Abstract—In this paper, we propose an accurate and real-time anomalous behavior detection and localization in crowded scenes, and two descriptors for representing anomalous behavior in video are proposed. We consider a video as being a set of cubic patches. Based on the low likelihood of an anomaly occurrence, and the redundancy of structures in normal patches in videos, two (global and local) views are considered for modeling the video. Our algorithm has two components, for (1) representing the patches using local and global descriptors, and for (2) modeling the training patches using a new representation. We have two Gaussian models for all training patches respect to global and local descriptors. The local and global features are based on structure similarity between adjacent patches and the features that are learned in an unsupervised way. We propose a fusion strategy to combine the two descriptors as the output of our system. Experimental results show that our algorithm performs like a state-of-the-art method on several standard datasets, but even more time-efficient.

Index Terms—Anomaly detection, Feature representation, fusion strategy, global and local descriptors.

I. INTRODUCTION

ANOMALY behavior detection in crowd scenes is one of the challenging problem in computer vision field. There are different definitions of an anomaly that appears in a video, depending on contexts. An event is an anomaly if it has a low likelihood of occurrence [6]. To describe unusual events in complex scenes, a high-dimensional model must be applied and enormous of training samples must be provided. In real-world anomaly problems, the number of training samples are limited. Therefore, this might face the so-called “curse of dimensionality”, in which the predictive power of the trained model reduces, as the dimensionality of the feature descriptors increases. Generally, developing a reliable model to be trained with high-dimensional without enormous of training samples is quite challenging.

In recent work, the reference (One or a set of) normal model is learned from training videos and then applied for detecting the anomaly event in a test phase. They consider a test video as being an anomaly if it does not belong to the learned models, generated to represent the video and to create a reference model from it what the researchers defined based on some features. In general, these features can be divided into two groups which are used for representing the (1) trajectory or (2) spatial-temporal changes. The works reported in [8], [19], [25]–[31] focus on the trajectories of objects in videos; authors labeled each object as anomaly that does not follow the learned normal trajectory; these method cannot handle the occlusion problem, and are also computationally very expensive for crowded scenes.

To overcome these weaknesses, researchers proposed some methods using low-level features such as optical flow or gradients. Their methods are based on learning the shape and spatial-temporal relations using low-level features distributions. [13] fits a Gaussian mixture model to features. [1] uses an exponential distribution. Clustering of test data using low-level features is exploited in [16]. In [2], [9], [10], [22], the normal patterns are fitted to a Markov random field, and [14], [18] apply latent Dirichlet allocations. [11] introduces a joint detector of temporal and spatial anomalies; the authors use a Mixture of Dynamic Textures (MDT) model. Yun and et al. in [35] introduces an informative Structural Context Descriptor (SCD) to represent the crowd individual, in their work, SCD variation (spatial-temporal) of the crowd, is analyzed to localize the anomaly region.

In [34] a method based on spatial-temporal oriented energy filtering is proposed. Boiman and Irani [53] use a reconstruction method for anomaly detection, they label a new observation as an anomaly, if it cannot be reconstructed using previous observations, a similar framework is proposed in [6] based on the learning a dictionary including main basis of normal events.

Today, sparse representations of events [6], [7], [12] in videos can be considered as the state of the art. References [6], [7], [11], [12], [14], [15], [32] achieved a good performance in anomaly detection, but their accuracy in anomaly localization is low, and most of them (except few number of them) are not capable to run in real-time on real-world anomaly problems. In [24] a context-aware anomaly detection algorithm is proposed, in which the authors represent the video using motions and context of videos.

In our previous work [23], we analyze the video using a patch-based method, and the video patches are represented in two levels: (1) Local, (2) Global. If a patch is considered as an anomaly in both levels, it is considered as being as anomaly. Similar to [23], in this paper, to overcome the weaknesses of previous work and to meet high performance, videos are represented with two different views (thus two partially independent feature sets). We propose an approach for integrating these views in a testing step to improve anomaly detection and
The overall scheme of our algorithm is shown in Fig. 2.

The main contributions are as follows:

1) Applying feature learning into video anomaly localization. This method is time-consuming for training, but the learned features are very discriminative to model the normal patches.
2) Introducing a descriptor-based similarity between patches and their adjacent patches for detecting sudden changes in spatial-temporal domains.
3) Representing video patches with two views. The local and global feature sets are used for each view. In the final decision, these views support each other.
4) Modeling all normal patches with Gaussian distributions. For a test video, the Mahalanobis distance is used to figure out its relevance for the normal patches.
5) Introducing a new measure for evaluating the anomaly localization performance.
6) Being real-time.

The overall scheme of our algorithm is shown in Fig. 2.

III. PROPOSED SYSTEM

A. Overall Scheme

We convert the video into cubic patches; a sketch of this video representation is shown in Figure 1.

