Contents

1 METHOD

1.1 Feature Extraction

In many computer vision tasks, higher level features, such as the ones learned with convolutional neural networks (CNN) are the most effective. To build our appearance features, we consider a pre-trained CNN architecture able to process the frames as fast as possible, namely VGG-f. 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 is an excellent choice as it can process about 20 frames per second on CPU. We hereby note that better anomaly detection performance can be achieved by employing deeper CNN architectures, such as:

(1) VGG-verydeep;
(2) GoogleNet;
(3) ResNet.

1.2 Learning Model

We use the one-class SVM approach of Schölkopf et al. To detect abnormal events in video. The training data in our case is composed of a few videos representing only normal events. We consider each video frame as an individual and independent sample, disregarding the temporal relations between video frames. Let $X = x1, x1, ..., xn|xi \in Rm$ denote the set of training frames. In this formulation, our one-class SVM model will learn to separate a small region capturing most of the normal frames 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 regions in the input space where the probability density of normal events lives:

$$g(x) = sign(\sum_{i=1}^{n} a_i k(x, x_i) - p), \tag{1}$$

where x is a test frame that needs to be classified either as normal or abnormal, $xi \in X$ is a training frame, k is a kernel function, αi are the weights assigned to the support vectors xi,and ρ is the distance from the hyperplane to the origin. If we desire a score reflecting the abnormality level of a frame, we can simply remove the (sign) transfer function from Eq. (1). The coefficients αi are found as the solution of the dual problem:

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \; subjectto \; 0 \leq \alpha_i \leq \frac{1}{vn}, \sum_{i=1}^{n} \alpha_i = 1 \tag{2}$$

where $v \in [0, 1]$ is a regularization parameter that controls the percentage of outliers to be excluded by the learned model. As noted by Scholkopf et al., the offset ρ can be recovered by exploiting that for any αi that is not at the lower or upper bound, the corresponding sample xi satisfies:

$$p = \sum_{j=1}^{n} \alpha_j k(x_j, x_i). \tag{3}$$

Since we already represent the frames in a high dimensional space (m = 43264) by extracting CNN features, we no longer have to embed the samples into a higher dimensional space. Hence, we decide to use the linear kernel function in our one-class SVM model, which corresponds to the feature map $\phi(x) = x$:

$$k(x, z) = \langle x, z \rangle \tag{4}$$

2 EXPERIMENTS

2.1 Data Sets

We show abnormal event detection results on two benchmark data sets:

- **Avenue.** We first consider the Avenue data set, 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 640×360 pixels. Locations of anomalies are annotated in ground truth pixel-level masks for each frame in the testing videos;
- **UMN.** The UMN Unusual Crowd Activity data set consists of three different crowded scenes, each with 1453, 4144, and 2144 frames, respectively. The resolution of each frame is 320 × 240 pixels. In the normal settings people walk around in the scene, and the abnormal behavior is defined as people running in all directions. We use the first 400 frames in each scene for training.

2.2 Results on the Avenue Data Set

We first compare our abnormal behavior detection framework based on deep features with two state-of-the-art approaches. The frame-level and pixel-level AUC metrics computed on the Avenue data set are presented in Table 1. Compared to the method of Del Giorno et al., our framework yields an improvement of 6.3%, in terms of frame-level AUC, and an improvement of 2.5%, in terms of pixel-level AUC. We also obtain better results than Lu et al., as our framework gains 3.7% in terms of frame-level AUC and 0.6% in terms of pixel-level AUC. Overall, our method is able to surpass the performance of both state-of-the-art methods. Figure 1. illustrates the frame-level anomaly scores, for test video 4in the Avenue data set, produced by our framework based on VGG-f features and one-class SVM. According to the ground-truth anomaly labels, there are two.

Table 1. Abnormal event detection results in terms of frame-level and pixel-level AUC on the Avenue data set. Our framework is compared with two state-of-the-art approaches

Method	Frame AUC	Pixel AUC
Lu et al.	80.9%	92.9%
Del Giorno et al.	78.3%	91.0%
VGG-f conv5+one-class SVM	84.6%	93.5%

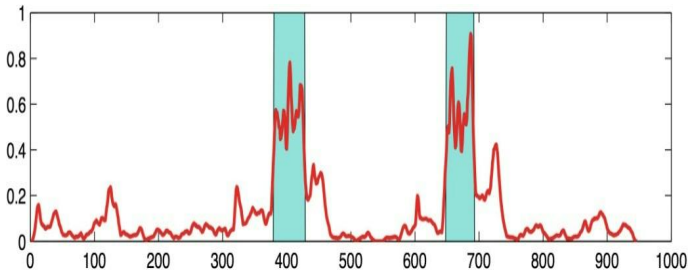

Fig. 1. Frame-level anomaly detection scores (between 0 and 1) provided by our framework for test video 4in the Avenue data set. The video has 947 frames. Ground-truth abnormal events are represented in cyan, and our scores are illustrated in red. B

Fig. 2. True positive (top row) versus false positive (bottom row) detections of our framework based on VGG-f features and one-class SVM. Examples are selected from the Avenue data set.

CONCLUSION

[h!] In this work, we have proposed a novel framework for abnormal event detection in video that is based on extracting deep features from pre-trained CNN models, and on using one-class SVM to learn a model of normality. We have conducted abnormal event detection experiments on two data sets in order to compare our approach with several state-of-the-art approaches. The empirical results indicate that our approach gives better performance than some of these approaches, while processing the video online at 20 FPS. Although our model can reach very good results, it completely disregards motion information and the temporal structure in video. In future work, we aim to improve our performance by including motion features into our framework. One possible approach would be to employ convolutional two-stream networks to extract both motion and appearance features. We also aim to evaluate our framework on other data sets.