

# Detecting abnormal events in video using Narrowed Motion Clusters

Radu Tudor Ionescu<sup>1,2</sup>, Sorina Smeureanu<sup>1,2</sup>,  
Marius Popescu<sup>1,2</sup>, and Bogdan Alexe<sup>1,2</sup>

<sup>1</sup>University of Bucharest, 14 Academiei, Bucharest, Romania

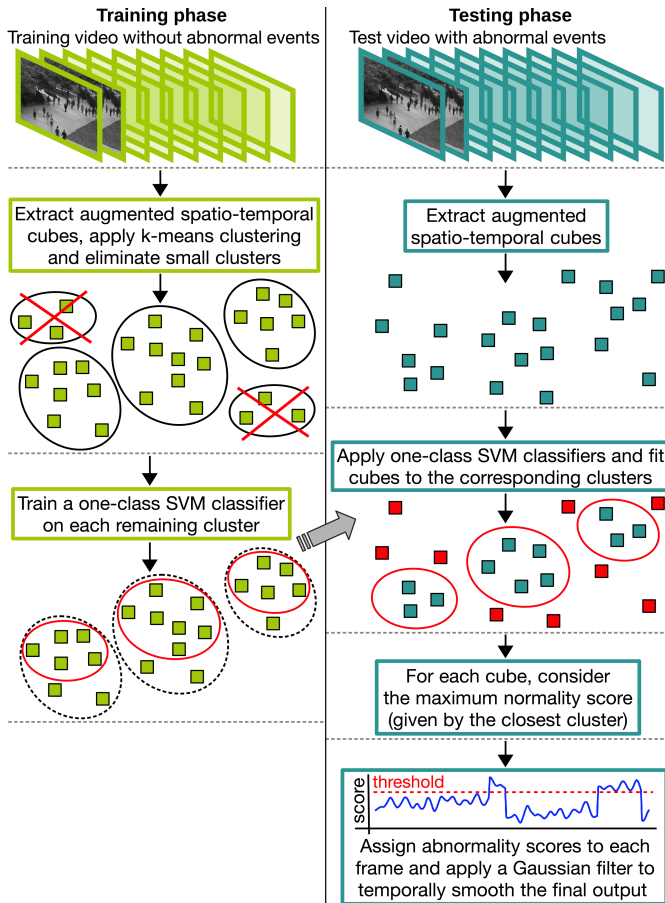
<sup>2</sup>SecurifAI, 24 Mircea Vodă, Bucharest, Romania

**Abstract.** We formulate the abnormal event detection problem as an outlier detection task and we propose a two-stage algorithm based on k-means clustering and one-class Support Vector Machines (SVM) to eliminate outliers. After extracting motion features from the training video containing only normal events, we apply k-means clustering to find clusters representing different types of motion. In the first stage, we consider that clusters with fewer samples (with respect to a given threshold) contain only outliers and we eliminate these clusters altogether. In the second stage, we shrink the borders of the remaining clusters by training a one-class SVM model on each cluster. To detect abnormal events in the test video, we analyze each test sample and consider its maximum normality score provided by the trained one-class SVM models, based on the intuition that a test sample can belong to only one cluster of normal motion. If the test sample does not fit well in any narrowed cluster, then it is labeled as abnormal. We also combine our approach based on motion features with a recent approach based on deep appearance features extracted with pre-trained convolutional neural networks (CNN). We combine our two-stage algorithm with the deep framework using a late fusion strategy, keeping the pipelines of the two approaches independent. We compare our method with several state-of-the-art supervised and unsupervised methods on four benchmark data sets. The empirical results indicate that our abnormal event detection framework can achieve better results in most cases, while processing the test video in real-time at 32 frames per second on CPU.

**Keywords:** Abnormal event detection, k-means, one-class SVM.

## 1 Introduction

Abnormal event detection in video is a challenging task in computer vision, since it is extremely hard, if not impossible, to define abnormal events independent of the context. For example, a truck driving by on the street is regarded as a normal event, but if the truck enters a pedestrian area, then it is regarded as an abnormal event. Two persons fighting in a box ring (normal event) versus fighting on the street (abnormal event) is another example. Although what is considered abnormal depends on the context, we can generally agree that abnormal behaviour should rather be represented by unexpected events [1] that occur



**Fig. 1.** Our anomaly detection framework based on Narrowed Motion Clusters. In the training phase, we apply a two-stage outlier detection algorithm based on k-means and one-class SVM. In the testing phase, we label a test sample as abnormal if its maximum normality score among the scores provided by the trained one-class SVM models is negative. Best viewed in color.