Generally, every video has dominant events. Thus, the normal patches must have the same relation with their adjacent patches and a high likelihood of occurrence in the video. With considering these criteria, the anomaly patches satisfy three conditions:

1) The similarity between anomaly patches with their adjacent (i.e. defined by spatial changes) patches which do not follow the same pattern as from normal patches to their adjacent patches.
2) Temporal changes of an anomaly patch do not follow the pattern of temporal changes of normal patches.
3) The occurrence likelihood of an anomaly patch is less than that of normal patches.

Conditions (1) and (2) are related to local features, and Condition (3) is related to a global one. In other words, Conditions (1) and (2) consider the relation between a patch and its adjacent ones, and Condition (3) describes the overall appearance of patches in the video. Because Conditions (1) and (2) are related to spatial-temporal changes, and Condition (3) differs from it, we model a combination of (1) and (2) as local views, and (3) as a global view. To avoid the “curse of dimensionality”, we model the views separately.

To achieve a good performance regarding the true-positive and false-positive rates, we include initially in the models that we have large numbers of false-positives as well as many high true-positives. In a final decision, if both view strongly (or one strongly and another weakly) reject a patch, it is considered to be an anomaly. (The verification results are categorized in three levels: Strongly accept, Weakly (accept/reject), strongly reject.)

Since, the sets of samples do not coincide that are initially classified as being false-positive in the first or in the second model, thus the number of final true-positives decreases (i.e. is equal to the number of samples being false-positive in both models), but the number of final true-positives is equal or smaller than the minimum number of true-positives in both models. An anomaly patch must satisfy Conditions (1), (2), and (3). Thus, by combining both models we expect to achieve a good performance regarding true-positives. This procedure is explained in more details in section III-D.

The input data are represented by two different views. Then, these representations are fitted to Gaussian distributions to compute the decision boundary of each of them. Finally, we use the decision table (based on global and local results) for anomaly patch detection. The two sets of features (global and local) will be introduced in the next section.

B. Global descriptor.

The global descriptor is the feature set that can describe the normal video patches. In [21] it is argued that classical handcrafted low-level features, such as HOG and HOF, may not be universally suitable and discriminative enough for every type of video. So, unlike previous work that uses low-level
features, we use an unsupervised feature learning method based on an auto-encoder. The structure of the auto-encoder is shown in Fig. 3. The auto-encoder learns sparse features based on gradient descent as a neural network. Suppose that we have $m$ normal patches with $(w, h, t)$ dimensions $x_i \in \mathbb{R}^{D = w \times h \times t}$ (the raw data) for learning features from them; the auto-encoder attempts to minimize Eqn. (1) by reconstruction the raw data:

$$L = \frac{1}{m} \sum_{i=1}^{m} \left\| x_i - W_2 \delta(W_1 x_i + b_1) + b_2 \right\|^2 + \sum_{i=1}^{w \times h \times t} s \sum_{j=1}^{s} (W_2^2)_{ji} + \beta \sum_{j=1}^{s} KL(\rho||\rho_j')$$

where $s$ is the number of nodes in the auto-encoder’s hidden layer, $W_1 \in \mathbb{R}^{s \times D}$ and $W_2 \in \mathbb{R}^{D \times s}$ are the weight matrices, which map the input layer nodes to hidden layer nodes, and hidden layer nodes to the output layer nodes, respectively. $W_{ji}$ is the weight between the $j^{th}$ hidden layer node and the $i^{th}$ output layer node, and $\delta$ is equal to the sigmoid function. Furthermore, $b_1$ and $b_2$, are the bias of the output layer and the hidden layer, respectively. $KL(\rho||\rho_j')$ is a regularization function and is set to enforce the activation of the hidden layer to be sparse. $KL$ is based on the similarity between a Bernoulli distribution with $\rho$ as parameter, and the active node distribution. The parameter $\beta$ is the weight of the penalty term (in the sparse auto-encoder objective). We can efficiently optimize the above objective with respect to $W_1$ via the stochastic gradient descent approach.

C. Local descriptor

We introduce a descriptor based on local features. The aim of local descriptor is modeling the relation each patches with neighboring patches. So, to construct this descriptor, the similarity between each patch with its neighbors is considered. We use structural similarity(SSIM) for computing the similarity between two patches; SSIM is one of the image-quality assessment methods [4], [36] and the main version of it, can be used to calculate the structural similarity between two images.

