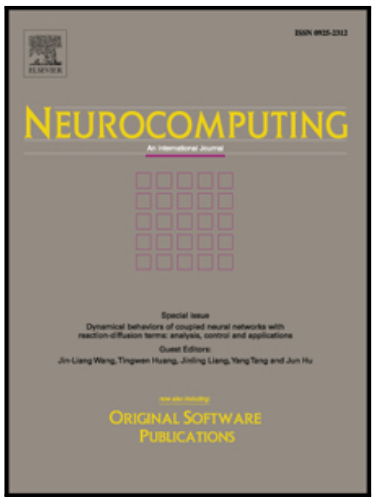


Video Anomaly Detection and Localization by Local Motion based Joint Video Representation and OCELM

Siqi Wang, En Zhu, Jianping Yin, Fatih Porikli

PII: S0925-2312(17)31411-X
DOI: 10.1016/j.neucom.2016.08.156
Reference: NEUCOM 18801

To appear in: *Neurocomputing*

Received date: 10 May 2016
Revised date: 27 June 2016
Accepted date: 13 August 2016

Please cite this article as: Siqi Wang, En Zhu, Jianping Yin, Fatih Porikli, Video Anomaly Detection and Localization by Local Motion based Joint Video Representation and OCELM, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2016.08.156

Video Anomaly Detection and Localization by Local Motion based Joint Video Representation and OCELM

Siqi Wang[a], En Zhu[a], Jianping Yin[a], Fatih Porikli[b]

[a]*College of Computer, National University of Defense Technology, Changsha 410073, China*
[b]*College of Engineering and Computer Science, Australian National University, Canberra ACT2601, Australia*

Abstract

Nowadays, human-based video analysis becomes increasingly exhausting due to the ubiquitous use of surveillance cameras and explosive growth of video data. This paper proposes a novel approach to detect and localize video anomalies automatically. For video feature extraction, video volumes are jointly represented by two novel local motion based video descriptors, SL-HOF and ULGP-OF. SL-HOF descriptor captures the spatial distribution information of 3D local regions' motion in the spatio-temporal cuboid extracted from video, which can implicitly reflect the structural information of foreground and depict foreground motion more precisely than the normal HOF descriptor. To locate the video foreground more accurately, we propose a new Robust PCA based foreground localization scheme. ULGP-OF descriptor, which seamlessly combines the classic 2D texture descriptor LGP and optical flow, is proposed to describe the motion statistics of local region texture in the areas located by the foreground localization scheme. Both SL-HOF and ULGP-OF are shown to be more discriminative than existing video descriptors in anomaly detection. To model features of normal video events, we introduce the newly-emergent one-class Extreme Learning Machine (OCELM) as the data description algorithm. With a tremendous reduction in training time, OCELM can yield comparable or better performance than existing algorithms like the classic OCSVM, which makes our approach easier for model updating and more applicable to fast learning from the rapidly generated surveillance data. The proposed approach is tested on UCSD ped1, ped2 and UMN datasets, and experimental results show that our approach can achieve state-of-the-art results in both video anomaly detection and localization task.

Keywords: video anomaly detection and localization, local motion based descriptors, extreme learning machine

1. Introduction

Surveillance cameras are gradually penetrating almost every corner of contemporary society. They play a center role in numerous realms such as municipal administration, traffic management and public security. The surging number of surveillance cameras naturally gives rise to a huge amount of surveillance video data, which are extremely tedious and time-consuming for manual analysis. Consequently, automatic video anomaly detection and localization are gaining increasing interest from both academia and industry.

Unlike classic object detection tasks like face detection or pedestrian detection, "anomaly" is a more abstract concept and its definition is not straightforward. Early research tends to concentrate on specific tasks in video anomaly detection. For example, Chung *et al.* [1] propose a behavior understanding system to detect abnormal behaviors of patients in a nursing center, while Foroughi *et al.* [2] adopt Support Vector Machine (SVM) for human fall detection. However, methods designed specifically for a certain task will obviously meet problems when dealing with unknown anomalies. Therefore, recent works in anomaly detection tend to consider video anomaly detection as an "outlier detection" problem [3], namely, only normal video events are modeled in the training phase and those events that divert significantly from normal events are viewed as anomalies. "Modeling normalcy" lays the foundation of most works in recent video anomaly detection research, including this paper. In addition to the anomaly definition, another key factor that has a significant impact on anomaly detection performance is whether the video scene is crowded. In uncrowded scenes where classic object tracking can be well performed, it is easy to extract high-level features with rich semantics, like object trajectory, for anomaly detection. A number of works like [4, 5, 6, 7] addressed such scenes soundly by object tracking and trajectory analysis. However, such methods perform poorly in crowded scenes with severe occlusion (See Fig. 1). Thus, robust low-level feature based approaches are proposed to address video anomaly detection in crowded scenes, which will be reviewed in Section 2.

In this paper, we aim to address anomaly detection and localization in videos with crowded or uncrowded scenes. Our approach can be roughly divided into two phases: Joint video representation and normalcy modeling. A flow chart of the proposed approach is shown in Fig. 2: First of all, training video volumes are jointly represented by two novel low-level descriptors, Spatially Localized Histogram of Optical Flow (SL-HOF) and Uniform Local Gradient Pattern based

Figure 1: Examples of crowded scenes from UCSD datasets.

Optical Flow (ULGP-OF), which are both based on local motion description in videos. To be more specific, after partitioning each spatio-temporal cuboid from videos spatially into numerous 3D local regions, SL-HOF is used to describe the motion of those local 3D regions and summarize their motion's spatial distribution. With the proposed Robust PCA based foreground localization scheme, ULGP-OF, which is a combination of the classic 2D texture descriptor Local Gradient Pattern (LGP) [8] and optical flow, is used to describe the motion of local region texture in video foreground. By virtue of SL-HOF and ULGP-OF, both motion statistics of local spatial region and local foreground texture are embodied by the proposed joint video representation. Subsequently, SL-HOF and ULGP-OF features are modeled respectively by OCELMs, which is an emerging data description algorithm that requires minimal training time to achieve a comparable or better data description performance. Finally, outlying video cuboids or patches are detected by the obtained OCELMs as video anomalies. Our contributions are three-fold:

- We propose a new SL-HOF descriptor to capture motion information of 3D local regions in the spatio-temporal video cuboid. Unlike HOF and MHOF descriptor in literature that describe the spatio-temporal cuboid as a whole, SL-HOF partitions the cuboids spatially into numerous 3D local regions and captures the spatial distribution information of those local regions' motion in a straightforward way, which can implicitly embed the structural information of foreground into the extracted features and characterize the motion of different foreground objects more precisely.

- We propose a novel ULGP-OF descriptor to describe the motion of local region texture in video foreground. In contrast to existing video descriptors that merely describe either motion or appearance of video, ULGP-OF not

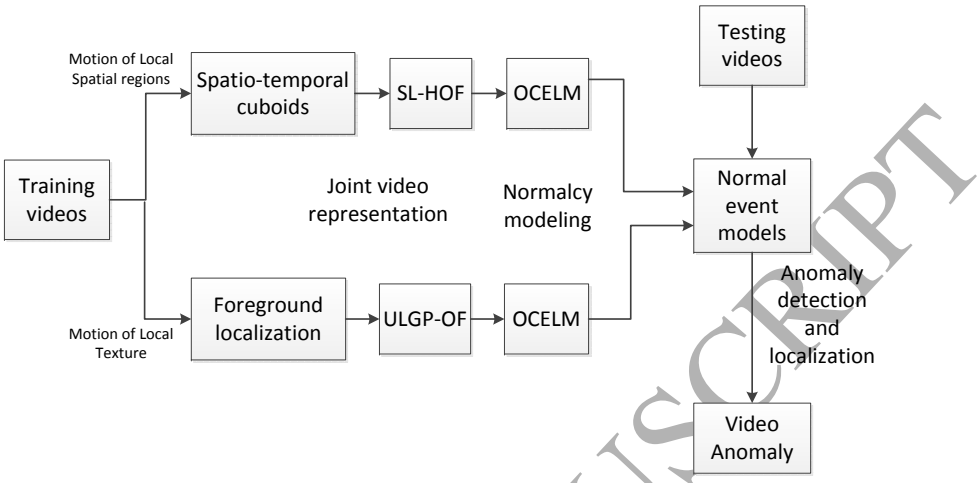


Figure 2: Flow chart of our approach.

only incorporates the local texture information but also the motion characteristics of local texture into the video representation, which enables ULGP-OF features to encode the interaction between texture and motion in video events. Therefore, ULGP-OF seamlessly combines texture and motion into the video representation. Meanwhile, a new foreground localization scheme is proposed to facilitate a more accurate localization of the video foreground texture for subsequent ULGP-OF feature extraction.

- We are the first to introduce the emerging OCELM into video anomaly detection as the data description algorithm for normalcy modeling. With a significant leap in learning speed, OCELM can achieve comparable or better performance than existing data description algorithms like OCSVM in literature. Furthermore, OCELM enables us to update model more easily and learn more rapidly from fast-growing surveillance data without loss of performance, which can be a promising solution to future video analysis.

The rest of papers are organized as follows: Section 2 reviews existing approaches of anomaly detection in crowded scenes, as well as other works related to the proposed approach. Section 3 presents the proposed video representation and analyzes the underlying reasons why SL-HOF and ULGP-OF can obtain a favorable video representation. Section 4 introduces the adopted OCELM for normalcy modeling. Section 5 reports the experimental results on commonly-used benchmark datasets of video anomaly detection, including comparing the

proposed video descriptors, data description algorithm as well as the detection and localization performance with those of the literature. Section 6 concludes this paper.

2. Related Work

In this section, we first introduce existing approaches in video anomaly detection and localization in terms of two aspects: Video representation and normalcy modeling. Then, relevant works on the low-level LGP descriptor and the emerging ELM are reviewed.

As to video representation, numerous low-level video descriptors are proposed for robust video anomaly detection in crowded scenes. One of the pioneering work of using low-level descriptors is Mahadevan *et al.* [9], who represent the appearance and dynamics of video frame patches by Mixture of Dynamic Texture (MDT). Mahadevan *et al.* also establish the most widely used video anomaly detection and localization datasets of crowded scenes, UCSD ped1 and ped2. Kratz *et al.* [10] apply spatio-temporal gradient (3D gradient) to characterizing video events in extremely crowded scenes, and Lu *et al.* [11] adopt this descriptor as well. Zhang *et al.* [12] use spatio-temporal gradients as the appearance cue of video event. In [13], Roshtkhari *et al.* represent the densely sampled spatio-temporal cuboids from videos by Histogram of Gradient (HOG). Similarly, Zhao *et al.* [14] combine HOG and Histogram of Optical Flow (HOF) to provide information of action and appearance in videos. Cong *et al.* [15] propose a Multi-scale HOF (MHOF) descriptor with different feature bases to preserve spatio-temporal contextual information. Cheng *et al.* [16] apply 3D HOG, HOF, and 3D SIFT to represent spatio-temporal interest points (STIPs) in video volumes.

