Computer Vision and Image Understanding 210 (2021) 103249

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Decoupled appearance and motion learning for efficient anomaly detection in surveillance video

Bo Li *, Sam Leroux, Pieter Simoens

IDLab, Department of Information Technology, Ghent University - imec, Ghent 9000, Belgium

ARTICLE INFO

Communicated by Nikos Paragios

MSC:
41A05
41A10
65D05
65D17

Keywords:
Anomaly detection
Surveillance video
Unsupervised learning

ABSTRACT

Automating the analysis of surveillance video footage is of great interest when urban environments or industrial sites are monitored by a large number of cameras. As anomalies are often context-specific, it is hard to predefine events of interest and collect labeled training data. A purely unsupervised approach for automated anomaly detection is much more suitable. For every camera, a separate algorithm could then be deployed that learns over time a baseline model of appearance and motion related features of the objects within the camera viewport. Anything that deviates from this baseline is flagged as an anomaly for further analysis downstream. We propose a new neural network architecture that learns the normal behavior in a purely unsupervised fashion. In contrast to previous work, we use latent code predictions as our anomaly metric. We show that this outperforms frame reconstruction-based and prediction-based methods on different benchmark datasets both in terms of accuracy and robustness against changing lighting and weather conditions. By decoupling an appearance and a motion model, our model can also process 16 to 45 times more frames per second than related approaches which makes our model suitable for deploying on the camera itself or on other edge devices.

1. Introduction

Rising concerns for public security and safety have increased the number of surveillance cameras installed in our streets and public places (Abati et al., 2019; Liu et al., 2018; Ionescu et al., 2019; Chandola et al., 2009; Morais et al., 2019). Human operators in a control room continuously inspect these video streams on a multi-screen video wall, looking for abnormal events that may mandate further inspection. As human operators can only process a few video streams at the same time, part of the surveillance workflow must be automated when the number of cameras grows. By automating the detection of anomalous events, human operators can focus on the appropriate response to these events, e.g. by requesting a police intervention.

Since anomalies are context-specific (Song et al., 2007), each video stream requires a tailored anomaly detection algorithm. A running person for example is considered an anomaly in a busy shopping street but it might be normal in a train station as people are often in a hurry to catch the train. In this work we introduce a new neural network architecture that is able to recognize anomalous events in a surveillance camera stream using only unsupervised training. We propose to decouple the learning of appearance and motion information which are the key factors for determining anomalies in a surveillance video. We first train an autoencoder by reconstructing individual frames to capture high level appearance features such as the location, shape and size of an object. Such features however cannot guarantee the detection of anomalies that are caused by motion related features such as speed or trajectory. We therefore add a second component that further exploits the spatiotemporal information of the frequently seen events by predicting the latent code for a future frame using the stacked latent codes of the previous k frames as the input. The underlying assumption is that the anomalous events are rare occasions and will not be modeled accurately by the networks. The predicted latent code of anomalous frames will hence deviate significantly from the observed latent codes.

Our approach is easy to implement and achieves state-of-the-art performance on benchmark datasets. We however do not only focus on detection accuracy but also address several other obstacles for real-world deployment. Our model is much more efficient than related approaches, which could make it possible to evaluate our model at the network edge, on or nearby the surveillance camera itself, as opposed to streaming all data to a central point for analysis. Inference at the edge is also a more privacy friendly paradigm since a human operator will not see the camera data unless his intervention is needed. Lastly, our experimental results indicate that detection performance based on prediction of latent codes is more robust against changing weather and lighting conditions.

The remainder of this paper is organized as follows: In Section 2 we give an overview of related anomaly detection methods. In Section 3 we introduce our approach and we experimentally validate it on different

* Corresponding author.
E-mail address: Bo.Li@UGent.be (B. Li).

https://doi.org/10.1016/j.cviu.2021.103249
Received 30 March 2020; Received in revised form 10 May 2021; Accepted 15 July 2021
Available online 18 July 2021

benchmark datasets in Section 4. In Section 5 we show that our approach is more robust against different distortions. We conduct an ablation study in Section 6 to analyze the role of different components of the model. We conclude in Section 7 and give a few pointers for future research directions.

2. Related work

Deep learning is currently the state-of-the-art method for many computer vision related tasks (Schmidhuber, 2015) and is also the technique behind the state-of-the-art anomaly detection methods for video surveillance type data. We can differentiate three different approaches to do anomaly detection with deep learning: reconstruction based methods, prediction based methods and methods that use characteristics of the latent codes to detect anomalies.

2.1. Reconstruction based methods

The most common approach is to build models that reconstruct their input. These models are based on an autoencoder architecture that contains a bottleneck for encoding high level features, creating a compressed representation of the input data. These compressed representations are then used to reconstruct the input data. The assumption here is that the reconstruction works fine for inputs that are similar to the data that was seen during training but that anomalous inputs cannot be modeled accurately by the learned features, resulting in a poor reconstruction. The reconstruction error is then used as a metric to detect anomalies.