The SSIM is based on three components which are related to contrast, illumination and structure of two images. if X and

![Diagram](image-url)

Fig. 3. Summary for learning the global features using an auto-encoder. Left: The step for learning features uses raw normal patches; components (1), (2), (3), (4), and (5) are needed; the aim is to reconstruct the input paths with adjusting $W_1$ and $W_2$ using gradient descent. Middle: Auto-encoder structure. Right: Representing the $y$ patch using the $W_1$ weights $(y \times W_1)$; (1), (2), and (3) are just used; this is a multiplication of two matrixes, so it is very fast.
Y be two images, the similarity between X and Y images is calculated based on Equ. (9).

\[
SSIM(X, Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}
\]  

(2)

Where \(\mu_x, \mu_y, \sigma_x\) and \(\sigma_y\) are the mean of X, the mean of Y, the variance of X and the variance of Y, respectively. \(C_1\) and \(C_2\) are small constants. The \(\sigma_{xy}\) can be calculate based on Equ. (3).

\[
\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \mu_x)(Y_i - \mu_y)
\]

(3)

where \(N\) is the size of images(X and Y must be the same size as \(N\)). We extend the SSIM for computing the similarity between two patches. Suppose we divided the video into patches with \((w, h, t)\) dimensions, t is depth of each patch(number of frames which are used). For calculating the similarity between the X and Y patches the Equ. (4) is used.

\[
SSIM_p(X, Y) = \sum_{i=1}^{t} SSIM(X^{(w\times h)}_i, Y^{(w\times h)}_i)
\]

(4)

where \(X_i\) and \(Y_i\) are the equivalent 2D patch from the \(i^{th}\) frame of X and Y patches as which are considered as an image. Figure 4 shows the calculating the similarity between two patches. Figure 5 shows spatial and temporal features. The local descriptor is a combination of these features \([d_0 \cdots d_0, D_0 \cdots D_3]\).  

D. Anomaly Classifier

Two Gaussian classifiers \(C_1\) and \(C_2\) learn normal classes. For classifying \(x'\) patches, we use two partially independent feature sets (global and local). We compute the Mahalanobis distance \(f(y)\). If \(f(y)\) is larger than the threshold then it is considered to specify an abnormal patch, where \(y\) equals \(W_1 \times x'\), in the global classifier, and \(y\) equals \([d_0 \cdots d_0, D_0 \cdots D_3]\) in the local classifier. To be more accurate we classify the patches into three classes: (1) Strongly normal, (2) Partly normal or anomaly, (3) Strongly anomaly. To avoid numerical instabilities, density estimates are avoided. The \(C_1\) and \(C_2\) classifiers are defined as follows:

\[
C_i(x) = \begin{cases} 
\text{Strongly normal} & f(x) \leq \theta^i \\
\text{Strongly anomaly} & f(x) \geq \alpha^i \\
\text{Partly (normal, anomaly)} & \text{otherwise}
\end{cases}
\]

(5)

with

\[
f(x) = (x - \mu)^T \Sigma^{-1} (x - \mu)
\]

(6)

Where \(\mu\) and \(\Sigma\) are mean and covariance matrix, respectively. Selecting a “good” threshold is important for the performance; it can be selected based on training patches. We select the thresholds(\(\theta\) and \(\alpha\)) based on training patches using Equ. (7) and Equ. (8).

\[
\arg\min_{\theta^i} \| \frac{\#(\text{Detected Normal Patches})}{\#(\text{All Normal Patches})} - \zeta \|
\]

(7)

\[
\arg\min_{\alpha^i} \| \frac{\#(\text{Detected Anomaly Patches})}{\#(\text{All Anomaly Patches})} - \zeta \|
\]

(8)

Where \#1 and \#2 are related to the number of patches which are detected in respect to local and global descriptors, respectively.

The \(\zeta \in [0.6, 1]\) is a regularization parameter which is used for adjusting the boundary of three classes. \(\zeta \times 100\%\) percent of anomaly and normal samples where have low likelihood (and can make error) are classified as partly(normal or abnormal) class. For more details see Figure 6.

In this paper, two aspects will be considered for both thresholds:

(1) Selecting overall threshold which is used for all scenes in a video, this means by changing the context of video the threshold does not change and will be constant.

(2) Selecting local threshold as it will change from a scene to another scene(or different interval time), in other word, the threshold will be updated based on the context.

Respect to generality and complexity term the overall threshold has a better performance rather than local threshold, but in comparison with accuracy term it is worse.

As we mentioned in Figure 3 patch representation based on global descriptor is very fast, but local representing is more

![Fig. 4. Computing the similarity between X and Y patches.](image-url)
Fig. 5. A local descriptor for centroid(0) patch Top: The $SSIM_p$ between the 0 with 9 adjacent patches is computed (8 patches are around the 0 patch and one patch (p) is the behind of 0 patch, exactly) Bottom: The SSIM between every frame with previous of it, is calculated. this part of descriptor is related to temporal-change.

