

Detecting abnormal events in video using Narrowed Normality Clusters

Radu Tudor Ionescu^{1,2}, Sorina Smeureanu^{1,2}, Marius Popescu^{1,2}, Bogdan Alexe^{1,2}

¹University of Bucharest, 14 Academiei, Bucharest, Romania

²SecurifAI, 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. In the feature extraction stage, we propose to augment spatio-temporal cubes with deep appearance features extracted from the last convolutional layer of a pre-trained neural network. After extracting motion and appearance features from the training video containing only normal events, we apply *k*-means clustering to find clusters representing different types of normal motion and appearance features. In the first stage, we consider that clusters with fewer samples (with respect to a given threshold) contain mostly 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 normality. If the test sample does not fit well in any narrowed normality cluster, then it is labeled as abnormal. We compare our method with several state-of-the-art methods on three 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 24 frames per second on a single CPU.

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 context. For example, a truck driving by on the street is considered a perfectly normal event, but if the truck drives through a pedestrian area, then it is regarded as an abnormal event. Another example that illustrates the importance of context is represented by two people fighting in a boxing ring (normal event) versus fighting on the street (abnormal event). In addition to the reliance on context, we can generally

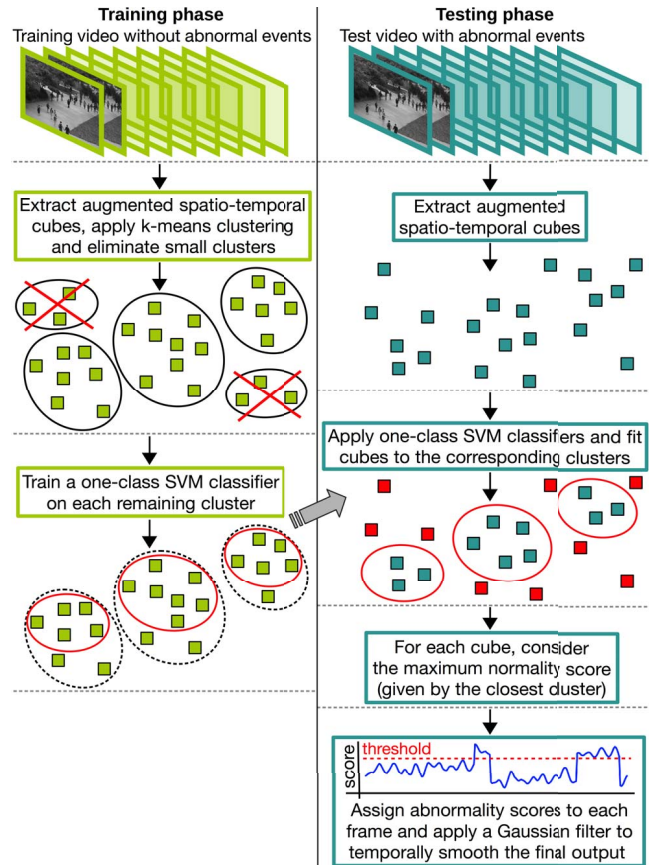


Figure 1. Our anomaly detection framework based on Narrowed Normality Clusters. In the training phase, we apply a two-stage outlier detection algorithm based on *k*-means and one-class SVM. In the testing phase, we label a test sample as abnormal if its maximum normality score among the scores provided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

agree that abnormal behavior should rather be represented by rare (less expected) events [20] that do not occur as often as usual (more familiar) events. Due to the scarcity and variability of abnormal events, it is generally hard to obtain a representative set of anomalies at training time. Hence, traditional supervised learning methods are usually ruled out. Therefore, most abnormal event detection approaches [3, 9, 22, 25, 27, 29, 30, 41, 43] are based on learning a model of normality from training videos con-

taining only familiar events. At test time, events are labeled as abnormal if they deviate from the normality model. We approach abnormal event detection in a similar manner, and propose to build a model of familiarity using the two-stage outlier detection algorithm illustrated in Figure 1. We first extract spatio-temporal cubes [12, 19, 27], which we augment with additional information about location, motion direction and object appearance. 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 normal motion and appearance. 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 [5, 8, 21], 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 normality. We therefore coin the term *Narrowed Normality Clusters (NNC)* for our two-stage outlier detection algorithm. To detect 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 and appearance. If the test sample does not fit well in any normality cluster, its corresponding maximum normality score will be negative. Consequently, the respective test sample is labeled as abnormal.