less often than familiar (normal) events. As it is generally impossible to find a sufficiently representative set of anomalies, the use of traditional supervised learning methods is usually ruled out. Hence, most abnormal event detection approaches [2, 3, 4, 5, 6, 7, 8, 9, 10] learn a model of familiarity from training video and label events as abnormal if they deviate from the model. We approach abnormal event detection in a similar manner, and propose to build a model of normality using a two-stage outlier detection algorithm illustrated in Figure 1. We first extract spatio-temporal cubes [6, 11, 12], which we augment with additional information about location and motion direction. After extracting augmented spatio-temporal cubes from the training video containing only normal events, we apply k-means clustering to find clusters representing different types of motion. In the first stage, we eliminate the clusters with fewer samples (with

respect to a pre-defined threshold), based on the hypothesis that these smaller clusters contain predominantly outliers. Different from other outlier detection approaches based on k-means [13, 14, 15], we do not modify the clustering algorithm. Instead, we propose a simple and straightforward approach that aims to coarsely remove some of the outliers, leaving the outliers that are harder to pinpoint for the second stage. In the second stage, we narrow down the borders of the remaining clusters by training a one-class Support Vector Machines (SVM) classifier on each individual cluster. In the end, the learned one-class SVM models represent narrowed clusters of different types of normal motion. We therefore coin the term *Narrowed Motion Clusters (NMC)* for our two-stage outlier detection algorithm. To detected abnormal events in a test video, we analyze each augmented spatio-temporal cube and consider its maximum normality score among the scores provided by the trained one-class SVM models, based on the natural intuition that a test sample (spatio-temporal cube) should belong to a single narrowed cluster of normal motion. The novelty of our paper consists of (i) the idea of assembling together two popular approaches for outlier detection (k-means and one-class SVM) into a simple and fast framework and (ii) the idea of *narrowing* down the clusters by learning a tight border around each cluster.

We perform abnormal event detection experiments on the Avenue [6], the Subway [16] and the UMN [8] data sets in order to compare our NMC approach with several state-of-the-art abnormal event detection methods [3, 6, 8, 11, 12, 17, 18, 19, 20, 21, 22, 23, 24, 25]. The empirical results indicate that, on two of the test sets (Avenue and Subway), we obtain better results than all these approaches. It is important to mention that our approach yields impressive results, while running in real-time at 32 frames per second on CPU. We also combine our approach based on motion features with a recent approach [23] based on deep appearance features extracted with pre-trained convolutional neural networks (CNN). In this approach, the CNN features are fed into a one-class SVM classifier in order to learn a model of normality from training data. We combine our NMC algorithm with the framework based on appearance features presented in [23] using a late fusion strategy, keeping the pipelines of the two approaches independent. By combining motion and appearance features, we are able to further improve the accuracy on all data sets.

We organize the paper as follows. We present related work on abnormal event detection in Section 2. We describe our outlier detection framework in Section 3. We present the abnormal event detection experiments in Section 4. Finally, we draw our conclusions in Section 5.

## 2 Related Work

Abnormal event detection is usually formalized as an outlier detection task [2, 3, 4, 5, 6, 7, 8, 9, 10, 17, 24, 25, 26, 27], in which the general approach is to learn a model of normality from training data and consider the detected outliers as abnormal events. Some abnormal event detection approaches [3, 6, 17, 26, 27] are based on learning a dictionary of atoms representing normal events, and label the events not represented by the dictionary as abnormal. At a conceptual level,

we can find some resemblance between our approach and dictionary learning. However, going down to the implementation level, there are some important differences. We can interpret the use of k-means to group the training samples into clusters as an unconventional way of building a dictionary of atoms. To the best of our knowledge, there are no dictionary learning approaches that try to remove a part of the atoms as outliers. Unlike dictionary learning approaches, we eliminate the smaller clusters in our framework. Another difference is that we consider that a test sample can belong to a single cluster, or in other words, it can be reconstructed by a single atom. Hence, instead of using the reconstruction error given by a set of basis vectors as the abnormality score, we consider the maximum normality score among the scores given by a set of one-class SVM models, each trained on a different cluster.

Recent abnormal event detection approaches have employed locality sensitive hashing filters [25] or deep features [9, 18, 19, 20, 21, 23] to achieve better results. Hasan et al. [18] propose two autoencoders, one that is learned on conventional handcrafted spatio-temporal local features, and another one that is learned end-to-end using a fully convolutional feed-forward architecture. Hinami et al. [19] proposed to train convolutional neural networks on multiple visual tasks to exploit semantic information that is useful for detecting and recounting abnormal events, while Smeureanu et al. [23] simply applied convolutional neural networks (CNN) pre-trained on the ILSVRC benchmark [28]. Luo et al. [20] propose a Temporally-coherent Sparse Coding approach, which can be mapped to a stacked Recurrent Neural Network which facilitates the parameter optimization and accelerates the anomaly prediction. The approach presented in [21] is based on training Generative Adversarial Nets (GAN) using normal frames and corresponding optical-flow images in order to learn an internal representation of the scene normality. The test data is compared with both the appearance and the motion representations reconstructed by the GAN and abnormal areas are detected by computing local differences.

There have been some approaches that employ unsupervised steps for abnormal event detection [9, 24, 26, 27]. The approach presented in [27] is to build a model of familiar events from training data and incrementally update the model in an unsupervised manner as new patterns are observed in the test data. In a similar fashion, Sun et al. [24] train a Growing Neural Gas model starting from training videos and continue the training process as they analyze the test videos for anomaly detection. Ren et al. [26] use an unsupervised approach, spectral clustering, to build a dictionary of atoms, each representing one type of normal behavior. Their approach requires training videos of normal events to construct the dictionary. Xu et al. [9] use Stacked Denoising Auto-Encoders to learn deep feature representations in an unsupervised way. However, they still employ one-class SVM to predict the anomaly scores. There are some works that do not require any kind of training data for abnormal event detection [11, 12]. The approach proposed by Del Giorno et al. [11] is to detect changes on a sequence of data from the video to see which frames are distinguishable from all the previous frames. As the authors want to build an approach independent

of temporal ordering, they create shuffles of the data by permuting the frames before running each instance of the change detection. Ionescu et al. [12] apply unmasking, a technique based on training a binary classifier to distinguish between two consecutive video sequences while removing at each step the most discriminant features. Their hypothesis is that higher training accuracy rates of the intermediately obtained classifiers represent abnormal events.