![Diagram](image)

Fig. 6. The policy of selecting thresholds (for simplicity, one dimensional data is considered). this policy is exploited to calculate the boundary of three classes(abnormal, weak(normal, abnormal), normal). $Ω$ is computed with optimizing the $\arg \min_\alpha \| \frac{\#(Detected normal patches)}{\#(All normal patches)} - 1 \|$. (See Eqn. and Eqn. for $\alpha$ and $\theta$)

Table 1: The SSIM between every frame with previous of it, is calculated. this part of descriptor is related to temporal-change.

| SSIM | Value  |
|------|--------|
| SSIM$_p(0,1)$ | $d_0$ |
| SSIM$_p(0,2)$ | $d_1$ |
| $\vdots$ | $\vdots$ |
| SSIM$_p(0,p)$ | $d_p$ |

As mentioned before, if both $C_1$ and $C_2$ classifiers, label a patch as being an anomaly(both strongly or at least one be strongly, and other be weakly), it is considered to be an anomaly, but if either one or both of them consider the patch as being a normal(strongly), or if both of them consider a patch as being a normal(or anomaly) weakly, the algorithm classifies it as being a normal patch. A summary of these criteria is shown as $F$ function in the following equation:

$$F(x) = \begin{cases} 
\text{Anomaly} & \text{if } \sum_{i=1}^{2} \Phi(C_i) \geq 1.5 \\
\text{Normal} & \text{otherwise} \end{cases}$$

As a conclusion to this section, the anomaly detection can be presented as following steps:

1. Modeling the training samples using two (global and local) descriptors and finding the optimum parameters.
2. Labeling the test patches with respect to both Gaussian model.

The algorithm (Preprocessing: Creating Model) and (Anomaly Event Detection) show (1) and (2) as a summery of the proposed systems.

Algorithm 1 Preprocessing: Creating Model.

**Input:** N training patches: $(x^1, x^2, \ldots, x^N \in R^{D = w \times h \times t})$

**Output:** $(\mu_1, \Sigma_1)$ and $(\mu_2, \Sigma_2)$, $\alpha$ and $\beta$ thresholds.

1. $(x^1_G, x^2_G, \ldots, x^N_G \in R^{a=1000}) = G(x^1, x^2, \ldots, x^N \in R^D)$
2. $(x^1_L, x^2_L, \ldots, x^N_L \in R^{a=1000}) = L(x^1, x^2, \ldots, x^N \in R^D)$
3. $\mu_1 = \text{Mean}(x^1_G, x^2_G, \ldots, x^N_G \in R^{a=1000})$
4. $\mu_2 = \text{Mean}(x^1_L, x^2_L, \ldots, x^N_L \in R^{a=1000})$
5. $\Sigma_1 = \text{Covariance}(x^1_G, x^2_G, \ldots, x^N_G \in R^{a=1000})$
6. $\Sigma_2 = \text{Covariance}(x^1_L, x^2_L, \ldots, x^N_L \in R^{a=1000})$
7. for i=1 to 2
   a. $\arg\min_{\alpha_i} \| \frac{\#(Detected Normal patches)}{\#(All Normal patches)} - \zeta \|$
   b. $\arg\min_{\beta_i} \| \frac{\#(Detected Anomaly patches)}{\#(All Anomaly patches)} - \zeta \|$
8. end
9. return $(\mu_1, \Sigma_1), (\mu_2, \Sigma_2)$, $\alpha_1$, $\beta_1$, $\alpha_2$ and $\beta_2$ thresholds.

E. Anomaly detection using feature learning

We learn the features from raw training data, and classify the video patches as specified in the previous section (III-D). But
Algorithm 2 Anomaly Event Detection.

Input: $(\mu_1, \Sigma_1)$ and $(\mu_2, \Sigma_2)$, $\alpha^1$, $\beta^1$, $\alpha^2$ and $\beta^2$, Testing patch of $y \in R^{40 \times 5 \times 1}$.

Output: $z^*$ the label of $y$.

1. $y_G = G(y)$
2. $y_L = L(y)$
3. $\Phi_1 = \{ \begin{cases} 0 & \text{if} \ D_{\text{dis}(local)} \leq \theta^1 \\ 1 & \text{if} \ D_{\text{dis}(local)} > \alpha^1 \\ 0.5 & \text{otherwise} \end{cases}$
4. $\Phi_2 = \{ \begin{cases} 0 & \text{if} \ D_{\text{dis}(global)} \leq \theta^2 \\ 1 & \text{if} \ D_{\text{dis}(global)} > \alpha^2 \\ 0.5 & \text{otherwise} \end{cases}$
5. $z^* = \begin{cases} \text{Anomaly} & \text{if} \sum_{i=1}^{5} \Phi_i \geq 1.5 \\ \text{Normal} & \text{otherwise} \end{cases}$

return $z^*$

Based on the idea in [3], using small patches and large patches leads to increases in the false-positive rate and decreases in the true-positive rate, respectively. Also, when the patches become larger, the input dimension of the auto-encoder increases, so the number of weights in the network, which need to be learned, will also increase. Under the condition of limited training examples, learning of features from large patches is impractical (for example $40 \times 40 \times 5$). In addition to these weaknesses, the features which are learned using auto-encoder are not invariant to spatial changing.