In summary, the novelty of our paper consists of (i) augmenting spatio-temporal cubes [27] with deep appearance features, (ii) assembling together two popular methods for outlier detection (k-means and one-class SVM) into a simple and fast framework and (iii) narrowing down normality clusters by learning a tight border around each cluster.

We conduct experiments on the Avenue [27], the Subway [2] and the UMN [30] data sets in order to compare our NNC approach with several state-of-the-art abnormal event detection methods [9, 10, 12, 15, 17, 19, 26, 27, 28, 30, 31, 34, 37, 38, 42]. The empirical results indicate that, on two of the test sets (Avenue and Subway), we obtain better results than all these approaches. We also report the second-best score on the third data set (UMN). It is important to mention that our approach yields impressive results, while running in real-time at 24 frames per second on a CPU.

We organize the paper as described next. We present re-

lated 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. We draw our final conclusions in Section 5.

2. Related Work

Abnormal event detection is commonly formalized as an outlier detection task [3, 9, 10, 14, 22, 25, 27, 29, 30, 32, 38, 41, 42, 43], in which the general approach is to learn a model of normality from training data and label the detected outliers as abnormal events. Some abnormal event detection approaches [9, 10, 14, 27, 32] are based on learning a dictionary of atoms representing normal events, and on labeling the events not represented in 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 [42] or deep features [15, 17, 26, 28, 31, 37, 41] to achieve better results. Hasan et al. [15] proposed 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. [17] proposed to train convolutional neural networks (CNN) on multiple visual tasks to exploit semantic information that is useful for detecting and recounting abnormal events, while Smeureanu et al. [37] simply applied convolutional neural networks pre-trained on the ILSVRC benchmark [33]. Luo et al. [28] proposed a Temporally-coherent Sparse Coding approach, which can be mapped to a stacked Recurrent Neural Network which facilitates parameter optimization and accelerates anomaly prediction. The approach presented in [31] 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. Liu et al. [26] proposed a method for abnormal event detection based on a deep future frame prediction framework. The approach uses the difference between a predicted future frame and the ground-truth frame to detect abnormal events. For a better detection rate, the authors add a temporal constraint based on optical flow along with the spatial constraints.

There have been some approaches that employ unsupervised steps for abnormal event detection [14, 32, 38, 41]. In [14], the authors presented a method that constructs a model of familiar events from training data. The model is incrementally updated in an unsupervised manner as new patterns are observed in the test data. In a similar manner, Sun et al. [38] proposed to train a Growing Neural Gas model starting with the training videos and continuing with the test videos, as the test videos are analyzed for anomaly detection. Ren et al. [32] used an unsupervised approach, spectral clustering, to construct a dictionary of atoms, each representing a single type of normal behavior. However, the approach of Ren et al. [32] requires training videos of normal events to build the dictionary. In order to learn deep feature representations in an unsupervised manner, Xu et al. [41] employed Stacked Denoising Auto-Encoders. In the end, they used one-class SVM classifiers to detect the abnormal events. There are some works that do not require any kind of training data in order to detect abnormal events [12, 19]. The approach proposed by Del Giorno et al. [12] detects changes in a short sequence of frames from the video by deciding which frames are distinguishable from all the previous frames. Since Del Giorno et al. [12] aimed to obtain an approach independent of temporal ordering, they created shuffles of the test data by permuting the frames before running each instance of the change detection. Ionescu et al. [19] applied unmasking, a technique based on training a binary classifier to distinguish between two consecutive short video sequences, while gradually removing the most discriminant features. Their hypothesis is that the higher training accuracy rates of the immediately obtained classifiers represent abnormal events.