Regarding the feature representation, we use spatio-temporal cubes to represent motion, as other recent approaches [6, 11, 12]. Unlike all these approaches, we propose to augment each cube with its location within a spatial pyramid applied over the video frames, and with the mean direction given by the 3D motion gradients inside the cube. Our experiments show that the augmentation is useful. Regarding the outlier detection approach, there are a few works [13, 29] that applied k-means clustering for abnormal event detection. Abuolaim et al. [29] use k-means at a coarse level to divide the data points into precisely three clusters: normal, abnormal and ambiguous. On the other hand, we apply k-means with a completely different purpose, namely to obtain many clusters representing different types of normal motion. Moreover, their approach does not allow to set an abnormality threshold. More closely to our approach, Auslander et al. [13] define three possible assumptions (see Section 4.1 in [13]) for using clustering to detect anomalies. Interestingly, our approach is based on similar assumptions. However, their approach adopts only the first two assumptions defined in [13], while we satisfy the second assumption by eliminating smaller clusters (in the first stage), and the first and third assumptions by training a one-class SVM on each cluster (in the second stage).

### 3 Method

We propose an abnormal event detection framework based on a two-stage algorithm for outlier detection. Our anomaly detection framework is divided into a training phase and testing phase, as illustrated in Figure 1. We next provide an overview of our approach, leaving the additional details about the more important steps for later. From both training and testing videos, we extract spatio-temporal cubes. In the training phase, we cluster the extracted spatio-temporal cubes using k-means and we eliminate the smaller clusters as outliers. On each remaining cluster, we train a one-class SVM model to remove outlier cubes. In the testing phase, each spatio-temporal cube is tested against each one-class SVM model to obtain a set of normality scores. The maximum score is used (with a change of sign) as the abnormality score for the respective test cube. By putting together the cubes from an entire frame, we obtain an anomaly prediction map for each frame. To obtain pixel-level anomaly predictions, the prediction map can be simply resized to match the size of the input video frame. To obtain frame-level predictions, we consider the highest score in the prediction map as the anomaly score of the respective frame. We then apply a Gaussian filter to temporally smooth the final frame-level anomaly scores.

### 3.1 Feature Extraction

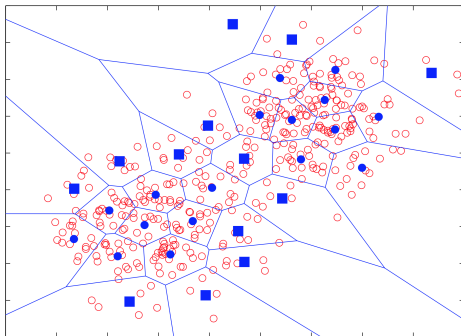
**Encoding motion.** Given the input video, we uniformly partition each frame to a set of non-overlapping  $10 \times 10$  patches. Corresponding patches in 5 consecutive frames are stacked together to form a spatio-temporal cube, each with resolution  $10 \times 10 \times 5$ . We then compute 3D gradient features on each spatio-temporal cube and normalize the resulted feature vectors using the  $L_2$ -norm. Until this point, our approach of representing motion is essentially the same as [6, 11, 12]. Similar to [6, 11, 12], we eliminate cubes in a region, if the video is static in the respective region. Different from [6, 11], we do not employ Principal Component Analysis to reduce the feature vector dimension from 500 to 100 components. Moreover, we diverge from standard spatio-temporal cube representation by augmenting the cubes with additional information about location and motion direction, as described next.

**Encoding location.** We divide each frame into a spatial pyramid [30] with two levels, the first level containing  $2 \times 2$  bins and the second one containing  $4 \times 4$  bins. We encode the location of each spatio-temporal cube as a one-hot vector for each level of the pyramid. This gives 20 additional features ( $2 \times 2 + 4 \times 4$ ) for each cube. The purpose of recording spatial information into the cube representation is to accurately detect situations in which abnormal events can appear in only some region of the video. For instance in a traffic surveillance video, people crossing the street on a crosswalk is a normal event, but if they cross it outside the designated area this should be labeled as abnormal.

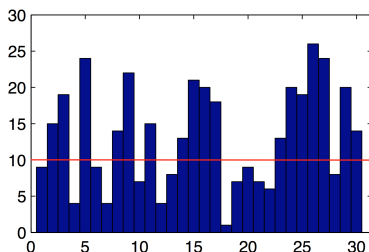
**Encoding mean direction.** To extract the mean motion direction from each spatio-temporal cube, we first consider the individual patches of the cube. In each patch, we compute the center of mass of the 3D gradients. We then encode the displacement of the center of mass in consecutive patches as vectors representing motion direction. For a better estimation of the mean motion direction, we also compute motion direction vectors after dividing each patch into  $2 \times 2$  bins. Finally, the motion direction vectors are quantized into an orientation-based histogram with 8 bins. The histogram bins are evenly spread over 0 to 360 degrees. Our histogram is produced in a similar way to the histogram corresponding to a cell in the Histograms of Oriented Gradients (HOG) descriptor [31]. Along with the histogram, we add another feature given by the sum of all vector magnitudes. In total, there are 9 additional features for augmenting the cube. The purpose of recording the mean direction into the cube representation is to enable the accurate detection of abnormal events triggered by objects moving in a certain direction. For example in a traffic surveillance video, a car driving the wrong way should be labeled as abnormal.