So, to overcome these challenges (and have a trade-off between true positive and false positive performance), we learn the features from (small) $10 \times 10 \times 5$ patches. To create a model using these features, in the test phase the large patches ($40 \times 40 \times 5$) are considered. Because the learned classifier is adapted for $10 \times 10 \times 5$ patch representations, we convolve the learned feature ($W_4$) in $40 \times 40 \times 5$ patches, without overlapping, and pool the 16 extracted feature vectors from the $40 \times 40 \times 5$ patches. So, we use mean pooling to achieve a representation of $40 \times 40 \times 10$ patches that can be checked with the learned classifier using $10 \times 10 \times 5$ patches. This procedure is shown in Figure 7. In a summary the benefits of this procedure (convolving and pooling) are as follows:

1. The learned feature are invariant in respect to spatial changing.
2. The capability of representing the large patches using small patch representation.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

We compare our algorithm with state of the art methods on UCSD[1] and UMN[2] benchmarks. We empirically demonstrate that our approach can be used in surveillance system.

A. System setting

Feature learning is done with $10 \times 10 \times 5$ patches. Training and testing phases in anomaly detection is done with $10 \times 10 \times 5$ and $40 \times 40 \times 5$ patch sizes, respectively. In anomaly detection, the size $40 \times 40 \times 5$ is exploited. Feature learning is done with an auto-encoder with 0.05 sparsity. Each $10 \times 10 \times 5$ patch is represented by a 1000-dimensional feature vector (1000 are selected, experimentally). Before feature learning, normalization is performed to set the mean and variance to 0 and 1, respectively. The $\zeta$ set to be equal 0.9, as it lead to the optimum $\alpha$ and $\theta$ regarding to the best performance in true positive and false positive. All experiments are done using a PC with 3.5 GHz CPU and 8G RAM in MATLAB 2012a.

B. UCSD datasets

This dataset includes two subsets, Ped1 and Ped2, that are related to different outdoor scenes. Both of them are recorded with a static camera and 10 fps. The dominant mobile objects in this scene are walkers where the crowd density varies from low to high crowded. An object such as a car, skateboarder, wheelchair, or bicycle is considered as being an anomaly. All training frames in this dataset (i.e., Ped1 and Ped2) are normal

1. www.svcl.ucsd.edu/projects/anomaly/dataset.html
2. mha.cs.umn.edu/Movies/Crowd-Activity-All.avi
Anomaly

The relationship of them. They are defined as:

\[ TPR = \frac{True \ positive}{True \ positive + False \ negative} \]
\[ FPR = \frac{False \ positive}{False \ positive + True \ negative} \]  

C. Evaluation Methodology

We compare our results with state-of-the-art methods using a ROC curve and EER, similar to [13]. For a better understanding of ROC, True Positive Rate(TPR) and False Positive Rate(FPR) terms should be introduced because ROC reflects the relationship of them. They are defined as: 

\[ TPR = \frac{True \ positive}{True \ positive + False \ negative} \]  
\[ FPR = \frac{False \ positive}{False \ positive + True \ negative} \]  

We use two measures, at frame level and at pixel level. In addition to these measures, we define a new measure for the accuracy of anomaly localization, called dual pixel level. The measures that we use are as follows:

1) **Frame level**: In this measure, if one pixel detects an anomaly then it is considered as being an anomaly.

2) **Pixel level**: If at least 40 percent of anomaly ground truth pixels are covered by pixels that are detected by the algorithm, then the frame is considered to be an anomaly.

3) **Dual pixel level**: In this measure, a frame is considered as being an anomaly if (1) it satisfies the anomaly condition at pixel level and (2) at least \( \beta \) percent (for example, 10 percent) of pixels detected as anomaly are covered by the anomaly ground truth. If, in addition to the anomaly region, irrelevant regions are also considered as being an anomaly, then this measure does not identify the frame as being positive.

Suppose that the algorithm detects some region as being an anomaly, and just one of these regions has an overlap with anomaly ground truth; the number of false regions is not considered in the frame and pixel level measures. Such a region is called a "lucky guess". For considering the "lucky guess" region, we introduce the dual pixel level. This measure is sensitive to a "lucky guess".

Figure 8 shows an example for the different measures of anomaly detection.

D. Performance

1) **The Influence of patch size and patch overlapping**: As mentioned in previous sections, we consider the video as being a set of cubic patches. The size of patches and the level of overlapping patches are two important parameters which are related to video representing. As the final performance of system depends on these parameters, we run some experiments for concluding the optimum values. We run our algorithm with different patch sizes:20, 25, 30, 35,40. Table I shows the influence of patch size on the frame level accuracy and the algorithm’s running time. Non-overlapping patches are exploited in this experiment. The accuracy is calculated using Eq(11). This experiment is done on 4th video subsequence of Ped2. Decreasing the patch size and improving accuracy are in a direct relation, but it has inverse relation with running time performance.

| Patch size | Accuracy | Time(second/frame) |
|-----------|----------|--------------------|
| 20        | 94.0%    | 0.63               |
| 25        | 92.0%    | 0.42               |
| 30        | 87.2%    | 0.25               |
| 35        | 85%      | 0.20               |
| 40        | 50%      | 0.12               |

Accuracy = \[ \frac{True \ positive + True \ negative}{All \ Samples} \]  

We evaluate the global descriptor performance in respect to: 0, 5, 10, 15 and 20 pixels overlapping. Figure 2 shows the effect of overlapping on frame level measure. This experiment is done on all video sequences in Ped2, and are based on global descriptor, only(i. e. One Gaussian classifier with two classes(normal and abnormal) are applied).