Regarding the feature representation, we use spatio-temporal cubes to represent motion, as other recent approaches [12, 19, 27]. Unlike all these approaches, we propose to augment each cube with its location within a spatial pyramid applied over the video frames, with the mean direction given by the 3D motion gradients inside the cube, and with deep appearance features. Our experiments show that the augmentation is useful. Regarding the outlier detection approach, there are a few works [1, 5] that applied k-means clustering for abnormal event detection. Abuolaim et al. [1] used 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 normality. Moreover, their approach does not allow to set an abnormality threshold, and thus, it cannot be optimized for better precision or recall. More closely to our approach, Auslander et al. [5] defined three possible assumptions (see Section 4.1 in [5]) 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 [5], 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 a testing phase, as illustrated in Figure 1. We next provide an high-level summary 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. During inference, 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

Unlike other approaches [9, 41], we apply the same steps in order to extract motion and appearance features from video, irrespective of the data set.

Encoding motion. Given the input video, we resize all frames to 120×160 pixels and uniformly partition each frame to a set of non-overlapping 10×10 patches. Corresponding patches in 5 consecutive frames are stacked together to form a spatio-temporal cube. The dimension of each spatio-temporal cube is $10 \times 10 \times 5$. We next derive 3D gradient features from 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 [12, 19, 27]. Similar to [12, 19, 27], we eliminate cubes in a region, if the video is static in the respective region. Different from [12, 27], we do not em-

ploy Principal Component Analysis to reduce the feature vector dimension from 500 to 100 components, as it has no impact on the performance. Moreover, we diverge from the standard spatio-temporal cube representation by augmenting the cubes with additional information about location, motion direction and object appearance, as described next.

Encoding location. We divide each frame into a spatial pyramid [24] with two levels, the first level containing 2×2 bins and the second one containing 4×4 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 2×2 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 [11]. 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.

Encoding object appearance. In many computer vision tasks, for instance image difficulty prediction [18], higher level features, such as the ones learned with convolutional neural networks (CNN) [23] are the most effective. To build our appearance features, we consider a shallow pre-trained CNN architecture, namely VGG-f [7], which is able to process the video frames in real-time on a CPU. Considering that we want our detection framework to work in real-time on a standard desktop computer, not equipped with expensive GPU, the VGG-f architecture is an excellent choice. We hereby note that better anomaly detection performance can probably be achieved by employing deeper CNN architectures, such as VGG-verydeep [36] or ResNet [16]. We use a VGG-f model pre-trained on the ILSVRC bench-

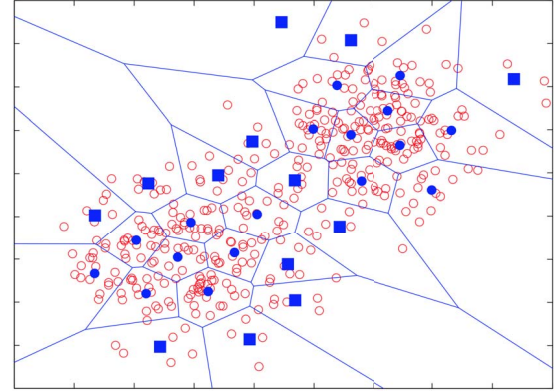


Figure 2. A set of 400 data points sampled from two normal distributions of different means. The points are clustered into 30 clusters using k-means. The centroids of clusters with less than 10 samples are represented with a large blue square. Best viewed in color.