When it comes to normalcy modeling, a variety of methods are proposed in the literature. one category of prevailing normal data description approaches is sparse coding. For example, Cong *et al.* [15] propose to select a limited number of normal training features to form a dictionary by solving a $l_{2,1}$-norm optimization problem, so that each normal feature can be reconstructed linearly by the dictionary with low reconstruction error. Zhu *et al.* [17] adopt a similar sparse coding based approach except that the Euclidean distance in optimization objective is replaced by Earth Mover's Distance (EMD). Zhao *et al.* [14] utilize the Laplacian Sparse Representation (LSR) to encode spatio-temporal feature vectors. To overcome the high computational cost in testing process, Lu *et al.* [11] propose to learn a series of smaller "sparse combinations" rather than a dictionary in [15], which enables a high testing speed by avoiding solving l_1-norm optimization problem.

Despite that sparse coding can yield relatively good performance, inducing sparsity by l_1 or $l_{2,1}$-norm optimization is usually time-consuming in both training and testing phase, and it often involves tuning parameters like Lipschitz Constant and reconstruction error bound, which are not straightforward to tune. In addition to sparse coding, other works also rely on OCSVM [18, 19, 20] or Support Vector Data Description (SVDD) [12] to model normal video features. OCSVM and SVDD can be implemented at a satisfactory speed during testing since they do not involve solving any difficult optimization problems. However, they still suffer from slow learning speed, especially when dealing with a large amount of video data. Other representative approaches include: Adam *et al.* [21] place several monitor points uniformly on video frames and represent normal event by optical flow histogram statistics. Mehran *et al.* [22] propose a social force (SF) model for anomaly detection. Li *et al.* [23] propose a joint detector to produce spatial and temporal saliency score based on hierarchical MDT (H-MDT), and Conditional Random Field (CRF) is used to guarantee consistency of anomaly judgement. Chen *et al.* [16] detect video anomaly by hierarchical feature representation and Gaussian Process Regression (GPR).

In this paper, we propose ULGP-OF descriptor to characterize the motion of local foreground texture, which is motivated by the low level descriptor LGP [8]. LGP is an improved version of the classic 2D texture descriptor Local Binary Pattern (LBP) [24]. LBP and LGP are popular in 2D static image texture description due to their sound properties such as invariance to monotonic gray-level change and robustness to local deformation. Another key component of our approach is OCELM. Proposed by Huang *et al.* [25], ELM has become a hot research topic due to its ultra-fast learning speed and favorable generalization performance when compared with classic learning algorithms like SVM. The efficacy of ELM has been demonstrated by a variety of machine learning tasks, including regression and multi-class classification [26], semi-supervised learning and clustering [27], online sequential learning [28]. Recently, deep architecture of ELM is also introduced into ELM for deep feature learning [29, 30]. For instance, Yang *et al.* [31] study the general architecture of multilayer ELM (ML-ELM) with subnetwork nodes for efficient feature dimension reduction. Cao *et al.* [32] combine ELM and sparse representation classifier (SRC) to enable a fast and accurate landmark recognition. ELM has also been applied successfully to various practical applications, e.g., object detection [33], image quality assessment [34], face recognition [35], 3D graphics shape learning [36], etc. OCELM is proposed by Leng *et al.* [37], however, only small UCI datasets and synthetic datasets are tested. We are the first to introduce OCELM into computer vision, and we also show OCELM

is able to yield state-of-the-art performance in video anomaly detection and localization tasks on benchmark datasets when combined with the proposed video representation.

3. Video Representation

In this section we show how to represent spatio-temporal video cuboids by SL-HOF and represent frame patches obtained by the foreground localization scheme by ULGP-OF, to obtain a joint video representation. Reasons of the proposed descriptors' effectiveness are also discussed.

3.1. Optical Flow and HOF

Before we present SL-HOF and ULGP-OF descriptor, we would briefly review the concept of optical flow and the classic HOF descriptor. As a powerful tool to describe motion in videos, optical flow [38] [39] estimates the magnitude and direction of each individual pixel in video frames by two neighboring frames. Based on optical flow, HOF descriptor is proposed. To be more specific, the calculation procedure of HOF is shown in Fig. 3: Suppose an optical flow vector $\mathbf{v}_i$ is calculated for each pixel i in a video unit (a frame patch or a spatio-temporal cuboid). The optical flow magnitude $|\mathbf{v}_i|$ is voted into D directions by the direction of optical flow to obtain a D-bin histogram as a HOF feature. HOF is one of the most widely used video descriptor that can robustly summarize the statistics of motion magnitude and direction in videos. In video anomaly detection, spatio-temporal cuboids from video are usually described by HOF to produce a video representation. However, the main drawback of such representation is that the spatial location information of each pixel's optical flow is entirely erased by calculating a histogram.

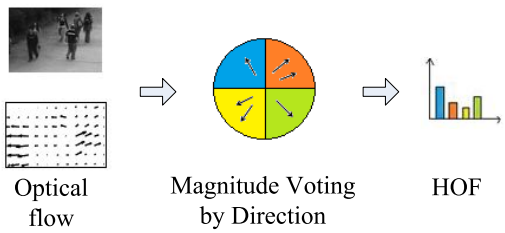


Figure 3: HOF feature extraction ($D = 4$).

3.2. SL-HOF Descriptor

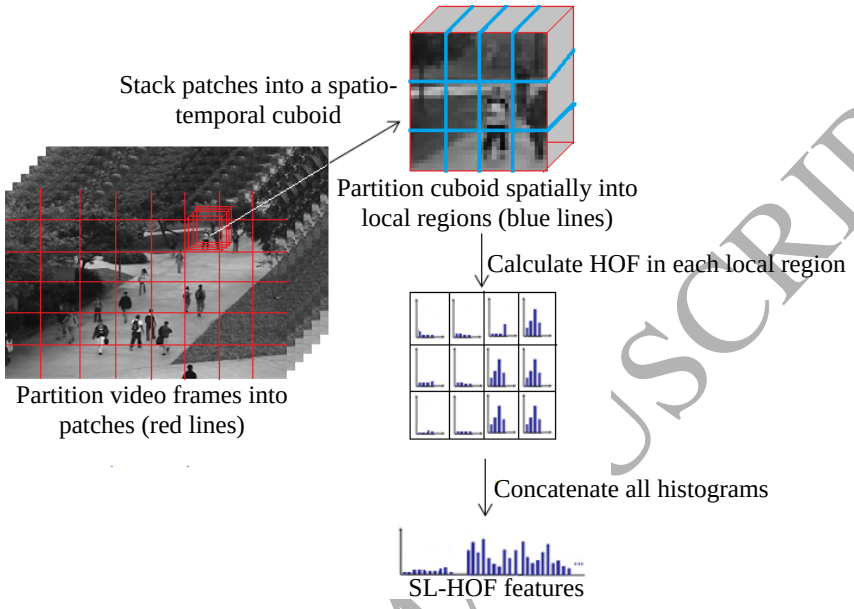
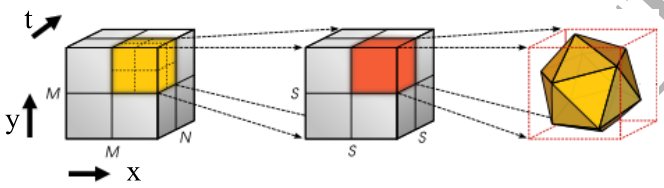


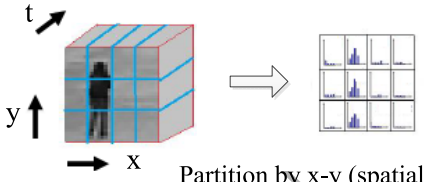
Figure 4: SL-HOF feature extraction ($m = 3$, $n = 4$).

We substitute the widely-used HOF descriptor by the proposed SL-HOF descriptor to enable the descriptor to not only depict the motion magnitude and direction, but also capture the spatial distribution of optical flow in spatio-temporal cuboids. We obtain a SL-HOF based video representation by the following steps (See Fig. 4): Firstly, video frames from normal training volumes are split into $M \times N$ non-overlapping patches with equal size, and d temporally consecutive patches at the same spatial location of video frames are stacked into a spatio-temporal cuboid. In our approach, we adopt a short temporal window (a small d like $d = 5$) to avoid the foreground changes drastically within the cuboid. A cuboid can be viewed as a local "video event" that contains one or several foreground objects. Secondly, the spatio-temporal cuboid is further partitioned spatially into $m \times n$ non-overlapping 3D local regions by different spatial $(x - y)$ locations (Uniform partition is adopted in our approach, but other partitions are also feasible). By the term "spatially localized", we mean that the cuboid is not further partitioned into smaller regions by temporal (t) location because it dampens the motion statistics of local regions (please note "spatially localized" does not mean that SL-HOF only extracts feature from spatial space like 2D texture

descriptor do). This differentiates SL-HOF from cell-based descriptors like 3D HOG [40], which partitions the cuboid not only by spatial location but also by temporal location (See Fig. 5). Next, a sub-histogram of optical flow $\mathbf{h}_i$ is extracted from 3D local region i, which differentiates SL-HOF from classic HOF that extracts histogram from the entire spatio-temporal cuboid. Finally, all histograms $\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{m\times n}$ are concatenated to obtain a SL-HOF feature $\mathbf{f}_{SLHOF} = [\mathbf{h}_1; \mathbf{h}_2; ...; \mathbf{h}_{m\times n}]$. m and n can be adjusted according to different foreground object scales.



Partition by x-y (spatial) location and t (temporal) location



Partition by x-y (spatial) location

Figure 5: Different partition ways of spatio-temporal cuboids by 3D HOG (top) and SL-HOF (bottom).

The operations of SL-HOF descriptor are pretty straightforward and understandable. However, our experiment in Section 5 will show SL-HOF can work significantly better than other frequently-used video descriptors. The main advantage of SL-HOF over HOF is that SL-HOF aims to characterize the motion of 3D local regions rather than the entire spatio-temporal cuboid. This scheme can benefit video representation in two aspects:

First of all, it preserves the spatial distribution information of optical flows in the spatio-temporal cuboid by calculating a sub-histogram for each spatial location, which actually preserves pixels' spatial location information erased by histogramization to a certain extent, and such information is able to reflect the structural information of foreground objects. We illustrate this point by a example of a walking pedestrian (normal event) and a man in the wheelchair (abnormal event) moving at a close speed to the same direction. As can be seen in Fig. 6, HOF descriptor can be easily fooled since two objects share close speed and mov-

ing direction. By contrast, SL-HOF is able to get very different representation for two objects due to their different structures: Strong optical flow histograms ("strong" means bins of the histogram has large vote values) can be observed in region 2, 6 and 10 for the walking man, while region 3, 6, 7, 8, 10, 11 and 12 obtain strong histograms for man in the wheelchair, thus leading to quite different SL-HOF features. Such information clearly contributes to discriminating abnormal foreground objects with abnormal structures, but no existing work explores this to our knowledge.