2) **Comparison with other methods**: Depending on experiments, the patches dimension are considered equal to \((40 \times 40 \times 5)\) with 20 pixels overlapping. Some frame results on UCSD Ped1 dataset are shown in Figure 10(The abnormal events are localized with green borders). Figure 10 confirms that the proposed method is able to detect and localize anomaly events as well. Also, Figure 11 shows a qualitative comparison with other methods on UCSD Ped2, in which the 1st and 2nd...
rows is generated by temporal MDT and spatial MDT, the 3rd, 4th and 5th rows are generated by MPPCA, social force and optical flow, respectively. The two bottom rows (6th and 7th) are the results of our methods, based on the global descriptor and jointly(local and global) descriptors. In all three scenes, other methods miss the anomaly events or they detect some irrelevant regions as anomaly. The Figure 11 indicates that our algorithm has the best performance(either in anomaly detection or anomaly localization) in the set of considered algorithms.

Statistical results for comparisons are provided in this section.

Our results in comparison with the other proposed methods on Ped1 are available in Table II. The methods which are presented by Yuan et al. [35] and Xiao et al. [39] have the best results in frame level measure. Our method is, outperform(by 0.6% and 1.6%) both of them. The pixel level EER of presented method is also better than the considered approach(except [39]). Our method’s EER is 16.7% where the best result is 16%. We are the worse by 0.7%.

In Figure 12 (left), the frame-level ROC of our method(with local and overall threshold) is compared with other methods on the Ped2 dataset. It shows that our method is comparable to other methods. However, the ROC of the proposed method with local threshold is superior to overall threshold case. But, our algorithm with overall threshold work like state-of-the-art method, even better in time consuming. As we tune and set the overall threshold(α and θ) once, so it is faster in training and testing steps than local threshold. The 12 different subsequences on Ped2 do not have the same thresholds, but as we attempt to have a high true positive in each models, the minimum value of thresholds, which are calculated based on Eqn. (7) and Eqn. (8) are considered.(i.e. as a preprocessing the two thresholds for all 12 sequences are computed and the minimum of them is applied for all ). The EER of frame level for different methods on Ped2 is shown in Table III. This confirms that our method has a good performance compared with the others. The methods which are reported by Tan Xiao et al. [39] and iHOT [38] have a close result to our’s.(We are 2.5% and 1.09% better, respectively)

Figure 12 (Right) is related to pixel-level ROC on Ped2. Our methods is superior compared with the other methods as well as frame-level measure. In Table III, we compare the pixel level EER of our approach to that of other approaches. Our method’s EER is 16% where the next best result is 17% reported for the Tan Xiao et al. [39] method. Our method is 1% better than the otherwise best result. The results show (ROC and EER) that our algorithm outperforms the other methods for the pixel-level measure.

We also use a dual-pixel level measure to analyse the accuracy of anomaly localization. Figure 13 shows the effect of β on our algorithm. The algorithm has a good performance, even better than a state of the art, in pixel level with β=5% and 0% percent. Table II and III 3rd column) show the dual-pixel level measure as the β is considered equal to 0.05. The dual-pixel EER of our method on Ped1 and Ped2 are 26.2% and 25.8% where, it can be compared with the pixel-level of the other methods. Fig. 13 illustrates comparisons at frame level and dual pixel level(with β equal to 0, 5%, 10%) of our approach; in contrast to the best performance algorithms of reported algorithms, the pixel level measure is very close to frame level measure in our algorithm.
Fig. 10. Examples of anomaly detection on UCSD Ped1. (With 20 pixels overlapping with adjacent patches)

Fig. 11. Example of anomaly detection from three scenes of Ped2. First row to 7th row show Temporal MDT, Spatial MDT, MPPCA, Social force, Optic flow, Our method (feature learning only), an Our method (combined views). In 5 first rows, the anomaly event pixels are colored red, In 6th and 7th rows the anomaly are localized with green and red borders, respectively.