### 3.2 Two-Stage Outlier Detection

**First stage detection based on k-means.** We cluster the augmented spatio-temporal cubes extracted from the training video to find clusters representing different types of motion. Next, we eliminate the clusters with fewer samples, based on the assumption that these smaller clusters contain mostly outlier samples. We note that the same assumption also sits at the basis of the method



**Fig. 2.** A set of 400 data points sampled from two normal distributions of different means. The points are clustered into 30 clusters using k-means. The centroids of clusters with less than 10 samples are represented with a large blue square. Best viewed in color.



**Fig. 3.** A histogram representing the number of data points in each cluster. The histogram corresponds to the k-means clustering applied over the 400 data points illustrated in Figure 2. A threshold of 10 is used to detect clusters of outliers. Best viewed in color.

proposed in [13]. Nonetheless, we motivate the assumption through the following toy example. We generate 400 data points sampled from two normal distributions of different means. We group the points into  $k = 30$  clusters using k-means and we illustrate the result in Figure 2. We then count the number of points in each cluster and obtain the histogram depicted in Figure 3. In this example, we consider that the clusters with less than 10 data points contain mostly outliers. The centroids of these smaller clusters are marked with a large blue square in Figure 2. We can clearly see that the marked clusters are farthest from both normal distribution means, indicating that the containing points are indeed outliers. Nevertheless, our aim is to test out the assumption on real data for abnormal event detection in video. Although the training does not contain abnormal events, we believe that k-means helps to remove noisy or weak patterns that can be observed in the normal video.

**Second stage detection based on one-class SVM.** After removing the smaller k-means clusters, we are left with a set of clusters  $\mathcal{C} = \{c_1, c_2, \dots, c_r \mid r \leq k\}$  that accurately model the stronger patterns of normality. However, k-means does not provide a tight boundary around the remaining clusters, and, in some cases, it leaves a lot of room to accommodate outliers. For example, the borders of some remaining clusters represented in Figure 2 span to infinity. To alleviate

this problem, we propose to narrow down the borders of the remaining clusters by training a one-class SVM [32] classifier on each cluster. We note that the border learned by SVM is tighter (or narrower) than the border of the original cluster (which includes all cluster’s samples), since the one-class SVM model is forced to single out a small percentage of samples within the cluster as outliers. In this sense, we can say that one-class SVM narrows (or tightens) the border around the cluster’s centroid. Hence, the learned one-class SVM models can be interpreted as a set of narrowed clusters representing different types of normal motion. To train our set of classifiers, we consider each spatio-temporal cube as an individual and independent sample, disregarding the temporal relations between cubes. Let  $\mathcal{X} = \{x_1, x_2, \dots, x_n \mid x_i \in \mathbb{R}^m\}$  denote the set of training cubes in a given cluster  $c_j$ . In this formulation, our one-class SVM model will learn to separate a small region capturing most of the normal cubes from the rest of feature space, by maximizing the distance from the separating hyperplane to the origin. This results in a binary classification function  $g$  which captures a region in the input space where the probability density of a particular type of normal motion lives:

$$g(x) = \text{sign} \left( \sum_{i=1}^n \alpha_i k(x, x_i) - \rho \right), \quad (1)$$

where  $x$  is a test cube that needs to be classified either as normal or abnormal,  $x_i \in \mathcal{X}$  is a training cube,  $\alpha_i$  are the weights assigned to the support vectors  $x_i$ ,  $\rho$  is the distance from the hyperplane to the origin, and  $k$  is a kernel function, in our case, the linear kernel. If we desire a score reflecting the normality level of a spatio-temporal cube, we can simply remove the sign transfer function from Equation (1) and obtain a scoring function. It is important to note that for each cluster  $c_j \in \mathcal{C}$ , we have a different scoring function  $g_{c_j}$ . Then, for a given test cube, we will have a set of  $r$  normality scores. However, since the narrowed clusters are independent (they reside in different areas of the feature space), we can naturally assume that a spatio-temporal cube belongs to a single cluster. Therefore, we consider the maximum normality score, the one that corresponds to the narrowed cluster that better fits (is closer to) the test cube. If the test spatio-temporal cube does not fit well in any normality cluster, its corresponding maximum normality score will be negative (the cube is outside the nearest cluster). Consequently, the respective test sample is labeled as abnormal.

## 4 Experiments

### 4.1 Data Sets

**Avenue.** We first consider the Avenue data set [6], which contains 16 training and 21 test videos. In total, there are 15328 frames in the training set and 15324 frames in the test set. Each frame is  $640 \times 360$  pixels. Locations of anomalies are annotated in ground truth pixel-level masks for each frame in the testing videos. Hinami et al. [19] argue that the Avenue test set contains five videos (1, 2, 8, 9 and 10) with static abnormal objects that are not properly annotated.