mark [33] to extract deep features as follows. Given the input video, we resize the frames to 224×224 pixels. We then subtract the mean image from each frame and provide it as input to the VGG-f model. We remove the fully-connected layers (identified as *fc6*, *fc7* and *softmax*) and consider the activation maps of the last convolutional layer (*conv5*) as appearance features. While the fully-connected layers are adapted for object recognition, the last convolutional layer contains valuable appearance and pose information which is more useful for our anomaly detection task. Ideally, we would like to have at least slightly different representations for a person walking versus a person running, hence *conv5* is more suitable than *fc6* or *fc7*. From the *conv5* layer, we obtain 256 activation maps, each of 13×13 units. We then resize each activation map to match the number of spatio-temporal cubes in a frame, i.e. from 13×13 to 12×16 , using bicubic interpolation. We then concatenate each set of 256 filter activations to the corresponding spatio-temporal cube.

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 normality. 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 proposed in [5]. 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

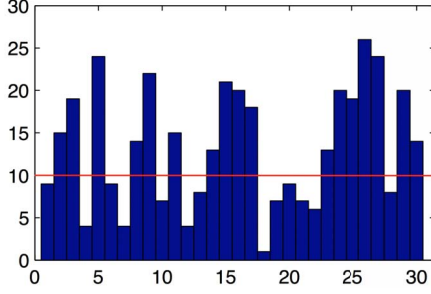


Figure 3. A histogram representing the number of data points in each cluster. The histogram corresponds to the k-means clustering applied over the 400 data points illustrated in Figure 2. A threshold of 10 is used to detect clusters of outliers. Best viewed in color.

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, in the context of 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 occur 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 [35] 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 regard, we can state 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 and appearance. To train our set of classifiers, we consider each spatio-temporal cube as an independent and individual sample, disregarding the temporal relations among 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 . The one-class SVM model corresponding to a cluster c_j will learn to separate a small region capturing the normal cubes inside cluster c_j from the rest of the feature space, by maximizing the distance from the separating hyperplane to the origin. This associates to each cluster a binary classification function g that singles out a region in the input space where the probability density

of a particular type of normality 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 must be classified either as normal or abnormal, $x_i \in \mathcal{X}$ is a training cube, α_i are the weights assigned to the support vectors x_i , ρ is the distance from the hyperplane to the origin, and k is a kernel function, in our case, the linear kernel. If we just need 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

We consider three data sets for the abnormal event detection experiments.

Avenue. The Avenue data set [27] is composed of 16 training videos and 21 test videos. In total, the Avenue data set contains 15328 frames for training and 15324 frames for testing. The resolution of each frame is 360×640 pixels. The locations of anomalies are annotated in ground-truth pixel-level masks for each frame in the test videos. Hinami et al. [17] argued 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 evaluated their approach on a subset (Avenue17) that excludes these five videos. When we compare our results with those reported in [17], we also remove the same five videos for a fair comparison.

Subway. The Subway surveillance data set [2] is one of the largest data sets for anomaly detection in video. The Subway data set is formed of two videos, one of 96 minutes (Entrance gate) and another one of 43 minutes (Exit gate). The Entrance gate video contains 144251 frames, while the Exit gate video contains 64903 frames. The resolution of each video frame is 384×512 pixels. Abnormal events are labeled at the frame level.

UMN. The UMN Unusual Crowd Activity data set [30] is composed of three different crowded scenes of various

lengths. The first scene contains 1453 frames, the second scene contains 4144 frames, and the third scene contains 2144 frames. The resolution of each frame is 240×320 pixels. In the normal scenario, 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 opt for ROC curves and the corresponding *area under the curve* (AUC) computed with respect to ground-truth frame-level annotations, and, when available (for the Avenue data set), pixel-level annotations. We use the same frame-level and pixel-level AUC definitions as in previous works [10, 12, 19, 27, 29, 38, 41]. 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. Before the evaluation, we smooth the pixel-level detection maps with the same filter used by [12, 19, 27] in order to obtain the final pixel-level detections. We exclude the recent approach (based on k-means) of Abuolaim et al. [1] 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 [10, 14, 27, 29, 41, 42] include the Equal Error Rate (EER) as evaluation metric, but some recent works [12, 19] 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 [12, 19], we refrain from employing the EER in our evaluation.