For our use case of anomaly detection in video surveillance, it is not enough to only model spatial information by processing individual frames, we also need to consider the temporal information to detect anomalies that are caused by motion, such as high speed or irregular movement. Different approaches have been explored to incorporate this information into the model. Hasan et al. (2016) exploit the spatiotemporal information by reconstructing multiple stacked frames using an autoencoder. Xu et al. (2017) apply a denoising autoencoder (Vincent et al., 2008) to reconstruct frames. They use optical flow maps to describe the motion information. Similar to Xu et al. (2017), Nguyen and Meunier (2019) also encode motion information with optical flow map. Another option is to use Recurrent Neural Networks (RNN) or Long Short-Term Memory networks (LSTM) (Schmidhuber, 2015) to model the time dimension (Chong and Tay, 2017; Luo et al., 2017a; Zhou et al., 2019). Finally, there is also work that explores the use of a 3D convolutional networks (C3D) to model spatiotemporal representations. Tran et al. (2015) show that C3D can encapsulate information related to shapes and motions in video sequence better than a 2D based model, thus boosting the anomaly detection accuracy.

2.2. Prediction based methods

Instead of reconstructing the input, it is also possible to predict future frames. This requires a better understanding of temporal information. Liu et al. (2018) use a generative adversarial network (GAN) (Goodfellow et al., 2014) that takes stacked frames and optical flow features as input. Anomalies are detected at test time by measuring the difference between the predicted and observed future frame. Any deviation from the expected frame is considered as an anomaly. For more specialized applications we can also use domain specific features. Morais et al. (2019) for example deal with human-related anomaly detection by reconstructing and predicting the decomposed global body movement and local body posture from the human skeleton movement. Differently from the above works, our method (i) works on the single frame instead of stacked frames, so we only need to process each frame once (ii) predicts the future in the feature space, thus alleviate the blurry prediction problem of using pixel-wise Mean Square Error (MSE) as objective function (Mathieu et al., 2015) (iii) does not require any preprocessing steps such as the computation of optical flow maps to apply motion constraints in detecting anomalies.

2.3. Latent code based methods

Both lines of previous work generate expected frames and detect anomalies by measuring the difference in pixel space with the actual input frame. However, it is well known that pixel-wise similarity measurements do not necessarily correspond with human understanding of images (Larsen et al., 2016; Mathieu et al., 2015) and are often very sensitive to minor changes in brightness or color. On the other hand, the high level features extracted by a neural network are shown to be less sensitive to these distortions (Zhang et al., 2018). There are some very recent approaches that extract high level features with an autoencoder and then use a classifier such as a one-class SVM on the extracted features (Bouindour et al., 2019; Ionescu et al., 2019) to detect anomalies. The assumption is that the classifier will distribute the anomalies outside of the learned manifold. The work that is most similar to our approach is the Latent Space Autoregression model from Abati et al. (2019). They use features from a deep 3D convolutional autoencoder combined with an autoregressive network to model the probability distribution underlying the latent representation. They combine the reconstruction error and the likelihood of the latent code to identify the anomalies. Our approach is similar in that we also extract features and work in a high level latent code. However, our approach uses a 2D autoencoder to extract features which highly reduces the number of parameters and computational cost. We use a less complicated feed forward model to do the prediction which does not make any assumptions on the distribution family of the latent code. It allows us to directly use the Mean Squared Error (MSE) between the predicted latent code and the latent code that is extracted from the encoder as the anomaly score. Finally, we explicitly predict the latent code of a future frame instead of relying on the reconstruction of the current frame to capture the motion information.

3. Architecture

In this paper we propose a decoupled architecture to learn the spatiotemporal information which is important for determining anomalies in surveillance videos. We first train an autoencoder to reconstruct individual input frames and aim to represent the appearance information such as shape, location and outlook of an object in the latent codes. Then to further stress the appearance information for frequently seen events and the dynamical aspects in a video, we stack these extracted latent codes from the encoder for a sequence of k frames and use that as the input for a second network to predict the latent code for a future frame. The model is assumed to be only able to predict the latent codes for frequently seen events with high accuracy. The difference between the predicted and the observed latent codes is then used as the anomaly metric. The following sections explain these in detail.

3.1. Learning appearance features

To learn high level features, we use a U-Net (Ronneberger et al., 2015) type autoencoder that is trained to reconstruct individual input frames. To force the model to focus on the foreground, we subtract a background frame from each input frame. This background frame is calculated as a frame with per-pixel mean RGB values over all training data. The encoder learns to extract latent codes from a single frame and the decoder learns to reconstruct the input based on the extracted features. The original U-Net architecture has shortcut connections between encoder and decoder. To avoid the trivial solution of copying feature maps from the encoder to the decoder and to improve the regularization power, we add a shortcut connection between the previous frame T_{k-1} and current frame T_k. In other words, the feature maps that are calculated using frame T_{k-1} are concatenated with the feature maps from frame T_k in the upsampling path for reconstructing frame T_k. The detailed architecture is shown in Fig. 1(a) and (b).