Hence, they evaluate their approach on a subset (Avenue17) that excludes these five videos. When we compare our results with those reported in [19], we also remove the five videos for a fair comparison.

**Subway.** One of the largest data sets for anomaly detection in video is the Subway surveillance data set [16]. It contains two videos, one of 96 minutes (Entrance gate) and another one of 43 minutes (Exit gate). The Entrance gate video contains 144251 frames and the Exit gate video contains 64903 frames, each with  $512 \times 384$  resolution. Abnormal events are labeled at the frame level.

**UMN.** The UMN Unusual Crowd Activity data set [8] consists of three different crowded scenes, each with 1453, 4144, and 2144 frames, respectively. The resolution of each frame is  $320 \times 240$  pixels. In the normal settings people walk around in the scene, and the abnormal behavior is defined as people running in all directions.

## 4.2 Evaluation

As evaluation metrics, we employ ROC curves and the corresponding *area under the curve* (AUC) computed with respect to ground truth frame-level annotations, and, when available (Avenue), pixel-level annotations. We define the frame-level and pixel-level AUC as in previous works [6, 7, 9, 11, 12, 17, 24]. At the frame-level, a frame is considered a correct detection if it contains at least one abnormal pixel. At the pixel-level, the corresponding frame is considered as being correctly detected if more than 40% of truly anomalous pixels are detected. We smooth the pixel-level detection maps with the same filter used by [6, 11, 12] in order to obtain our final pixel-level detections. We exclude the recent approach (based on k-means) of Abuolaim et al. [29] from our evaluation, because their approach is constrained to provide a single point on the ROC curve and the frame-level or pixel-level AUC metrics cannot be determined in their case. Many works [6, 7, 9, 17, 25, 27] include the Equal Error Rate (EER) as evaluation metric, but some recent works [11, 12] argue that metrics such as the EER can be misleading in a realistic anomaly detection setting, in which abnormal events are expected to be very rare. As we agree with perspective of [11, 12], we do not employ the EER in our evaluation.

## 4.3 Implementation Details

We extract spatio-temporal cubes from the training and test video sequences using the code available online at <https://alliedel.github.io/anomalydetection/>. We use our own implementation to augment the cubes with location and mean direction. To cluster the cubes, we employ the k-means implementation from VLFeat [33] based on the original Lloyd algorithm [34]. We use k-means++ [35] initialization. We repeat the clustering 10 times and choose the partitioning with the minimum energy. We choose the number of clusters  $k$  such that we have on average 1000 cubes per cluster, hence  $k$  is proportional to the size of the training data. We then eliminate the clusters with less than 500 cubes. To remove the outliers from each cluster, we employ the one-class SVM implementation from

Method	Frame AUC	Pixel AUC
Lu et al. [6]	80.9	92.9
Hasan et al. [18]	70.2	-
Del Giorno et al. [11]	78.3	91.0
Smeureanu et al. [23]	84.6	93.5
Ionescu et al. [12]	80.6	93.0
Luo et al. [20]	81.7	-
cubes + one-class SVM	81.3	93.0
aug. cubes + one-class SVM	82.8	93.2
aug. cubes + k-means + one-class SVM	85.4	93.5
aug. cubes + k-means + 1-NN	77.6	90.6
Ours (NMC)	87.6	93.7
Ours (NMC + CNN)	87.8	93.8

**Table 1.** Abnormal event detection results (in %) in terms of frame-level and pixel-level AUC on the Avenue data set. Our framework and its preliminary versions are compared with several state-of-the-art approaches [6, 11, 12, 18, 20, 23], which are listed in temporal order.

Method	Frame AUC	Pixel AUC
Hinami et al. [19]	89.8	-
Ours (NMC)	90.2	93.9
Ours (NMC + CNN)	90.4	94.0

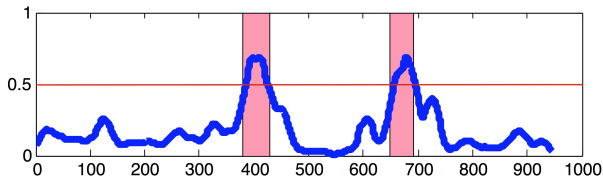
**Table 2.** Abnormal event detection results (in %) in terms of frame-level and pixel-level AUC on the Avenue17 data set. Our framework is compared with [19].

LibSVM [36]. In all the experiments, we set the regularization parameter of one-class SVM to 0.01, which means that the model will have to single out 99% of the training frames as normal (the other 1% are outliers).

We combine our approach based on motion features with the approach based on appearance features presented in [23]. For the appearance features, we consider the pre-trained VGG-f [37] model provided in MatConvNet [38], and extract the features from the *conv5* layer according to [23]. For a faster processing time, we extract features for one in every two frames in the test video, without observing any drop in performance. A one-class SVM classifier with regularization 0.2 is employed to learn a model of normal appearance from the training videos. The frame-level scores produced by this framework are assigned to the corresponding spatio-temporal cubes. The cube-level scores thus obtained are combined with the cube-level scores of our approach as a simple average. At test time, we are able to process the test videos at nearly 32 FPS using two parallel threads (one for each abnormal event detection framework) on a computer with Intel Core i7 2.3 GHz processor and 8 GB of RAM.