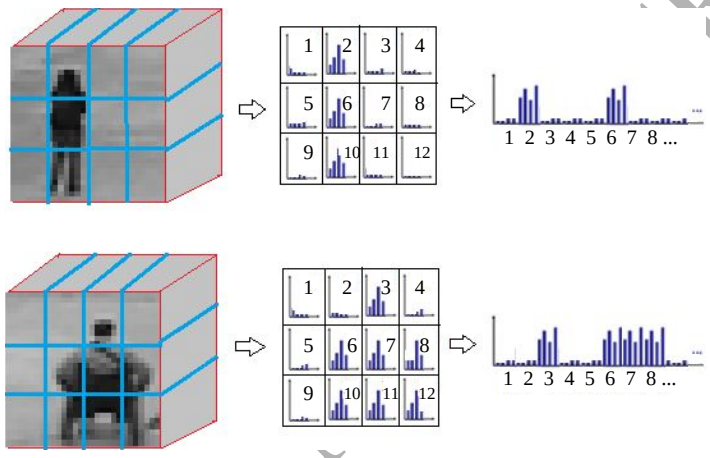


Figure 6: Foreground structural information embedded in SL-HOF features.

Secondly, SL-HOF can characterize the motion of foreground object more accurately, which is explained by Fig. 7: A walking man (normal event) and a skater (abnormal event) both moving towards right. Like the example above, their difference of HOF representation is minor when their speed is close. However, we can discover their difference easily by SL-HOF representation: The skater on the skateboard are moving as a whole. Therefore, region 2, 3, 6, 7, 10 and 11 can all observe a strong optical flow histogram. However, the local motion of a walking man's body parts is not consistent like a skater since human's two legs advance alternatively while walking. In the example of Fig. 7, the man's supporting leg in region 7 and 11 remain static while the other leg in region 6 and 10 steps forward rapidly, which leads to weaker histograms in region 6 and 10. Thus, the yielded SL-HOF features of walking man and skater evidently differ from each other. Consequently, SL-HOF can yield more discriminative video representation than HOF, and a quantitive comparison between SL-HOF and other descriptors will be given in Section 5.

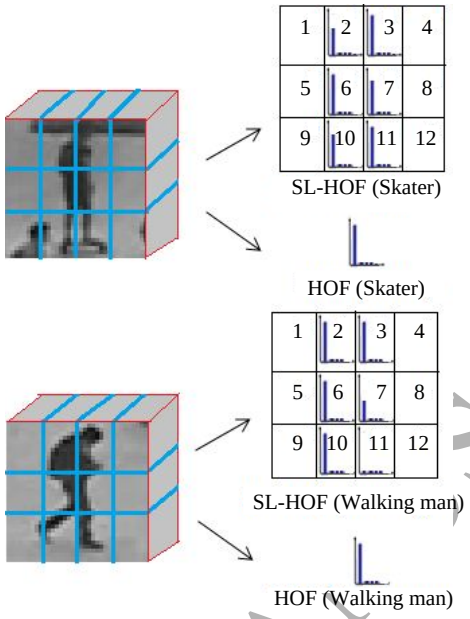


Figure 7: Foreground motion description using SL-HOF features.

3.3. Foreground Localization

By representing spatio-temporal video cuboids with SL-HOF descriptor, we can efficiently capture the local motion statistics in 3D local regions, including motion magnitude, direction and spatial distribution. In addition to motion statistics of local regions, we also attempt to incorporate texture (appearance) information of video foreground into our video representation. Instead of describing texture of video frames directly as [9] and [12] do, we characterize the motion of texture in video foreground, because only texture of active foreground contributes to anomaly detection. However, one problem is that partitioning videos into spatio-temporal cuboids is unable to locate foreground objects accurately with many foreground objects being "torn apart" by the partition. To alleviate this problem for ULGP-OF based texture motion description, we therefore propose a new foreground localization scheme to extract patches of foreground.

The procedure of the proposed foreground localization scheme is shown in Fig. 8: Firstly, since the surveillance videos are usually shot by static cameras, Robust Principle Component Analysis (RPCA) [41] can model the scene background $\mathbf{B}$ by considering it as a low-rank matrix recovery problem in Eq. 1:

$$\min_{\mathbf{A},\mathbf{E}} = ||\mathbf{A}||_* + \lambda||\mathbf{E}||_1 \quad s.t. \quad \mathbf{D} = \mathbf{A} + \mathbf{E} \tag{1}$$

Where $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N]$, and $\mathbf{d}_i$ is obtained by squeezing the i_{th} video frame into a column vector. $|\cdot|_*$ is the nucleus norm. N denotes the number of training video frames. $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N]$ is supposed to be a low-rank matrix formed by $\mathbf{a}_i$, which is the column vector squeezed from the background of i_{th} training video frame. $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N]$ is the sparse matrix formed by the foreground of each video frame. Having obtained $\mathbf{A}$ from Eq. 1, the procedure of foreground localization is shown in Fig. 8: First, we obtain the background image by averaging the background of each frame: $\mathbf{B} = reshape(\frac{1}{N}\sum_{i=1}^{N}\mathbf{a}_i)$, where the function $reshape(\cdot)$ reshapes the vector into a matrix with the same size as the original video frame. Then we subtract $\mathbf{B}$ from an input video frame $\mathbf{I}$ to extract the foreground $\mathbf{F} = |\mathbf{I} - \mathbf{B}|$, where $|\cdot|$ denotes getting element-wise absolute value of the matrix. With foreground $\mathbf{F}$, each pixel's probability of being foreground is estimated by a sigmoid transformation $p_{i,j} = 2/exp(-\lambda \cdot \mathbf{F}_{i,j}^2) - 1$. The sigmoid transformation can map $\mathbf{F}_{i,j}$, the absolute value of difference between background $\mathbf{B}$ and a frame $\mathbf{I}$ at pixel (i, j), into [0,1] to facilitate the later binarization. In our experiments, we simply estimate $\lambda = 1/\sqrt{N_{frame}}$, where N_{frame} is the number of pixels on a video frame. The third step is to obtain a binarization map $\mathbf{F}'$ of foreground using the estimated probability $p_{i,j}$ by binarizing pixels with $p_{i,j} > 0.5$ into 1, while others into 0. Finally, Algorithm 1 is used to generate a series of bounding boxes with equal size to locate the foreground objects and cover the majority of foreground pixels. Four input parameters are needed for Algorithm 1: Binarization map $\mathbf{F}'$, bounding box height H and width W, minimum number of foreground pixels T_{fore} that should be covered by bounding boxes and minimum number of remaining foreground pixels that should be covered by a new bounding box T_{gain}, while the output is the centers of bounding boxes $\mathbf{C}_{box}$. T_{fore} is used to ensure that most foreground pixels are covered. T_{gain} is set to avoid generating redundant boxes that overlap too much with previous bounding boxes. If the number of remaining foreground pixels covered by a new bounding box is less than T_{gain}, this box will not be added to $\mathbf{C}_{box}$, because it does not bring enough "gain" to covering the foreground pixels and is considered redundant. Candidate centers are sorted by number of covered foreground pixels in descending order to encourage generating boxes at the approximated center of foreground objects. Suppose the total number of foreground pixels are N_{fore} and the total number of pixels in a bounding box is N_{box}, we simply use $T_{fore} = 0.975 \cdot N_{fore}$ and $T_{gain} = 0.05 \cdot N_{box}$. As shown in Fig. 8, the proposed scheme can locate video

foreground fast and accurately in a simple way, and each video patch inside a bounding box is then represented by ULGP-OF descriptor.

Algorithm 1 Foreground Localization.

Input:

$\mathbf{F}'$, H, W, T_{fore}, T_{gain}

Output:

$\mathbf{C}_{box}$

Initialize current number of covered foreground pixels $N_{cover} = 0$, bounding boxes center $\mathbf{C}_{box} = \varnothing$, candidate centers $\mathbf{C}_{candi}$ are initialized to be all foreground pixels.
Sort $\mathbf{C}_{candi}$ by the number of foreground pixels that can be covered by the bounding box centered at one candidate center in descending order.
while $N_{cover} < T_{fore}$ **do**
    Select the first center C_{cur} from $\mathbf{C}_{candi}$, calculate the number of remaining foreground pixels in $\mathbf{F}'$ covered by the bounding box at C_{cur}, N_{cur}.
    if $N_{cur} > T_{gain}$ **then do**
        $N_{cover} = N_{cover} + N_{cur}$,
        $\mathbf{C}_{box} = \mathbf{C}_{box} \bigcup C_{cur}$,
        For $\mathbf{F}'_{i,j}$ covered by current bounding box, set $\mathbf{F}'_{i,j} = 0$.
    end if
    Remove C_{cur} from $\mathbf{C}_{candi}$.
end while

3.4. ULGP-OF Descriptor

Having located the video foreground, we propose to represent the foreground texture (appearance) inside each bounding box by ULGP-OF descriptor. The proposed ULGP-OF is based on 2D texture descriptor Uniform LGP (ULGP), which will be reviewed in the first place. A LGP code can be calculated by the procedure shown in Fig. 9: For a 3×3 pixel local image area, the intensity of center pixel and its 8 surrounding sampling points are denoted by x_c and x_i, $i = 1, 2, ...8$, respectively. The local gradient at each sampling points x_i is approximated by $g_i = |x_i - x_c|$, and a binarization threshold T is calculated by averaging 8 local gradients: $T = \frac{1}{8}\sum_{i=1}^{8} g_i$, where g_i is the local gradients of the i_{th} neighbor point, $i = 1, 2, ...8$. Then g_i is binarized into 0 and 1 by T to obtain an 8-bit binary code ranging from 0 to 255, which encodes local texture pattern in this local image area. A LGP feature vector is obtained by calculating a 256-bin histogram of all

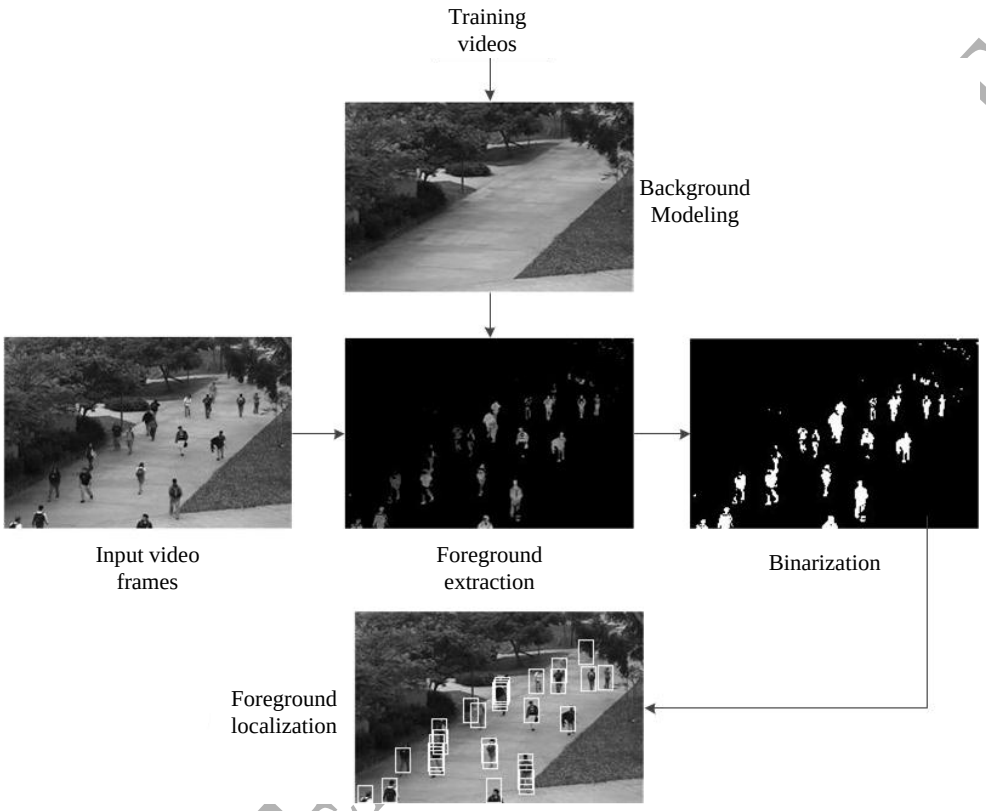


Figure 8: Foreground localization.