4.3. Parameter and Implementation Details

We extract spatio-temporal cubes from the training and the 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. For the appearance features, we consider the pre-trained VGG-f [7] model provided in MatConvNet [40], and extract the features from the *conv5* layer. For a faster processing time, we extract features for one in every two frames in the test video, without observing any drop in performance. To cluster the augmented cubes, we employ the k-means implementation from VLFeat [39] based on the original Lloyd algorithm [13]. We use k-means++ [4] 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 always proportional to the number of training cubes. For instance, there are almost 280 thousand cubes extracted from the Avenue training videos, hence we set $k = 280$ for the Avenue data set. We then eliminate the clusters with less than 500 cubes, irrespective of the data

Method	AUC	
	Frame	Pixel
Lu et al. [27]	80.9	92.9
Hasan et al. [15]	70.2	-
Del Giorno et al. [12]	78.3	91.0
Smeureanu et al. [37]	84.6	93.5
Ionescu et al. [19]	80.6	93.0
Luo et al. [28]	81.7	-
Liu et al. [26]	85.1	-
cubes + one-class SVM	81.3	93.0
aug. cubes + one-class SVM	84.2	93.4
aug. cubes + k-means + one-class SVM	86.4	93.7
aug. cubes + k-means + 1-NN	78.8	91.5
NNC (ours)	88.9	94.1

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 (ablated) versions are compared with several state-of-the-art approaches [12, 15, 19, 26, 27, 28, 37], which are listed in temporal order.

set. To remove the outliers from each cluster, we employ the one-class SVM implementation from LibSVM [6]. 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 cubes as normal (the other 1% are outliers). At test time, we are able to process the test videos at nearly 24 FPS 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 [12, 15, 19, 26, 27, 28, 37] 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 includes ablation results for stripped-down 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 2.9% (up to 84.2%). Another 2.2% (up to 86.4%) 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 2.5% and reach a frame-level AUC of 88.9%. 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 78.8%, which is nearly 10% lower than the result of NNC. This ablation result shows the importance of using one-class SVM after k-means in order to learn a tight border around each cluster.

Using Narrowed Normality Clusters of augmented spatio-temporal cubes, we are able to surpass the results re-

Method	Frame AUC	Pixel AUC
Hinami et al. [17]	89.8	-
NNC (ours)	91.1	94.3

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 [17].

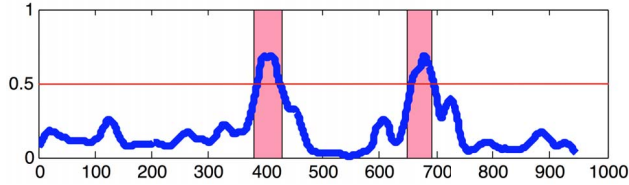


Figure 4. Frame-level anomaly detection scores (between 0 and 1) provided by our approach for test video 4 in the Avenue data set. The video has 947 frames. Ground-truth abnormal events are represented in pink and our scores are illustrated in blue. Best viewed in color.



Figure 5. True positive (top row) versus false positive (bottom row) detections of our framework. Examples are selected from the Avenue data set. Best viewed in color.

ported in previous works in terms of frame-level and pixel-level AUC. Compared to the most recent works [19, 26, 28, 37], our framework brings an improvement of more than 3.8% in terms of frame-level AUC. Since our framework is able to process the video online on a single CPU, we consider that our results on the Avenue data set are noteworthy.