#### 4.4 Results on the Avenue Data Set

We first compare our approach with several state-of-the-art approaches [6, 11, 12, 18, 20, 23] that reported results on the Avenue data set. The corresponding frame-level and pixel-level AUC metrics are presented in Table 1. The table also



**Fig. 4.** Frame-level anomaly detection scores (between 0 and 1) provided by our approach based on combining NMC and CNN, for test video 4 in the Avenue data set. The video has 947 frames. Ground-truth abnormal events are represented in pink, and our scores are illustrated in blue. Best viewed in color.



**Fig. 5.** True positive (top row) versus false positive (bottom row) detections of our framework based on NMC and CNN. Examples are selected from the Avenue data set. Best viewed in color.

includes results for preliminary versions of our approach to show the performance gain brought by each component. A basic approach based on spatio-temporal cubes and one-class SVM yields a frame-level AUC of 81.3%. When we augment the cubes, the frame-level AUC grows by 1.5%. Another 2.6% are gained when we employ k-means and train one-class SVM models on all  $k$  clusters. By removing the smaller clusters we obtain an improvement of 2.2% and reach a frame-level AUC of 87.6%. We also tested an approach that removes the smaller k-means clusters in the first stage, but replaces the one-class SVM in the second stage with a one nearest neighbor (1-NN) model based the Euclidean distance to the nearest cluster centroid. The obtained frame-level AUC is 77.6%, which is exactly 10% lower than the result of NMC. This ablation result shows the importance of using one-class SVM after k-means to learn a tight border around each cluster.

Using Narrowed Motion Clusters of spatial-temporal cubes alone, we are able to surpass the results reported in previous works in terms of frame-level and pixel-level AUC. When we combine NMC with the one-class SVM based on CNN features [23], the results slightly improve. Compared to the most recent works [12, 20, 23], our framework brings an improvement of more than 3.2% in terms of frame-level AUC. Since our framework is able to process the video online, we consider that our results on the Avenue data set are noteworthy.

We also compared our approach with [19] on the Avenue17 data set, a subset of the Avenue data set. Our frame-level AUC scores presented in Table 2 are better than those reported by Hinami et al. [19]. It is worth nothing that our approach yields better performance on the Avenue17 data set, indicating that the five removed test videos are actually more difficult than those left in Avenue17.

Method	Frame AUC		
	Entrance gate	Exit gate	Average
Cong et al. [17]	80.0	83.0	81.5
Saligrama et al. [22]	89.1	-	-
Cheng et al. [3]	92.7	-	-
Hasan et al. [18]	94.3	80.7	87.5
Del Giorno et al. [11]	69.1	82.4	75.8
Ionescu et al. [12]	71.3	86.3	78.8
Ours (NMC)	91.8	94.2	93.0
Ours (NMC + CNN)	92.4	94.5	93.5

**Table 3.** Abnormal event detection results (in %) in terms of frame-level AUC on the Subway data set. Our framework is compared with several state-of-the-art approaches [3, 11, 12, 17, 18, 22], which are listed in temporal order.



**Fig. 6.** True positive (top row) versus false positive (bottom row) detections of our framework based on NMC and CNN. Examples are selected from the Subway Entrance gate. Best viewed in color.

As Hinami et al. [19] observed, the removed videos contain abnormal objects that are not properly annotated, hence methods are prone to reach high false positive rates on these five test videos.

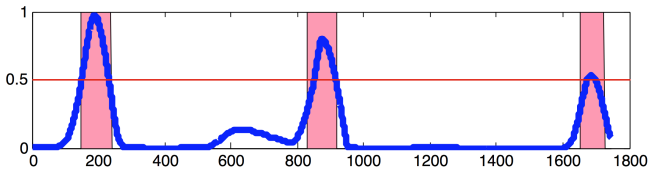
Figure 4 depicts the frame-level anomaly scores produced by our approach against the ground-truth labels on test video 4 of the Avenue data set. We notice that our scores correlate well to the ground-truth labels. There are two abnormal events in this video and we can easily identify both of them by setting a threshold of 0.5, without including any false positive detections. We also show some examples of true positive and false positive detections in Figure 5. The true positive abnormal events are (from left to right) *a person running*, *a child running* and *a person throwing an object*. The first (left-most) false positive detection represents two persons walking synchronously. The last two false positive example indicate that our method detects *a child running* even if the child is partially occluded, or *a person throwing an object* before the object is in the air.