3) Run time comparison: We compare the average time to process one frame in Table IV. If the patches of a frame are extracted with 20% overlapping, the performance is like state of the art, and except [39] we are superior all methods. Extraction patches without overlapping leads to a faster approach as it is shown in latest row of Table IV. So, with encountering a bit error (∼5% EER) and optimizing the code, our approach is highly efficient. We reach up to 25 fps (i.e., 0.04 second for each frame).

| Method          | Ped1 | Ped2 |
|-----------------|------|------|
| ICB             | 55   | 66   |
| MDT             | 17   | 23   |
| Roshkhar et al. | 0.16 | 0.18 |
| Li et al.       | 0.65 | 0.80 |
| Tan Xiao et al. | 0.22 | 0.29 |
| SRC et al.      | 3.8  | —    |
| OADC-S          | 0.6  | 0.6  |
| Ours(20%)       | 0.51 | 0.63 |
| Ours(0%)        | 0.11 | 0.13 |

### E. Performance on UMN datasets

The UMN dataset has three different scenes. In each scene, a group of people walks in an area, suddenly all people run away (escape); the escape is considered to be the anomaly. Figure 14 shows examples of normal and abnormal frames of this dataset.

| Method          | EER  | AUC  |
|-----------------|------|------|
| Chaotic invariants [19] | 5.3  | 99.4 |
| SF [14]          | 12.6 | 94.9 |
| Sparse [6]       | 2.8  | 99.6 |
| Saligrama et al. [16] | 3.4  | 99.5 |
| Li et al. [11]   | 3.7  | 99.5 |
| Ours             | 2.5  | 99.6 |

This dataset has some limitations. There are only three anomaly scenes in the dataset, and the temporal-spatial changes between normal and abnormal frames are very high. The UMN has no pixel-level ground truth. Based on this limitations, to evaluate our method, the EER and AUC in frame-level are used. The EER and AUC results are shown in Table 3. Because this dataset is simple, and anomaly localization is not important, only the global detector is used. Previous methods performed perfectly on this dataset. The AUC of our method is comparable with the otherwise best result, and the EER of our approach is better (by 0.3 percent) than the one of the best previous method.

### V. CONCLUSIONS

In this work, we presented a patch-based anomaly detection and localization method based on representing a video using...
global and local descriptors. The relation of a patch with all other patches which are seen in video are considered a
global descriptor, also the structural similarity a patch with
neighboring patches are named:local descriptor. Two Gaussian
classifiers are proposed based on these two forms of repre-
sentations. However, each of the two classifiers has a good
performance for anomaly detection (This is especially shown
on the UMN data set where the global descriptor achieves
state-of-the-art results.), but to be more accurate, a simple
strategy is applied to fusion the results of two descriptors as
the final output of the system . We introduced a new metric
for region level anomaly detection for suspicious regions.
The performance of our approach on the UMC data set is better
compared with the recent approaches and has also a lower
computational complexity.

**APPENDIX A**

PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

**APPENDIX B**

Appendix two text goes here.

**ACKNOWLEDGMENT**

The authors would like to thank...

**REFERENCES**

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

[2] Y. Benezech, P.-M. Jodoin, V. Saligrama, and C. Rosenberger. Ab-
normal events detection based on spatio-temporal co-occurrences. In
*CVPR*, pages 1446–1453, 2009.

[3] M. Bertini, A. Del Bello, and L. Seidenari. Multi-scale and real-
time non-parametric approach for anomaly detection and localization.
*Computer Vision Image Understanding*, 116(3):320–329, 2012.

[4] D. Brunet, E. R. Vrscay, and Z. Wang. On the mathematical properties
of the structural similarity index. *IEEE Trans. Image Processing*,
21(4):1488–1499, 2012.

[5] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks
in unsupervised feature learning. In *Int. Conf. Artificial Intelligence
Statistics*, pages 215–223, 2011.

[6] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal
event detection. In *CVPR*, pages 3449–3456, 2011.

[7] Y. Cong, J. Yuan, and Y. Tang. Video anomaly search in crowded
scenes via spatio-temporal motion context. *IEEE Trans. Information
Forensics Security*, 8(10):1590–1599, 2013.

[8] F. Jiang, J. Yuan, S. A. Tasfarisa, and A. K. Katsaggelos. Anomalous
video event detection using spatiotemporal context. *Computer Vision
Image Understanding*, 115(3):323–333, 2011.

[9] J. Kim and K. Grauman. Observe locally, infer globally: a space-time
MRF for detecting abnormal activities with incremental updates. In
*CVPR*, pages 2921–2928, 2009.

[10] L. Kratz and K. Nishino. Anomaly detection in extremely crowded
scenes using spatio-temporal motion pattern models. In *CVPR*, pages
1446–1453, 2009.

[11] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and
localization in crowded scenes. *IEEE Trans. Pattern Analysis Machine
Intelligence*, 36(1):18–32, 2014.

[12] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in
MATLAB. In *ICCV*, pages 2720–2727, 2013.

[13] V. Mahadevan, W. Li, V. Bhodalia, and N. Vasconcelos. Anomaly
detection in crowded scenes. In *CVPR*, pages 1975–1981, 2010.

[14] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior
detection using social force model. In *CVPR*, pages 935–942, 2009.

[15] M. J. Rosilhami and M. D. Levine. An on-line, real-time learning
method for detecting anomalies in videos using spatio-temporal com-
positions. *Computer Vision Image Understanding*, 117(10):1436–1452,
2013.