We also compare our approach with [17] 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. [17]. It is worth nothing that our approach yields better performance on the Avenue17 subset than on the full Avenue data set, indicating that the five removed test videos are actually more difficult than those left in Avenue17. As Hinami et al. [17] observed, the removed videos contain abnormal objects that are not properly annotated, hence methods are prone to reach higher 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 with the ground-truth labels. There are two abnormal events in this video and we can easily identify

Method	Frame AUC		
	Entrance gate	Exit gate	Average
Cong et al. [10]	80.0	83.0	81.5
Saligrama et al. [34]	89.1	-	-
Cheng et al. [9]	92.7	-	-
Hasan et al. [15]	94.3	80.7	87.5
Del Giorno et al. [12]	69.1	82.4	75.8
Ionescu et al. [19]	71.3	86.3	78.8
NNC (ours)	93.5	95.1	94.3

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 [9, 10, 12, 15, 19, 34], which are listed in temporal order.

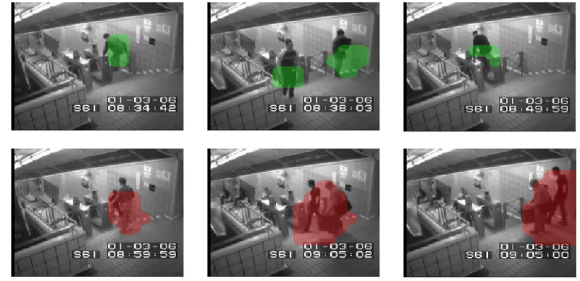


Figure 6. True positive (top row) versus false positive (bottom row) detections of our framework. Examples are selected from the Subway Entrance gate. Best viewed in color.

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 people walking synchronously*. The last two false positive examples 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 [9, 10, 12, 14, 15, 19, 27, 34, 42] that report results on the Subway data set, some of these works [14, 27, 42] did not use the frame-level AUC as evaluation metric. Therefore, we only compare our approach with those methods [9, 10, 12, 15, 19, 34] that reported the frame-level AUC. The results of our comparative study are reported in Table 3. On the Entrance gate video, we obtain the second-best score, after Hasan et al. [15]. They report a frame-level AUC of 94.3%, which is 0.8% higher than our score (93.5%). Things look differently on the Exit gate video, as we obtain the best score (95.1%) among all methods, surpassing the approach of Hasan et al. [15] by 14.4% and the second-best method [19] by 8.8%. On average (last column in Table 3), we obtain the best results on the Subway data set.

In Figure 6, we present some interesting qualitative re-

Method	Frame AUC			
	Scene			All scenes
	1	2	3	
Mehran et al. [30]	-	-	-	96.0
Cong et al. [10]	99.5	97.5	96.4	97.8
Saligrama et al. [34]	-	-	-	98.5
Del Giorno et al. [12]	-	-	-	91.0
Zhang et al. [42]	99.2	98.3	98.7	98.7
Sun et al. [38]	99.8	99.3	99.9	99.7
Smeureanu et al. [37]	98.8	93.6	98.9	97.1
Ionescu et al. [19]	99.3	87.7	98.2	95.1
Ravanbakhsh et al. [31]	-	-	-	99.0
NNC (ours)	99.9	98.2	99.8	99.3

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 [10, 12, 19, 30, 31, 34, 37, 38, 42], which are listed in temporal order.

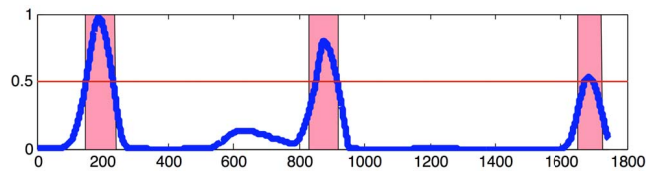


Figure 7. Frame-level anomaly detection scores (between 0 and 1) provided by our framework for the third scene in the UMN data set. The video has 1744 test frames. Ground-truth abnormal events are represented in pink and our scores are illustrated in blue. Best viewed in color.

sults obtained by our framework on the Entrance gate video. The true positive abnormal events are *a person jumping over the fence*, *two people walking in the wrong direction* and *a person jumping over the gate*, while false positive detections are *a person running* and *two people walking synchronously*.