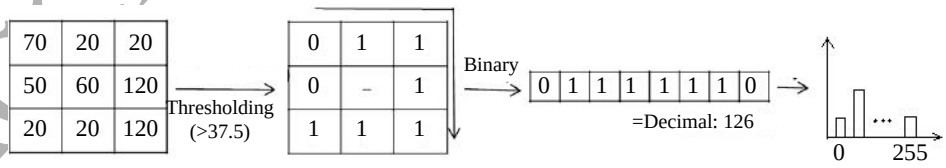
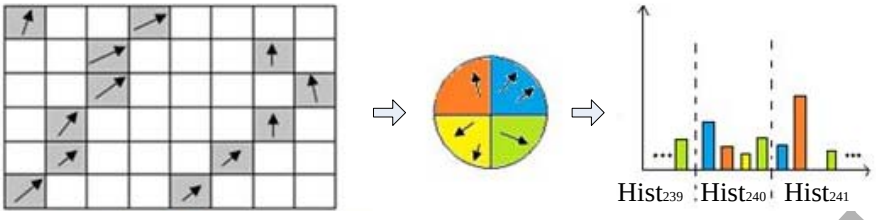


Figure 9: The encoding process of LGP.

LGP codes in an image. Each LGP code (or bin index) represents one type of local region texture. However, not each LGP pattern from 0 to 255 is informative for texture description, and different LGP codes do not emerge at a equal probability, thus often leading to a high-dimension sparse feature vector. Hence, Topi *et al.* [42] proposed the concept of "uniform pattern". "Uniform pattern" is based on the discovery that the majority of information is carried by edges in real-world images, and the LGP codes of object edges are highly likely to be "uniform", which means at most two 0-1 or 1-0 jumps should be observed in one circular LGP binary code. For instance, 00011100 and 00000000 are uniform codes while 01010011 is not. The constraint of uniformity excludes most uninformative codes, and it also reduces the dimension of a LGP feature vector from 256 to 58.

On the ground of ULGP, the core idea of ULGP-OF is to describe the motion of local texture, which is represented by different ULGP codes. We take ULGP code 11110000 (240) as an example to illustrate the calculation of ULGP-OF (See Fig. 10): For each pixel in a video frame, a LGP code is calculated to represent its local image texture. The optical flow at this pixel is considered as the motion of this local texture. Subsequently, if the LGP code is uniform, optical flows of all pixels with this LGP code are collected to calculate a D-bin optical flow sub-histogram in the same way of HOF (In Fig. 10, LGP=240, $D = 4$, the corresponding sub-histogram is $Hist_{240}$), which summarizes the motion statistics of the local texture represented by this LGP code. By concatenating sub-histograms of all ULGP codes, we obtain a $4 \times 58 = 232$-dimension ULGP-OF feature in Fig. 10. In essence, ULGP-OF replaces the original vote weight, the number of ULGP code, by optical flow magnitude. It is interesting to note that ULGP-OF can implicitly filter out the texture of background: since the video background is static, the optical flow magnitude of background pixel is 0 (or very close to 0), so the voting weight of the background pixel is 0. Adding a 0 weight will not change the calculated ULGP-OF feature. In other words, only foreground pixels with motion can have a significant influence on calculating the histogram of ULGP-OF, and what we care in the video is exactly the active foreground rather than the static background. Besides, ULGP-OF inherits the sound properties of LGP and HOF by seamlessly combining the two descriptors: For one thing, ULGP-OF can indicate the composition of video foreground texture like LGP, because the magnitude of each sub-histogram reflects the amount of each texture component in video foreground. For another, a ULGP-OF feature also contains motion and direction information like HOF. By ULGP-OF, we incorporate both local foreground texture and its motion into our video representation. A quantitive experiment is also given in Section 5 to demonstrate its efficacy.



Pixels with the same LGP code 240 and their optical flow

Voting

Calculate histogram $Hist_{240}$ and concatenation

Figure 10: The encoding process of ULGP-OF (LGP=240).

4. Normalcy Modeling

In this Section, we present the adopted one-class data description algorithm OCELM for normalcy modeling. We model the extracted SL-HOF and ULGP-OF features from normal video events by OCELM, which is a simple variant of the emerging ELM. The usage of ELM is motivated by the characteristics of video data. As a stream of 2D images, the data size of video can be far greater than 2D static images. For example, a five-minute 160×240 video can generate 7200 images and more than 10^6 patches or spatio-temporal cuboids for processing with a local patch size 10×10, which makes the training time required by traditional methods like sparse dictionary learning and OCSVM hardly bearable. Meanwhile, video data are generated rapidly at a real-time speed. Since all normal video events cannot be enumerated at one time and newly-incoming normal events are supposed to be included every now and then, the normal event models should be easy for re-training and updating, which can be pretty hard for OCSVM or sparse dictionary learning. As a result, ELM, which can achieve comparable or higher data description performance with much less training time required, becomes a promising solution to video anomaly detection. In addition, we would like to clarify why OCELM rather than the original basic ELM is adopted here: In video anomaly detection, usually there are only data of normal video events for training, because abnormal events are unpredictable and rarely seen when compared to normal events, which makes collecting training data of abnormal events particularly difficult. Besides, abnormal events are almost impossible to be completely enumerated for building a complete classification model. Therefore, due to the absence of training data of abnormal events, we formulate this problem as an one-class learning/outlier detection problem rather than a classic classification problem. We will review basic ELM first before we present OCELM.

Basic ELM is a three-layer feedforward neural network. The essence of ELM is to randomly generate the weights between the input layer and hidden layer, which are not tuned in subsequent training, and the weights between hidden layer and output layer are determined analytically by solving a least square optimization problem rather than classic error back-propagation. In other words, ELM's fast learning speed can be ascribed to not involving iterative weight tuning, and Huang *et al.* [25] prove the universal approximation capability of ELM. To be more specific, with the input training set $\mathbf{X}_{n\times d}$ and L hidden nodes (n and d are the number of training features and the number of feature dimension), the input features are randomly mapped to a new feature space as the output of hidden layer $\mathbf{H}_{n\times L}$. Then the output weights $\boldsymbol{\beta}$ between hidden layer and output layer are determined by Moore-Penrose persudo inverse:

$$\boldsymbol{\beta} = \mathbf{H}^T(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{2}$$

where C, $\mathbf{T}$ and $\mathbf{I}$ denote the regularization coefficient, target output and identity matrix, respectively. With the obtained neural network, the prediction of a new sample $\mathbf{x}$ is given by:

$$f(\mathbf{x}) = h(\mathbf{x})\boldsymbol{\beta} = h(\mathbf{x})\mathbf{H}^T(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{T} \tag{3}$$

where $h(\mathbf{x})$ is the random mapping of $\mathbf{x}$. If the random mapping is unknown, the prediction of $\mathbf{x}$ can be determined by using kernel tricks:

$$f(\mathbf{x}) = \mathbf{K}_{test}^T(\frac{\mathbf{I}}{C} + \mathbf{K}_{train}^T)^{-1}\mathbf{T} \tag{4}$$

where $\mathbf{K}_{train}$ and $\mathbf{K}_{test}$ are kernel matrices. Modifying an ELM into OCELM is straightforward: Since all training samples in one-class learning problems have the same label value y, the target output $\mathbf{T}$ is given by $\mathbf{T} = \mathbf{1}\cdot y$, which corresponds to a single output node (See Fig. 11). Assuming the actual outputs of OCELM for training samples are y_i, $i = 1, 2, ...n$, the mapping error of training sample x_i to the target value y is $d_i = |y_i - y|$. A threshold d_T is chosen to exclude a small fraction (p) of farthest training points ($d_i > d_T$), which can prevent the outliers in training set from degrading data description performance of OCELM. In practice, y is set to be 1 and p usually takes small value like 0.05 or 0.01. According to [37], we adopt Gaussian kernel based OCELM in our approach to obtain the best data description performance. As shown above, the training and testing of OCELM do not involve any iterative optimization procedure, resulting in a much faster learning speed.

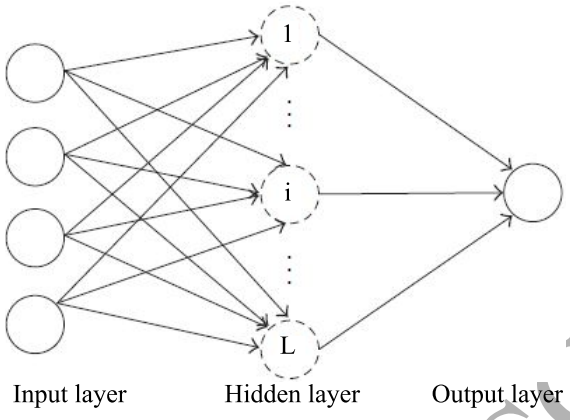


Figure 11: Network structure of OCELM.

5. Experiments

In this section, we report the experimental results regarding the proposed approach. First of all, we present the experimental setup and commonly-used evaluation criteria for video anomaly detection and localization in Section 5.1. Secondly, we demonstrate the effectiveness of the proposed descriptors, data description algorithm and joint video representation in Section 5.2, Section 5.3 and Section 5.4, respectively. In Section 5.5, the proposed approach is tested on three publicly available datasets and its performance is compared with other state-of-the-art approaches in literature. Finally, the computational efficiency of the proposed approach is discussed in Section 5.6.