#### 4.5 Results on the Subway Data Set

Although there are many works that used the Subway data set in the experiments [3, 6, 11, 12, 17, 18, 22, 25, 27], some of these works [6, 25, 27] did not use the frame-level AUC as evaluation metric. Therefore, we only compare our approach with those methods [3, 11, 12, 17, 18, 22] that reported the frame-level AUC. The results of the comparative study are reported in Table 3. On the Entrance

Method	Frame AUC			
	Scene 1	Scene 2	Scene 3	Average
Mehran et al. [8]	-	-	-	96.0
Cong et al. [17]	99.5	97.5	96.4	97.8
Saligrama et al. [22]	-	-	-	98.5
Del Giorno et al. [11]	-	-	-	91.0
Zhang et al. [25]	99.2	98.3	98.7	98.7
Sun et al. [24]	99.8	99.3	99.9	99.7
Smeureanu et al. [23]	98.8	93.6	98.9	97.1
Ionescu et al. [12]	99.3	87.7	98.2	95.1
Ravanbakhsh et al. [21]	-	-	-	99.0
Ours (NMC)	99.8	96.2	99.6	98.5
Ours (NMC + CNN)	99.9	96.8	99.8	98.8

**Table 4.** Abnormal event detection results (in %) in terms of frame-level AUC on the UMN data set. Our framework is compared with several state-of-the-art methods [8, 11, 12, 17, 21, 22, 23, 24, 25], which are listed in temporal order.



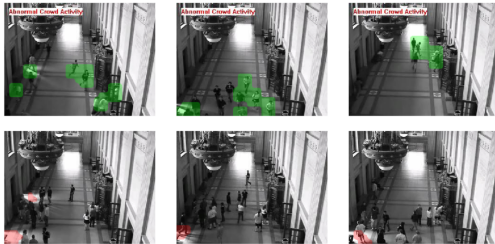
**Fig. 7.** Frame-level anomaly detection scores (between 0 and 1) provided by our framework based on the late fusion strategy, for the third scene in the UMN data set. The video has 1744 test frames. Ground-truth abnormal events are represented in pink, and our scores are illustrated in blue. Best viewed in color.

gate video, we obtain the third best score after [3, 18]. Hasan et al. [18] report a frame-level AUC of 94.3%, which is 1.9% higher than our best score (92.4%). Things look differently on the Exit gate video, as we obtain the best score (94.5%) among all methods, surpassing the approach of Hasan et al. [18] by 13.8% and the second best method [12] by 8.2%. On average, we obtain the best results on the Subway data set.

In Figure 6, we present some interesting qualitative results obtained by our framework on the Entrance gate video. The true positive abnormal events are *a person jumping over the fence*, *two persons walking in wrong direction* and *a person jumping over the gate*, while false positive detections are *a person running*, *two persons walking synchronously* and *camera shifting*.

#### 4.6 Results on the UMN Data Set

On the UMN data set, we compare our approach with several methods [8, 11, 12, 17, 21, 22, 23, 24, 25]. In Table 4, we report the frame-level AUC score for each individual scene, as well as the average score for all the three scenes. It is worth noting that UMN seems to be the easiest abnormal event detection data set, since most works report frame-level AUC scores above 95.0%. We reach the highest performance (99.9%) among all methods on the first scene. On the last



**Fig. 8.** True positive (top row) versus false positive (bottom row) detections of our framework based on NMC and CNN. Examples are selected from the UMN data set. Best viewed in color.

scene, we obtain the second best score (99.8%). Remarkably, our approach is able to correctly identify the three abnormal events in the third scene without any false positives, by applying a threshold of 0.5, as illustrated in Figure 7. Our lowest performance (96.8%) is on the second scene. Over all scenes, we reach the third highest frame-level AUC (98.8%), which is 0.9% lower than the best score (99.7%) obtained by Sun et al. [24].

In Figure 8, we present some interesting qualitative results obtained by our framework on the second scene, as it was almost impossible to find false positive detections in the other scenes. The true positive examples represent *people running around in all directions*, while the false detections are triggered by *people opening the doors* to enter or exit the room. In the first (left-most) false positive example it seems that our method detects the significant amount of light that enters the room as the doors open. Perhaps the first impression is that our approach is not robust to illumination variations. However, we noticed that our training video does not contain examples of people walking through the doors. Therefore, it is impossible to learn a complete model of normality that includes this kind of event (people walking through the doors).

## 5 Conclusion and Future Work

In this work, we proposed *Narrowed Motion Clusters*, a novel framework for abnormal event detection in video that is based on a two-stage outlier elimination algorithm. The algorithm works by removing outlier clusters obtained with k-means and by learning a tight border around each remaining cluster using one-class SVM. We also combined our approach based on motion features with a recent approach [23] based on CNN features in order to improve the performance. We conducted abnormal event detection experiments on four data sets to compare our approach with a series of state-of-the-art approaches [3, 6, 8, 11, 12, 17, 18, 19, 20, 21, 22, 23, 24, 25]. The empirical results indicate that our approach yields better performance than all other methods on the Avenue and the Subway data sets. Furthermore, our approach is in the top three methods on the UMN data set. In the same time, we can process the test video in real-time at 32 frames per second on CPU.

In future work, we aim to replace the approach of Smeureanu et al. [23] with a CNN that is fine-tuned on the abnormal event detection task. Alternatively,

we could develop an approach to train deep features on a related task, such as action recognition, and transfer the learned features to our task. We also aim to combine the spatio-temporal cubes with the learned deep features before the training stage, using an early fusion strategy. This could yield better results than the late fusion strategy applied in this work.