4.6. Results on the UMN Data Set

On the UMN data set, we compare our approach with several methods [10, 12, 19, 30, 31, 34, 37, 38, 42]. 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 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 (98.2%) is on the second scene. Over all scenes, we reach the second highest frame-level AUC (99.3%), which is 0.4% lower than the best score (99.7%) obtained by Sun et al. [38].

In Figure 8, we present some interesting qualitative results obtained by our framework on the second scene, as it



Figure 8. True positive (top row) versus false positive (bottom row) detections of our framework. Examples are selected from the UMN data set. Best viewed in color.

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 Normality 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. Our secondary contribution was to augment the spatio-temporal cubes with location, motion direction, and deep appearance features. We conducted abnormal event detection experiments on three data sets to compare our approach with a series of state-of-the-art approaches [9, 10, 12, 15, 17, 19, 26, 27, 28, 30, 31, 34, 37, 38, 42]. 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 second best on the UMN data set. At the same time, we can process the test video in real-time at 24 frames per second on a single CPU. In future work, we aim to develop an approach to train deep features on a closely related task, such as action recognition, and transfer the learned features to our task.

Acknowledgments. The work of Radu Tudor Ionescu was supported by a grant of Ministry of Research and Innovation, CNCS - UEFISCDI, project number PN-III-P1-1.1-PD-2016-0787, within PNCIDI III.

References

- [1] A. A. Abuolaim, W. K. Leow, J. Varadarajan, and N. Ahuja. On the Essence of Unsupervised Detection of Anomalous Motion in Surveillance Videos. In *Proceedings of CAIP*, pages 160–171, 2017. 3, 6
- [2] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008. 2, 5
- [3] B. Antic and B. Ommer. Video parsing for abnormality detection. In *Proceedings of ICCV*, pages 2415–2422, 2011. 1, 2
- [4] D. Arthur and S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proceedings of SODA*, pages 1027–1035, 2007. 6
- [5] B. Auslander, K. Gupta, and D. W. Aha. Comparative evaluation of anomaly detection algorithms for local maritime video surveillance. In *Proceedings of SPIE*, 2011. 2, 3, 4
- [6] C.-C. Chang and C.-J. Lin. LibSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 6
- [7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *Proceedings of BMVC*, 2014. 4, 6
- [8] S. Chawla and A. Gionis. k-means--: A unified approach to clustering and outlier detection. In *Proceedings of SDM*, pages 189–197, 2013. 2
- [9] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang. Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression. In *Proceedings of CVPR*, pages 2909–2917, 2015. 1, 2, 3, 7, 8
- [10] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. In *Proceedings of CVPR*, pages 3449–3456, 2011. 2, 6, 7, 8
- [11] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *Proceedings of CVPR*, pages 886–893, 2005. 4
- [12] A. Del Giorno, J. Bagnell, and M. Hebert. A Discriminative Framework for Anomaly Detection in Large Videos. In *Proceedings of ECCV*, pages 334–349, 2016. 2, 3, 6, 7, 8
- [13] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review*, 41(4):637–676, 1999. 6
- [14] J. K. Dutta and B. Banerjee. Online Detection of Abnormal Events Using Incremental Coding Length. In *Proceedings of AAAI*, pages 3755–3761, 2015. 2, 3, 6, 7
- [15] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *Proceedings of CVPR*, pages 733–742, 2016. 2, 6, 7, 8
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of CVPR*, pages 770–778, 2016. 4
- [17] R. Hinami, T. Mei, and S. Satoh. Joint Detection and Re-counting of Abnormal Events by Learning Deep Generic Knowledge. In *Proceedings of ICCV*, pages 3639–3647, 2017. 2, 5, 7, 8
- [18] R. T. Ionescu, B. Alexe, M. Leordeanu, M. Popescu, D. Papadopoulos, and V. Ferrari. How hard can it be? Estimating the difficulty of visual search in an image. In *Proceedings of CVPR*, pages 2157–2166, 2016. 4
- [19] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu. Unmasking the abnormal events in video. In *Proceedings of ICCV*, pages 2895–2903, 2017. 2, 3, 6, 7, 8
- [20] L. Itti and P. Baldi. A Principled Approach to Detecting Surprising Events in Video. In *Proceedings of CVPR*, volume 1, pages 631–637, 2005. 1
- [21] M.-F. Jiang, S.-S. Tseng, and C.-M. Su. Two-phase clustering process for outliers detection. *Pattern Recognition Letters*, 22(6):691–700, 2001. 2
- [22] J. Kim and K. Grauman. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In *Proceedings of CVPR*, pages 2921–2928, 2009. 1, 2
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of NIPS*, pages 1106–1114, 2012. 4
- [24] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of CVPR*, volume 2, pages 2169–2178, 2006. 4
- [25] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014. 1, 2
- [26] W. Liu, W. Luo, D. Lian, and S. Gao. Future Frame Prediction for Anomaly Detection – A New Baseline. In *Proceedings of CVPR*, pages 6536–6545, 2018. 2, 3, 6, 7, 8
- [27] C. Lu, J. Shi, and J. Jia. Abnormal Event Detection at 150 FPS in MATLAB. In *Proceedings of ICCV*, pages 2720–2727, 2013. 1, 2, 3, 5, 6, 7, 8
- [28] W. Luo, W. Liu, and S. Gao. A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework. In *Proceedings of ICCV*, pages 341–349, 2017. 2, 6, 7, 8
- [29] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos. Anomaly Detection in Crowded Scenes. In *Proceedings of CVPR*, pages 1975–1981, 2010. 1, 2, 6
- [30] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Proceedings of CVPR*, pages 935–942, 2009. 1, 2, 5, 8
- [31] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe. Abnormal Event Detection in Videos using Generative Adversarial Nets. In *Proceedings of ICIP*, pages 1577–1581, 2017. 2, 8
- [32] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moeslund. Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection. In *Proceedings of BMVC*, pages 28.1–28.13, 2015. 2, 3
- [33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, K. A., A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2, 4