5.1. Experimental Setup and Evaluation Criteria

The experiments are carried out on three datasets: UCSD ped1, UCSD ped2 and UMN datasets. UCSD ped1 and ped2 pedestrian datasets [9] are the most frequently-used datasets in video anomaly detection and localization. UCSD ped1 dataset contains 34 training video volumes and 36 testing video volumes, each volume consists of 200 frames with a resolution of 158×238. UCSD ped2 dataset contains 16 training volumes and 12 testing video volumes with 240×360 video frames, and the number of video frames in each volume varies from 120 to 180. Both UCSD ped1 and ped2 datasets contain challenging crowded and uncrowded scenes with different sorts of anomalies on campus pavement, such as skaters, bikers and vehicles. UMN dataset [22] is another widely-used dataset that contains normal crowd activities (walking) and abnormal crowd activities (escaping)

in different scenes, and it has 7740 240×320 frames in total (1450, 4415 and 2145 frames for scene 1-3 respectively). The first 400 frames in each scene are used for training and the rest are left for testing.

For detection and localization on UCSD ped1 and UCSD ped2 dataset, we partition the video volumes into $10 \times 10 \times 5$ spatio-temporal cuboids, and the cuboids with only minimal temporal gradient accumulation value are filtered out. The rest of cuboids are represented by SL-HOF with $m = 7$, $n = 8$ to yield the best performance, while we set the bins of optical flow direction $D = 4$. Spatio-temporal base in [15] is used to organize SL-HOF features and PCA is performed to reduce the SL-HOF feature dimension to 950. To detect anomalies with different sizes, SL-HOF features are extracted on multiple scales: 120×180, 100×150 and 80×100 for UCSD ped1 dataset and 180×270, 120×180 and 100×150 for UCSD ped2 dataset. Only SL-HOF features extracted from the same spatial location of video frames with the same scale are used to train and test. Foreground localization is performed on the original scale of video frames with 21×12 and 34×17 bounding box for UCSD ped1 and ped2 respectively, and the video frames are divided uniformly into 7×11 spatial regions. Likewise, only ULGP-OF features extracted from those bounding boxes, whose centers lies in the same local spatial region, are used to train and test. Anomalies detected by SL-HOF and ULGP-OF are combined as the final detection result. For UMN dataset, the detection is performed on one scale 80×100 since the foreground objects share a close size. The size of spatio-temporal cuboid is $20 \times 20 \times 5$ and $m = 2, n = 3$ for SL-HOF feature extraction, while spatial base [15] is adopted. The bounding box is set to be 40×20 for foreground localization on the original scale. To parameterize OCSVM and OCELM in the experiments, we select the regularization coefficient ν and C from $[2^{-20}, 2^{-11}, ..., 2^{0}]$ and $[2^{-10}, 2^{-11}, ...2^{11}, 2^{10}]$ respectively. Gaussian kernel width σ is selected from $[2^{-10}, 2^{-11}, ...2^{11}, 2^{10}]$. Parameters are determined by 10-fold cross-validation. The rejected ratio is set to be $p = 0.01$ for OCELM.

As for evaluation criteria, we adopt frame-level criteria for anomaly detection and pixel-level criteria for anomaly localization [23]: *Frame-level criteria.* A video frame that contains any detected abnormal pixel is considered as an abnormal frame. The abnormal frames detected by a method is compared with the ground truth frames on a per-frame basis. *Pixel-level criteria.* The pixel-level criteria are more precise and challenging than the frame-level one. Only when 40% pixels of the ground truth abnormal event are detected by the method, a frame can be viewed as a successfully detected abnormal frame. That is to say, the method is required to not only determine the frame index of abnormal events, but also localize the abnormal events roughly. In fact, anomaly localization can be viewed

Table 1: Comparison of video descriptors.

Descriptor	EER	AUC
MHOF	29%	76.45%
3D HOG	31%	73.98%
HOF	32%	75.56%
HOG+HOF	29%	76.28%
3D Gradient	33%	70.87%
ULGP-OF	23%	82.29%
SL-HOF	21%	85.73%

as a "refined" anomaly detection process. For frame-level and pixel-level evaluation, Equal Error Rate (EER), ROC Curves and Area Under the Curve (AUC) are calculated for a quantitive comparison. All experiments are carried out under MATLAB 2015b environment on a PC with 32 GB RAM and 3.90 Ghz Intel i7 4790 processor.

5.2. Descriptor Comparison

In this section, we design an experiment to compare the proposed SL-HOF and ULGP-OF descriptor with the following classic video descriptors: 3D Gradient [10], HOF, MHOF [15], 3D HOG [43], HOG+HOF [14]. Spatio-temporal cuboids are extracted from the training video volumes of UCSD ped1 dataset on a single scale (100×150) and represented by the above descriptors respectively. The extracted features are all modeled by the same Gaussian kernel based OCSVM for subsequent anomaly detection. ROC Curves, EERs and AUCs yielded by different descriptors under frame-level criteria clearly show the improvement of discriminative power by the proposed descriptors (See Fig. 12 and Table 1):

As can be seen from Fig. 12 and Table 1, both SL-HOF and ULGP-OF significantly outperform other descriptors in the experiment by a 6% to 12% EER improvement and a 6% to 15% AUC improvement. The results justify the proposed descriptors for video representation in video anomaly detection. Besides, it should also be noted that SL-HOF and ULGP-OF perform much better than two existing optical flow based descriptors, MHOF and HOF, which describe motion from the entire spatio-temporal cuboid rather than local motion. It verifies our claim that local motion based descriptors can lead to a more effective video

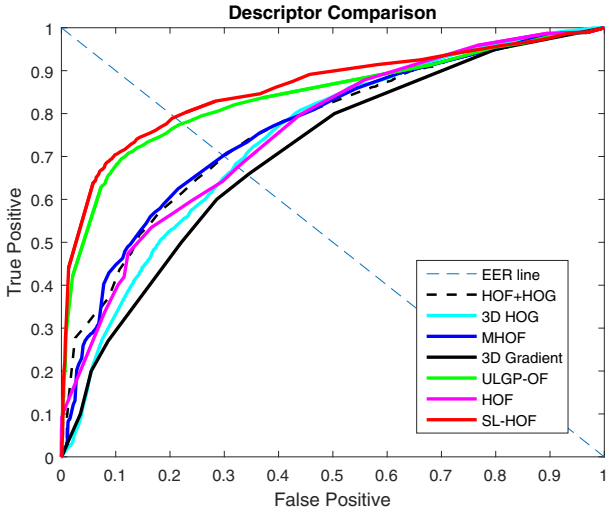


Figure 12: Descriptor comparison.

representation for anomaly detection.

5.3. Data Description Algorithm Comparison

In this section, we follow the experimental setup in Section 5.1 and use classic OCSVM [44] and the adopted OCELM as the data description algorithm respectively to compare their performance on UCSD ped1 and UCSD ped2 dataset under the more precise pixel-level evaluation criteria. The results are displayed in Fig. 13 and Table 2. To show the learning speed, the average training time required to train one OCELM or OCSVM with the SL-HOF features extracted from one spatial location of video frames is also listed in Table 2. We also show the learning time needed by Sparse Reconstruction Cost (SRC) [15], which is a representative sparse coding based method used in video anomaly detection, as a reference (The performance of SRC is omitted since the optimal parameterization of SRC cannot be determined by crossvalidation like OCELM and OCSVM, and we will report the performance of SRC in Section 5.5 directly from [15]).

As can be seen from Fig. 13 and Table 2, the adopted OCELM can achieve comparable or superior EERs and AUCs to the classic OCSVM with a 50 times faster learning speed. Actually, the advantage of OCELM should be even larger since OCSVM takes a faster C implementation while OCELM is implemented by Matlab. We also note that the learning speed of sparse coding based SRC is much slower than both OCSVM and OCELM, even though we only implement

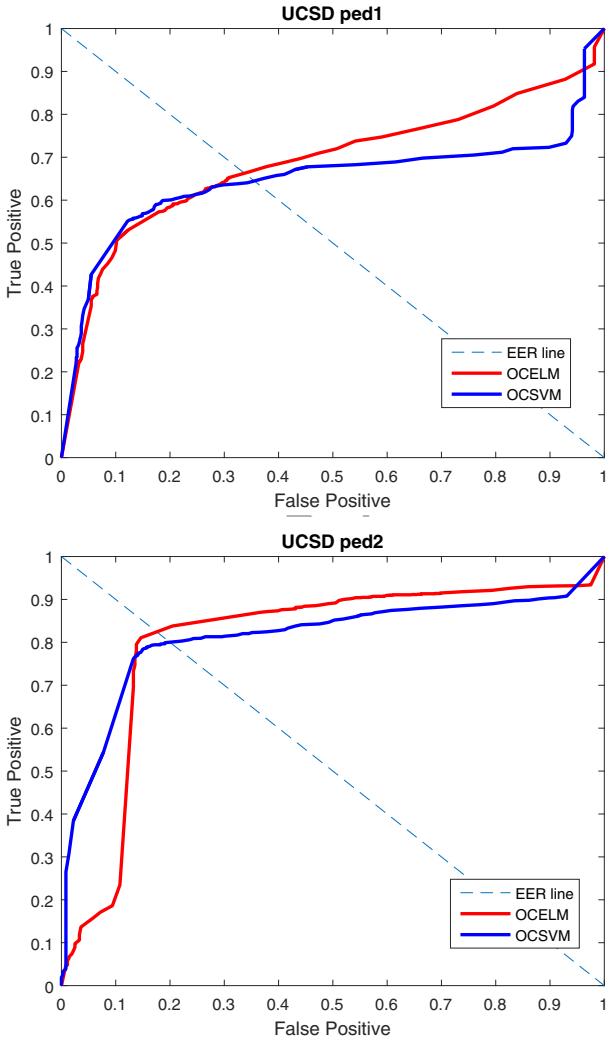


Figure 13: Comparsion of OCSVM and OCELM.

Table 2: Comparison of data description algorithms.

Algorithm	EER (ped1)	AUC	EER (ped2)	AUC	Training time
OCSVM	35%	64.79%	19%	81.27%	4.0008s
OCELM	33%	68.88%	17%	80.12%	0.7915s
SRC	-	-	-	-	25.7508s

Table 3: Anomaly localization performance by SL-HOF, ULGP-OF and joint video representation.

Video Descriptor	EER(ped1)	AUC	EER(ped2)	AUC
SL-HOF	36%	65.67%	19%	77.77%
ULGP-OF	43%	60.81%	26%	70.47%
Joint	33%	68.88%	17%	80.12%

the optimization by merely 50 iterations, which is not enough for the objective to converge in most cases. Besides, there are more parameters for SRC to tune (e.g., Lipschitz constant, reconstruction error bound). They are not straightforward to tune and cannot be determined conveniently by cross-validation like OCSVM and OCELM. Consequently, we adopt OCELM as our data description algorithm to model video normal events.