[16] V. Saligrama and Z. Chen. Video anomaly detection based on local
statistical aggregates. In *CVPR*, 2012.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting
and composing robust features with denoising autoencoders. In *Int.
ACM Conf. Machine Learning*, pages 1096–1103, 2008.

[18] X. Wang, X. Ma, and E. Grimson. Unsupervised activity perception
by hierarchical bayesian models. In *CVPR*, pages 1–8, 2007.

---

**Fig. 12.** Comparison ROC curve *Left:* Frame-level evaluation *Right:* pixel-level evaluation

**Fig. 13.** Comparison between dual pixel localization with β equal to 0 (pixel-level measure), 5%, 10%, and frame-level (This figure is borrowed from [23] where the patches are considered without overlapping).
Fig. 14. Examples of normal and abnormal crowded activities in scenes of the UMN dataset. **Top:** Normal. **Bottom:** Abnormal

[19] S. Wu, B. Moore, and M. Shah. Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. In CVPR, 2010.

[20] D. Xu, R. Song, X. Wu, N. Li, W. Feng, and H. Qian. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. Neurocomputing, 143:144–152, 2014.

[21] Y. Yang, G. Shu, and M. Shah. Semi-supervised learning of feature hierarchies for object detection in a video. In CVPR, pages 1650–1657, 2013.

[22] D. Zhang, D. Gatica-Perez, S. Bengio and I. McCowan. Semi-supervised adapted HMMS for unusual event detection. In CVPR.

[23] Sabokrou, Mohammad and Fathy, Mahmood and Hoseini, Mojtaba and Klette, Reinhard. Real-Time Anomaly Detection and Localization in Crowded Scenes. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2015

[24] Zhu, Yingying and Nayak, Nandita M and Roy-Chowdhury, Amit K. Context-aware modeling and recognition of activities in video. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2491–2498, 2013.

[25] Piciarelli, Claudio and Micheloni, Christian and Foresti, Gian Luca. Trajectory-based anomalous event detection. In Circuits and Systems for Video Technology, IEEE Transactions on, 18(11): 1544–1554, 2006.

[26] Antonakaki, Panagiota and Kosmopoulos, Dimitrios and Perantonis, Stavros J. Detecting abnormal human behaviour using multiple cameras. In Signal Processing 89(9), 1723–1738, 2009.

[27] Calderara, Simone and Heinemann, Uri and Prati, Andrea and Cucchiara, Rita and Tishby, Naftali. Detecting anomalies in peoples trajectories using spectral graph analysis. In Computer Vision and Image Understanding, 115(8): 1099–1111, 2011.

[28] Morris, Brendan Tran and Trivedi, Mohan M. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. In Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(11): 2287–2301, 2011.

[29] A system for learning statistical motion patterns. Hu, Weiming and Xiao, Xuejue and Pu, Zhouyu and Xie, Dan and Tan, Tieniu and Maybank, Steve. In Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(9): 1450–1464, 2006.

[30] Tung, Frederick and Zelek, John S and Clausi, David A. Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. In Image and Vision Computing, 29(4), 230–140, 2011.

[31] Piciarelli, Claudio and Foresti, Gian Luca. On-line trajectory clustering for anomalous events detection. In Pattern Recognition Letters, 27(15): 1835–1842, 2006.

[32] Online dominant and anomalous behavior detection in videos. Rosshkari, Mehrsan Javan and Levine, Martin D. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, 2611–2618, 2013.

[33] Boiman, Oren and Irani, Michal. Detecting irregularities in images and in video. In International Journal of Computer Vision, 74(1), 17–31, 2007.

[34] Zaharescu, Andrei and Wildes, Richard. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In Computer Vision–ECCV 2010, 563–576, 2010.

[35] Online Anomaly Detection in Crowd Scenes via Structure Analysis. Yuan, Yuan and Fang, Jianwu and Wang, Qi. In Cybernetics, IEEE Transactions on, 45(3), 562–575, 2015.

[36] Image quality assessment: from error visibility to structural similarity. Wang, Zhou and Bovik, Alan Conrad and Sheikh, Hamid Rahim and Simoncelli, Eero P. In Image Processing, IEEE Transactions on, 13(4), 600–612, 2004.

[37] Reddy, Vikas and Sanderson, Conrad and Lovell, Brian C. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on, 55–61, 2011.

[38] Mousavi, Hossein and Nabi, Moin and Galogah, Hamed Kiani and Perina, Alessandro and Murino, Vittorio. Abnormality detection with improved histogram of oriented tracklets. In International Conference on Image Analysis and Processing (ICIAP) 2015.

[39] Xiao, Tan and Zhang, Chao and Zha, Hongbin. Learning to Detect Anomalies in Surveillance Video. In Signal Processing Letters, IEEE, 22(9): 1477–1481, 2015. 2005.