVPR, 2011.

[55] P. Jodoin, J. Konrad, V. Saligrama, Modeling background activity for behavior subtraction, in: ICDSC, 2008.

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