- [34] V. Saligrama and Z. Chen. Video anomaly detection based on local statistical aggregates. In *Proceedings of CVPR*, pages 2112–2119, 2012. 2, 7, 8
- [35] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001. 5
- [36] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of ICLR*, 2014. 4
- [37] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe. Deep Appearance Features for Abnormal Behavior Detection in Video. In *Proceedings of ICIAP*, volume 10485, pages 779–789, 2017. 2, 6, 7, 8
- [38] Q. Sun, H. Liu, and T. Harada. Online growing neural gas for anomaly detection in changing surveillance scenes. *Pattern Recognition*, 64(C):187–201, 2017. 2, 3, 6, 8
- [39] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms. <http://www.vlfeat.org/>, 2008. 6
- [40] A. Vedaldi and K. Lenc. MatConvNet – Convolutional Neural Networks for MATLAB. In *Proceeding of ACMMM*, 2015. 6
- [41] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning Deep Representations of Appearance and Motion for Anomalous Event Detection. In *Proceedings of BMVC*, pages 8.1–8.12, 2015. 1, 2, 3, 6
- [42] Y. Zhang, H. Lu, L. Zhang, X. Ruan, and S. Sakai. Video anomaly detection based on locality sensitive hashing filters. *Pattern Recognition*, 59:302–311, 2016. 2, 6, 7, 8
- [43] B. Zhao, L. Fei-Fei, and E. P. Xing. Online Detection of Unusual Events in Videos via Dynamic Sparse Coding. In *Proceedings of CVPR*, pages 3313–3320, 2011. 1, 2