5.4. Joint Video Representation

In this section, we show that the proposed joint video representation can yield better localization performance than single descriptor based video representation. We represent the training video volumes by SL-HOF and ULGP-OF alone, and compare their anomaly localization performance with the proposed joint video representation under the pixel-level criteria. The comparison is made on UCSD ped1 and UCSD ped2 dataset. The results are shown in Fig. 14 and Table 3: Compared to single descriptor based video representation, the proposed joint video representation enhances the performance on both datasets. There are two reasons for the improvement: Firstly, the joint video representation combines the 3D local region motion information carried by SL-HOF and local texture motion information carried by ULGP-OF. Secondly, the proposed foreground localization scheme enables the proposed method to localize the abnormal video foreground more accurately than single spatio-temporal cuboid based video representation.

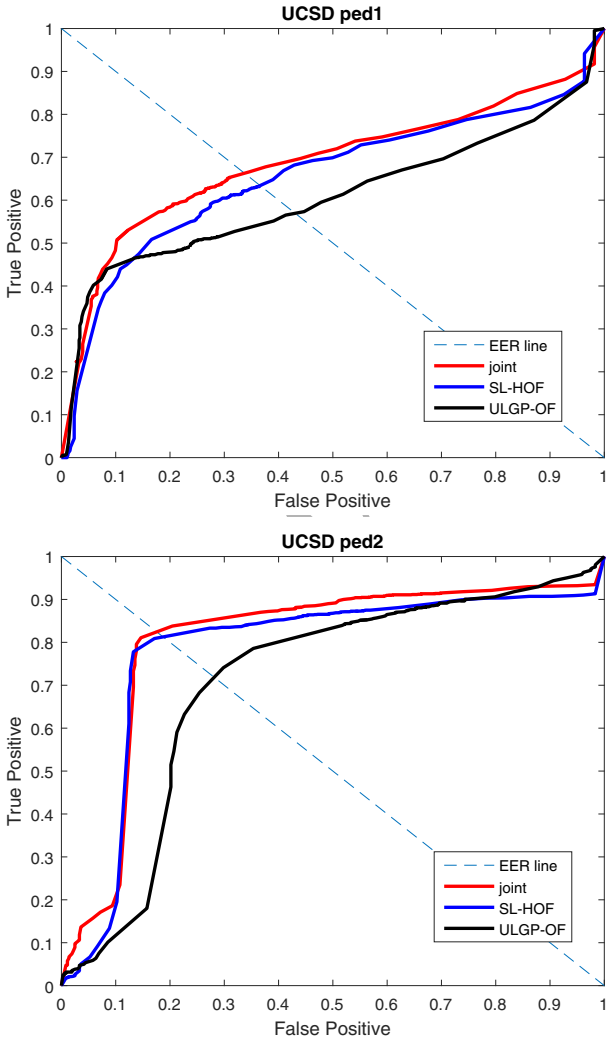


Figure 14: Joint video representation.

5.5. Method Comparison

5.5.1. UCSD ped1 and ped2 Datasets

In this section, we test the proposed anomaly detection and localization approach on UCSD ped1 and UCSD ped2 datasets. For UCSD ped1 dataset, the following state-of-the-art methods in literature are used for comparison: Sparse Reconstruction Cost (SRC) [15], Sparse Combination Learning (SCL) [11], Motion and appearance cues (MAC) [12], Gaussian Process Regression (GPR) [16], HMDT+CRF [23], Spatio-temporal Context (STC) [45], MDT [9], Social Force (SF) [22], Social Force+MPPCA (SF+MPPCA), Adam *et al.* [21], Dense STC [13]. The detection and localization ROC curves are plotted in Fig. 15 and Fig. 16, and the EERs and AUCs under the frame-level evaluation and pixel-level evaluation are listed in Table 5.5 ("-" means the result is not given in the literature). As it is seen from Table 5.5, the proposed approach achieves comparable anomaly detection performance to state-of-the-art results under frame-level evaluation criteria (EER 18% and AUC 88.5%), while it yields the best EER (33%) and AUC (68.9%) under the more precise pixel-level evaluation.

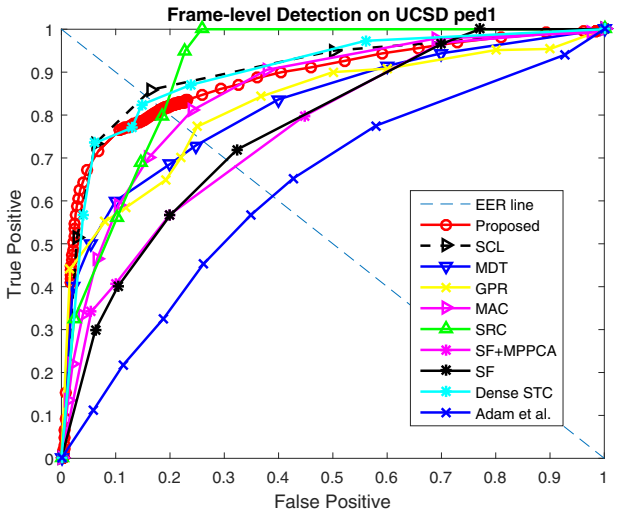


Figure 15: Frame-level anomaly detection on UCSD ped1 dataset.

For UCSD ped2 dataset, the following approaches are compared with our approach: Motion and appearance cues (MAC) [12], HMDT+CRF [23], Spatio-temporal Context (STC) [45], Spatio-temporal Composition (SC) [46], MDT [9], MPPCA [47], Social Force and MPPCA (SF+MPPCA), Bertini *et al.* [48] and

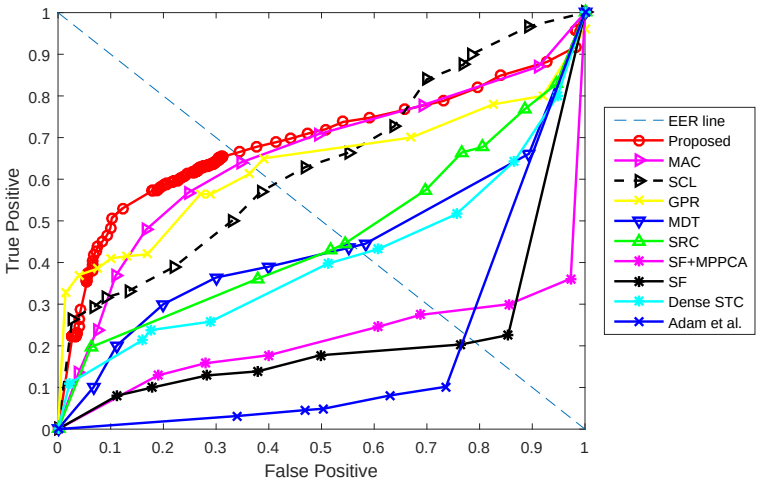


Figure 16: Pixel-level anomaly localization on UCSD ped1 dataset.

Table 4: Method comparison on UCSD ped1 dataset.

Method	EER (frame)	AUC	EER (pixel)	AUC
Proposed	18%	88.5%	**33**%	**68.9**%
GPR	24%	83.8%	37%	63.3%
MDT	25%	81.8%	56%	44.1%
HMDT+CRF	18%	-	35%	66.2%
STC	21%	87.2%	37%	-
MAC	-	85%	-	65%
SRC	19%	86%	54%	46%
SCL	**15**%	**92**%	41%	63.8%
Dense STC	16%	89%	58%	41.7%
SF	31%	67.5%	79%	19.7%
SF+MPPCA	32%	67%	71%	21.3%
Adam *et al.*	38%	65%	76%	13.3%

Adam *et al.* [21]. While the AUC of [46] under frame-level evaluation is slightly better, the proposed approach achieves the best EER under both frame-level (12%)

and pixel-level criteria (17%) as well as the best AUC under pixel-level criteria (80.1%) among all of the methods, while the AUC under frame-level evaluation (91.3%) is the second best. As a consequence, the proposed approach reports satisfactory results for anomaly detection and localization tasks on both UCSD ped1 and UCSD ped2 datasets, especially in anomaly localization task.

Examples of detected anomalies on UCSD ped1 and UCSD ped2 datasets are presented in Fig. 20 and Fig. 21, and it can be easily discovered that our approach can detect and localize multiple different anomalies in both crowded and uncrowded scenes. We also spot one interesting result in the last image of Fig. 21: Despite that the man with a bike is walking at a normal speed rather than riding fast, our approach still detects the bike as an anomaly since the structure and texture of a bike is different from that of the foreground in training videos (pedestrians), which actually reflects the proposed video descriptors' capability in incorporating structural and texture information into our video representation.

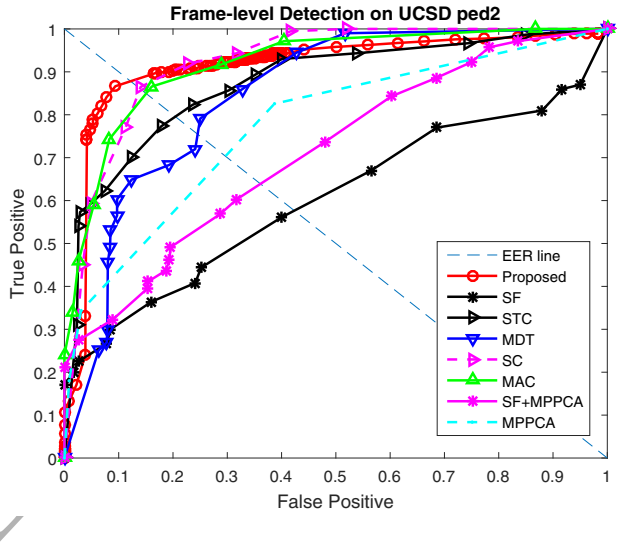


Figure 17: Frame-level anomaly detection on UCSD ped2 dataset.

5.5.2. UMN Dataset

We additionally test the proposed approach on UMN dataset, which is another widely used benchmark dataset for video anomaly detection. Since no pixel-level ground truth is provided like UCSD ped1 and ped2 dataset, we evaluate the performance of our approach only by the frame-level criteria. The EER, AUC and ROC

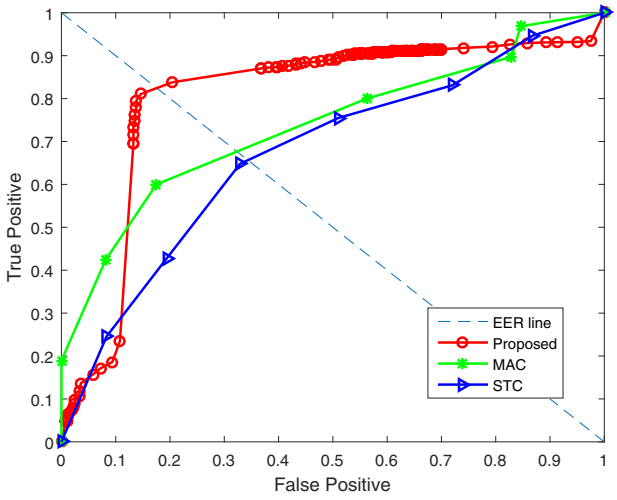


Figure 18: Pixel-level anomaly localization on UCSD ped2 dataset.