## References

1. Itti, L., Baldi, P.: A Principled Approach to Detecting Surprising Events in Video. In: Proceedings of CVPR. Volume 1., IEEE (June 2005) 631–637
2. Antic, B., Ommer, B.: Video parsing for abnormality detection. In: Proceedings of ICCV. (2011) 2415–2422
3. Cheng, K.W., Chen, Y.T., Fang, W.H.: Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. In: Proceedings of CVPR. (2015) 2909–2917
4. Kim, J., Grauman, K.: Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In: Proceedings of CVPR. (2009) 2921–2928
5. Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(1) (January 2014) 18–32
6. Lu, C., Shi, J., Jia, J.: Abnormal Event Detection at 150 FPS in MATLAB. In: Proceedings of ICCV. (2013) 2720–2727
7. Mahadevan, V., LI, W.X., Bhalodia, V., Vasconcelos, N.: Anomaly Detection in Crowded Scenes. In: Proceedings of CVPR. (2010) 1975–1981
8. Mehran, R., Oyama, A., Shah, M.: Abnormal crowd behavior detection using social force model. In: Proceedings of CVPR. (2009) 935–942
9. Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N.: Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. In: Proceedings of BMVC. (2015) 8.1–8.12
10. Zhao, B., Fei-Fei, L., Xing, E.P.: Online Detection of Unusual Events in Videos via Dynamic Sparse Coding. In: Proceedings of CVPR. (2011) 3313–3320
11. Del Giorno, A., Bagnell, J., Hebert, M.: A Discriminative Framework for Anomaly Detection in Large Videos. In: Proceedings of ECCV. (October 2016) 334–349
12. Ionescu, R.T., Smeureanu, S., Alexe, B., Popescu, M.: Unmasking the abnormal events in video. In: Proceedings of ICCV. (2017) 2895–2903
13. Auslander, B., Gupta, K., Aha, D.W.: Comparative evaluation of anomaly detection algorithms for local maritime video surveillance. In: Proceedings of SPIE, SPIE (2011)
14. Chawla, S., Gionis, A.: k-means-: A unified approach to clustering and outlier detection. In: Proceedings of SDM, SIAM (2013) 189–197
15. Jiang, M.F., Tseng, S.S., Su, C.M.: Two-phase clustering process for outliers detection. *Pattern Recognition Letters* **22**(6) (2001) 691–700
16. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(3) (March 2008) 555–560
17. Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: Proceedings of CVPR. (2011) 3449–3456

18. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: Proceedings of CVPR. (2016) 733–742
19. Hinami, R., Mei, T., Satoh, S.: Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge. In: Proceedings of ICCV. (2017) 3639–3647
20. Luo, W., Liu, W., Gao, S.: A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. In: Proceedings of ICCV. (2017) 341–349
21. Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., Sebe, N.: Abnormal Event Detection in Videos using Generative Adversarial Nets. In: Proceedings of ICIP. (2017)
22. Saligrama, V., Chen, Z.: Video anomaly detection based on local statistical aggregates. In: Proceedings of CVPR. (2012) 2112–2119
23. Smeureanu, S., Ionescu, R.T., Popescu, M., Alexe, B.: Deep Appearance Features for Abnormal Behavior Detection in Video. In: Proceedings of ICIAP. Volume 10485. (2017) 779–789
24. Sun, Q., Liu, H., Harada, T.: Online growing neural gas for anomaly detection in changing surveillance scenes. *Pattern Recognition* **64**(C) (April 2017) 187–201
25. Zhang, Y., Lu, H., Zhang, L., Ruan, X., Sakai, S.: Video anomaly detection based on locality sensitive hashing filters. *Pattern Recognition* **59** (2016) 302–311
26. Ren, H., Liu, W., Olsen, S.I., Escalera, S., Moeslund, T.B.: Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection. In: Proceedings of BMVC. (2015) 28.1–28.13
27. Dutta, J.K., Banerjee, B.: Online Detection of Abnormal Events Using Incremental Coding Length. In: Proceedings of AACL. (2015) 3755–3761
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., A., K., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* **115**(3) (2015) 211–252
29. Abuolaim, A.A., Leow, W.K., Varadarajan, J., Ahuja, N.: On the Essence of Unsupervised Detection of Anomalous Motion in Surveillance Videos. In: Proceedings of CAIP. (2017) 160–171
30. Lazebnik, S., Schmid, C., Ponce, J.: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: Proceedings of CVPR. Volume 2., Washington, DC, USA, IEEE Computer Society (2006) 2169–2178
31. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Proceedings of CVPR, Washington, DC, USA, IEEE Computer Society (2005) 886–893
32. Schölkopf, B., Platt, J.C., Shawe-Taylor, J.C., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. *Neural Computation* **13**(7) (July 2001) 1443–1471
33. Vedaldi, A., Fulkerson, B.: VLFeat: An Open and Portable Library of Computer Vision Algorithms. <http://www.vlfeat.org/> (2008)
34. Du, Q., Faber, V., Gunzburger, M.: Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review* **41**(4) (December 1999) 637–676
35. Arthur, D., Vassilvitskii, S.: k-means++: The Advantages of Careful Seeding. In: Proceedings of SODA, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2007) 1027–1035
36. Chang, C.C., Lin, C.J.: LibSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* **2** (2011) 27:1–27:27 Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.



37. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the Devil in the Details: Delving Deep into Convolutional Nets. In: Proceedings of BMVC. (2014)
38. Vedaldi, A., Lenc, K.: MatConvNet – Convolutional Neural Networks for MATLAB. In: Proceeding of ACMMM. (2015)