Table 5: Method comparison on UCSD ped2 dataset.

Method	EER (frame)	AUC	EER (pixel)	AUC
Proposed	**12**%	91.3%	**17**%	**80.1**%
SC	13%	**92**%	26%	-
STC	21%	89.1%	-	67.4%
MAC	-	90%	-	73.7%
MDT	25%	85%	-	-
HMDT+CRF	19%	-	30%	-
MPPCA	30%	77%	-	-
Bertini *et al.*	30%	-	68%	-
SF+MPPCA	36%	71%	-	-
Adam *et al.*	42%	63%	-	-

yielded by the proposed approach are compared with following state-of-the-art approaches: SRC [15], HMDT+CRF [23], Chaotic invariants [49], Local Statistics Aggregates (LSA) [50], SF [22]. The results are summarized in Fig. 19 and Table 6. As shown in Table 6, since the abnormal events in UMN are staged and the

Table 6: Detection results on UMN dataset.

Method	EER	AUC
Proposed	3.1%	99.0%
Chaotic invariants	5.3%	99.4%
HMDT+CRF	3.7%	99.5%
SRC	2.8%	99.6%
SF	12.6%	94.9%
LSA	3.4%	99.5%
Nearest neighbor	-	93%

type of its anomalies is much less than UCSD ped1 and ped2, the detection performance on UMN dataset is generally better than UCSD ped1 and ped2 datasets. Our approach achieved fairly comparable EER (3.1%) and AUC (99.0%) among the compared approaches. Examples of normal events and the detected abnormal events are shown in Fig. 22.

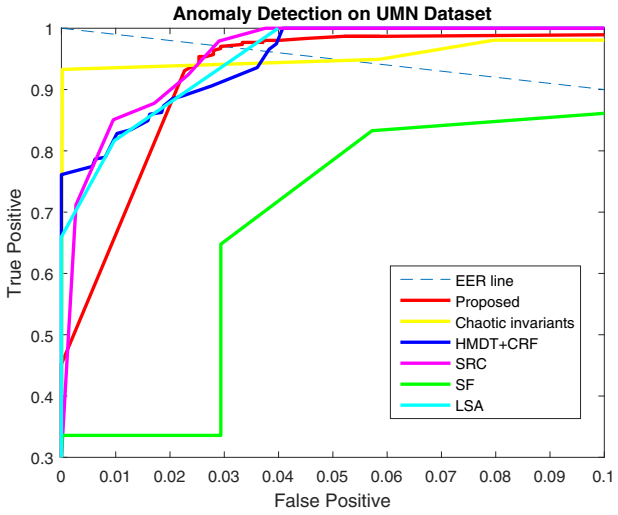


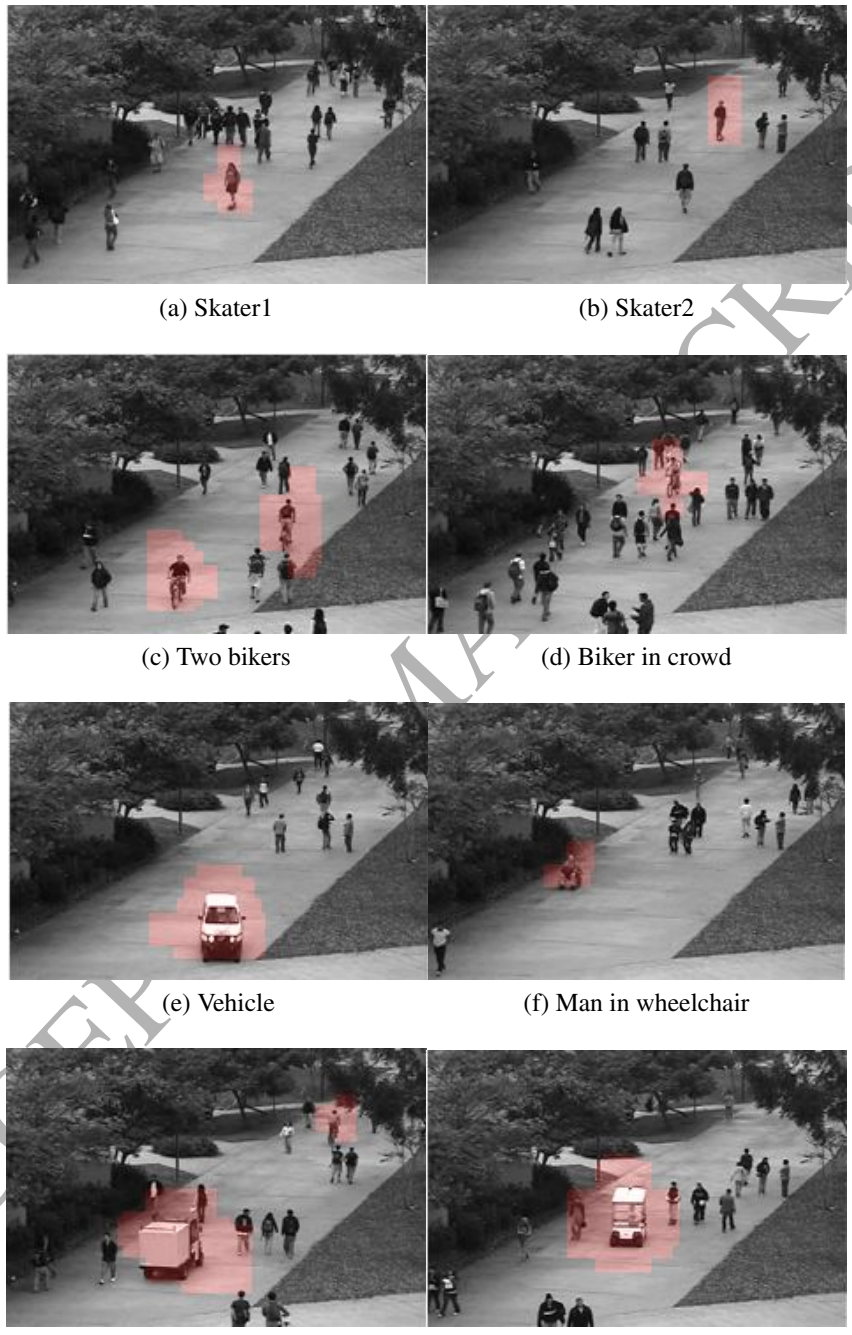
Figure 19: ROC curves of different methods on UMN datasets.

5.6. Computational Efficiency

We implement our algorithm under Matlab 2015b environment on a PC with 32 GB RAM and 3.90 Ghz Intel i7 4790 processor. It takes 1186.2s, 455.5s and 392.6s respectively to train normal event models for UCSD ped1, ped2 and UMN dataset. As for testing, the average processing time is 0.84s/frame for UCSD ped1 dataset, 1.66s/frame for UCSD ped2 dataset and 0.91s/frame for UMN dataset. To further accelerate the training and testing of the proposed approach, we can exploit the potential of parallel processing in terms of three aspects: First of all, the SL-HOF based video representation and ULGP-OF based video representation are independent, so they can be computed in parallel with each other instead of being computed sequentially. Secondly, multi-scale analysis is another major computational burden. The computation conducted on different video frame scales can be implemented in a parallel way. Thirdly, since only features that are extracted from the same spatial location on the video frame are used for training and testing, the training and testing of OCELM at different spatial locations can be paralleled rather than the time-consuming sequential processing in current implementation. Last but not least, a faster implementation like C++ will also boost the training and testing speed.

6. Conclusion

In this paper, we have proposed a novel video anomaly detection and localization approach by local motion based joint video representation and OCELM. We represent the motion of 3D local regions in spatio-temporal video cuboids by SL-HOF, which can implicitly capture the structural information of foreground object and depict foreground motion in a more accurate way. Combined with the new foreground localization scheme, the proposed ULGP-OF descriptor is used to characterize the motion of local texture within the video foreground. SL-HOF and ULGP-OF features extracted from training video volumes are modeled by OCELM, which enables us to learn a good data description in a much faster way than other data description algorithms like OCSVM and sparse coding. Experiments on public datasets show our approach can achieve state-of-the-art results on both anomaly detection and localization task. In our future work, we will explore applying hierarchical ELM-autoencoder to video analysis for a high speed automatic video representation learning. Ensemble OCELM will also be studied to describe data with several subclasses or clusters, which may further enhance OCELM's performance in data description.



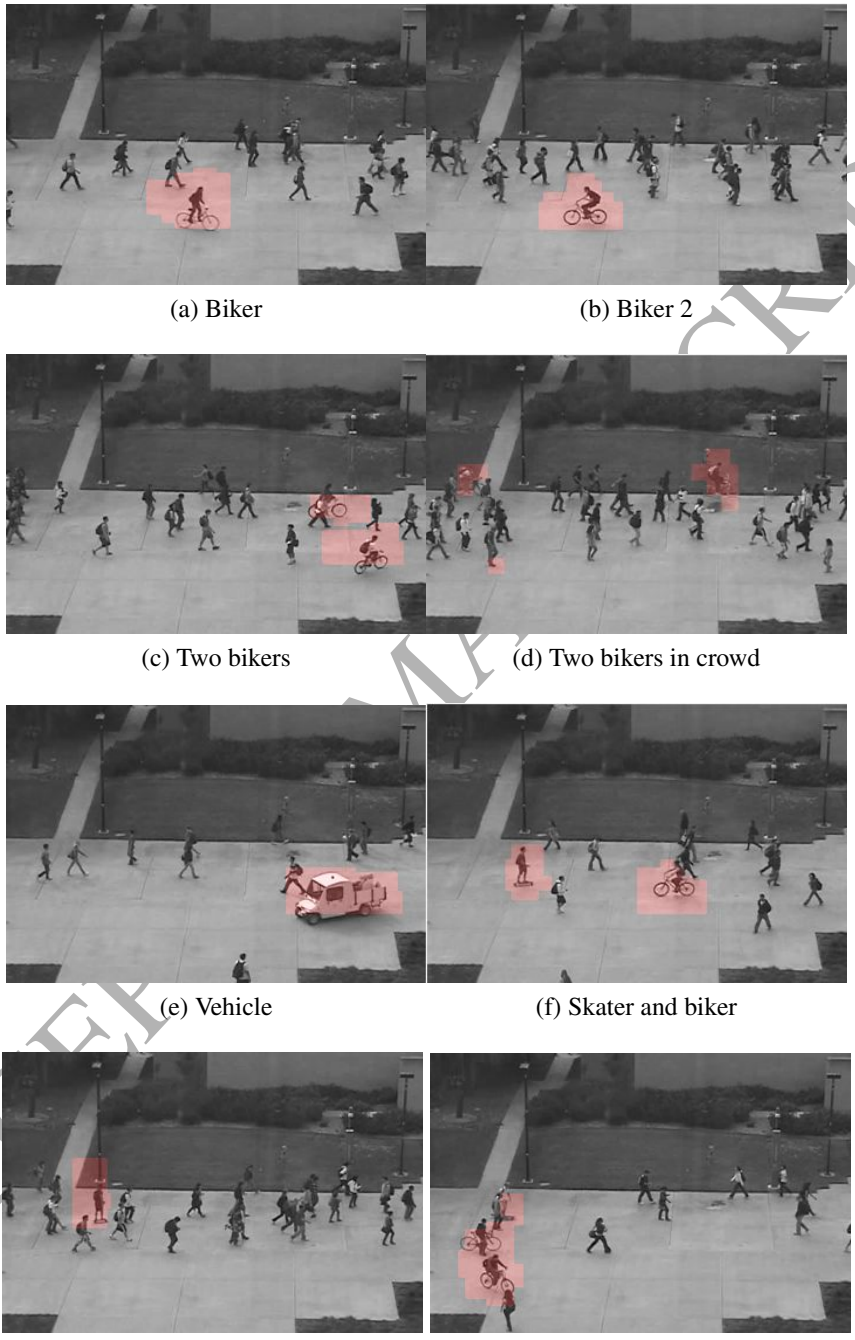
(a) Skater1 (b) Skater2

(c) Two bikers (d) Biker in crowd

(e) Vehicle (f) Man in wheelchair

(g) Vehicle and biker (h) Vehicle and slow skater

Figure 20: Different anomalies detected on UCSD ped1 dataset.



(a) Biker (b) Biker 2

(c) Two bikers (d) Two bikers in crowd

(e) Vehicle (f) Skater and biker

(g) Skater in crowd (h) Biker, man with a bike, skater

Figure 21: Different anomalies detected on UCSD ped2 dataset.

Figure 22: Normal events and detected abnormal events on UMN dataset.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Project No. 60970034, 61170287, 61232016).

References

[1] P. C. Chung, C. D. Liu, A daily behavior enabled hidden markov model for human behavior understanding, Pattern Recognition 41 (5) (2008) 1572–1580.

[2] H. Foroughi, A. Rezvanian, A. Paziraee, Robust fall detection using human shape and multi-class support vector machine, in: Proceedings of Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), IEEE, 2008, pp. 413–420.

[3] A. Sodemann, M. P. Ross, B. J. Borghetti, et al., A review of anomaly detection in automated surveillance, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (6) (2012) 1257–1272.

[4] Z. Fu, W. Hu, T. Tan, Similarity based vehicle trajectory clustering and anomaly detection, in: Proceedings of IEEE International Conference on Image Processing (ICIP), Vol. 2, IEEE, 2005, pp. II–602.

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

[6] A. Basharat, A. Gritai, M. Shah, Learning object motion patterns for anomaly detection and improved object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2008, pp. 1–8.

[7] T. Zhang, H. Lu, S. Z. Li, Learning semantic scene models by object classification and trajectory clustering, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 1940–1947.

[8] B. Jun, D. Kim, Robust face detection using local gradient patterns and evidence accumulation, Pattern Recognition 45 (9) (2012) 3304–3316.

[9] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 1975–1981.

[10] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1446–1453.

[11] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 fps in matlab, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), IEEE, 2013, pp. 2720–2727.

[12] Y. Zhang, H. Lu, L. Zhang, R. Xiang, Combining motion and appearance cues for anomaly detection, Pattern Recognition 51 (2016) 443–452.

[13] M. J. Roshtkhari, M. D. Levine, Online dominant and anomalous behavior detection in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 2611–2618.

[14] Y. Zhao, Y. Qiao, J. Yang, N. Kasabov, Abnormal activity detection using spatio-temporal feature and laplacian sparse representation, in: Neural Information Processing, Springer, 2015, pp. 410–418.

[15] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3449–3456.

[16] K.-W. Cheng, Y.-T. Chen, W.-H. Fang, Video anomaly detection and localization using hierarchical feature representation and gaussian process regression, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2909–2917.

[17] X. Zhu, J. Liu, J. Wang, C. Li, H. Lu, Sparse representation for robust abnormality detection in crowded scenes, Pattern Recognition 47 (5) (2014) 1791–1799.

[18] C. Piciarelli, G. L. Foresti, Surveillance-oriented event detection in video streams, IEEE intelligent systems (3) (2010) 32–41.

[19] D. H. Hu, X.-X. Zhang, J. Yin, V. W. Zheng, Q. Yang, Abnormal activity recognition based on hdp-hmm models., in: IJCAI, 2009, pp. 1715–1720.

[20] J. Yin, Q. Yang, J. J. Pan, Sensor-based abnormal human-activity detection, IEEE Transactions on Knowledge and Data Engineering 20 (8) (2008) 1082–1090.

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

[22] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 935–942.

[23] W. Li, V. Mahadevan, N. Vasconcelos, Anomaly detection and localization in crowded scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 36 (1) (2014) 18–32.

[24] T. Ojala, M. Pietikinen, T. Menp, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 971–987.

[25] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.

[26] G. B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification., IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) 42 (42) (2012) 513–29.

[27] G. Huang, S. Song, J. N. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines., IEEE Transactions on Cybernetics 44 (12) (2014) 1–1.

[28] L. Nan-Ying, H. Guang-Bin, . Saratchandran, P., . Sundararajan, N., A fast and accurate online sequential learning algorithm for feedforward networks., IEEE Transactions on Neural Networks 17 (6) (2006) 1411–1423.

[29] Y. Wang, Z. Xie, K. Xu, Y. Dou, Y. Lei, An efficient and effective convolutional auto-encoder extreme learning machine network for 3d feature learning, Neurocomputing 174 (2016) 988–998.

[30] H. Zhou, G. B. Huang, Z. Lin, W. Han, Y. C. Soh, Stacked extreme learning machines, Cybernetics IEEE Transactions on 45 (9) (2014) 1.

[31] Y. Yang, Q. M. J. Wu, Multilayer extreme learning machine with subnetwork nodes for representation learning, IEEE Transactions on Cybernetics.

[32] J. Cao, Y. Zhao, X. Lai, M. E. H. Ong, C. Yin, X. K. Zhi, N. Liu, Landmark recognition with sparse representation classification and extreme learning machine, Journal of the Franklin Institute 352 (10) (2015) 4528–4545.

[33] Z. Bai, G. B. Huang, Generic object recognition with local receptive fields based extreme learning machine , Procedia Computer Science 53 (1) (2015) 391–399.

[34] S. Decherchi, P. Gastaldo, R. Zunino, E. Cambria, J. Redi, Circular-elm for the reduced-reference assessment of perceived image quality, Neurocomputing 102 (2) (2013) 78–89.

[35] K. Choi, K. A. Toh, H. Byun, Incremental face recognition for large-scale social network services, Pattern Recognition 45 (8) (2012) 2868–2883.

[36] Z. Xie, K. Xu, W. Shan, L. Liu, Y. Xiong, H. Huang, Projective feature learning for 3d shapes with multiview depth images, Computer Graphics Forum 34 (7).

[37] Q. Leng, H. Qi, J. Miao, W. Zhu, G. Su, One-class classification with extreme learning machine, Mathematical Problems in Engineering 2015 (2015) 1–11.

[38] D. Sun, S. Roth, M. J. Black, Secrets of optical flow estimation and their principles, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2432–2439.

[39] Y. Li, E. Zhu, J. Zhao, J. Yin, A fast simple optical flow computation approach based on the 3-d gradient, IEEE Transactions on Circuits and Systems for Video Technology 24 (5) (2014) 842–853.

[40] A. Klaser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: BMVC, 2008.

[41] J. Wright, A. Ganesh, S. Rao, M. Yi, Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization., Advances in Neural Information Processing Systems 87 (4) (2009) 20:320:56.

[42] M. Topi, O. Timo, P. Matti, S. Maricor, Robust texture classification by subsets of local binary patterns, in: International Conference on Pattern Recognition, 2000, pp. 3947–3947.

[43] A. Klaser, M. Marszaek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Machine Vision Conference (BMVC), British Machine Vision Association, 2008, pp. 275–1.

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

[45] N. Li, X. Wu, D. Xu, H. Guo, W. Feng, Spatio-temporal context analysis within video volumes for anomalous-event detection and localization, Neurocomputing 155 (2015) 309–319.

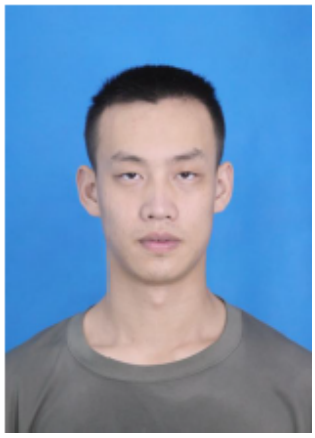
[46] M. J. Roshtkhari, M. D. Levine, An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions, Computer Vision and Image Understanding 117 (10) (2013) 1436–1452.

[47] J. Kim, K. Grauman, Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 2921–2928.

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

[49] S. Wu, B. E. Moore, M. Shah, Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2054–2060.

[50] Z. Chen, V. Saligrama, Video anomaly detection based on local statistical aggregates 157 (10) (2012) 2112–2119.



Siqi Wang is a PhD candidate of National University of Defense Technology, P. R. China. He received Bachelor of Engineering from National University of Defense Technology in 2014. His research interests include machine learning, computer vision and pattern recognition.

En Zhu received his M.S. degree and the PhD degree in computer science from the National University of Defense Technology, Changsha, China, in 2001 and 2005, respectively. From July 2009 to July 2010, he visited the Department of Computer Science, The University of York, York, U.K. He is currently a professor at College of Computer, National University of Defense Technology. His main research interests are pattern recognition, image processing and machine learning

Jianping Yin received Ph. D. degree in computer science and technology from the National University of Defense Technology (NUDT) in 1990. He currently holds the positions of professor and the head of China Computer Federation Technical Committee on Theoretical Computer Science. His research interests include artificial intelligence, pattern recognition and network algorithm.

Fatih Porikli is an IEEE Fellow and a Professor with the Research School of Engineering, Australian National University, Canberra, ACT, Australia. He is also acting as the Leader of the Computer Vision Group at NICTA, Sydney, NSW, Australia. He received the Ph.D. degree from NYU, New York, NY, USA, in 2002. Previously he served as a Distinguished Research Scientist at Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He has contributed broadly to object detection, motion estimation, tracking, image-based representations, and video analytics. He is the coeditor of two books on Video Analytics for Business Intelligence and Handbook on Background Modeling and Foreground Detection for Video Surveillance. He is an Associate Editor of five journals including IEEE Signal Processing Magazine, SIAM Imaging Sciences, EURASIP Journal of Image & Video Processing, Springer Journal on Machine Vision Applications, and

Springer Journal on Real-time Image & Video Processing. His publications won three best paper awards and he has received the R&D100 Award in the Scientist of the Year category in 2006. He served as the General and Program Chair of several IEEE conferences in the past.