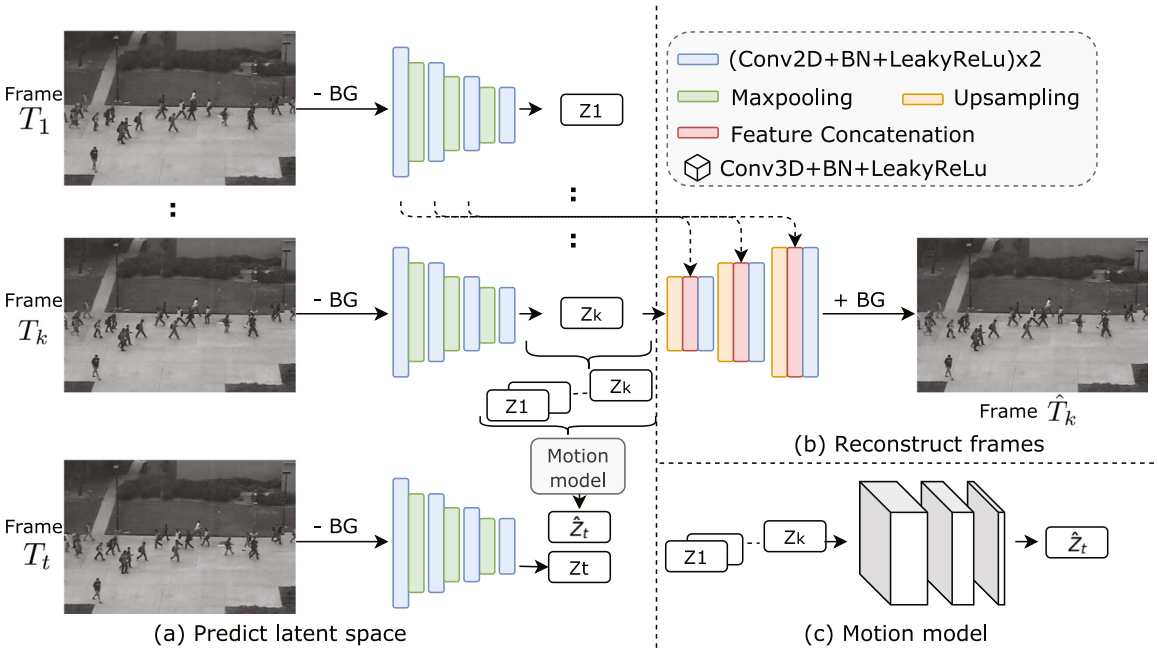

Fig. 1. Overview of our approach. We use the same encoder to extract a latent code for each input frame, where k is the number of frames in per input sequence and t is the frame that is 6 timesteps into the future ($k+6$). In the training phase, these latent codes are used to (a) predict latent codes for future frames with the motion model and (b) reconstruct current frames with the decoder. In the inference phase, we only use the encoder and the motion model. (c) *Conv3D* layers are used in the motion model to learn spatiotemporal information.

3.2. Learning motion features

Spatiotemporal information is important for detecting anomalies in videos. However, the features that are extracted by encoding a single frame as described above can only focus on the spatial information such as shape, location and size of an object and cannot guarantee the detection of the motion-related anomalies. We thus include a second component that can attend to both spatial and temporal dimension to further learn the dynamical aspects of video sequence. Previous work considered learning the temporal information either by predicting optical flows using a pretrained FlowNet (Hasan et al., 2016; Liu et al., 2018; Xu et al., 2015; Zhou et al., 2019; Nguyen and Meunier, 2019) or by predicting future frames in pixel space (Abati et al., 2019; Liu et al., 2018). This however has three drawbacks. First, the optical flow estimation is computationally expensive as it requires around 0.1 s to evaluate a single frame on a GPU machine (Ilg et al., 2017). Secondly, we need to consider the interaction between appearance and motion information. For example, a vehicle driving with very high speed is usually an anomaly except when it is an ambulance. Independently encoding the appearance and motion information using a pretrained optical flow model cannot take this into account. Finally, the pixel-wise Mean Square Error (MSE) objective function for predicting future often generates blurry frames (Mathieu et al., 2015; Zhang et al., 2018). Instead, we decide to predict the latent code of a future frame through a small motion learning model as shown in Fig. 1(c).

To predict the latent code of frame T_t, we extract latent codes for k previous input frames $T_1 ... T_k$. The extracted latent codes $z_1 ... z_k$ from the encoder for each of past frames $T_1 ... T_k$ are then concatenated along the temporal dimension and used as the input for the motion model to predict the latent code $\hat{z}_t$ for a future frame T_t. We use 3D convolutional layers in the motion model since these can attend to both motion and appearance whereas a 2D convolution layer is only able to work in the spatial direction (Tran et al., 2015). Each convolutional block includes a 3D convolutional layer with kernel size $3 \times 3 \times 3$, stride 2 on the temporal dimension and stride 1 on the feature dimension. This is followed by a BatchNormalization (Ioffe and Szegedy, 2015) and a leaky-relu activation layer. We use three convolutional blocks in the motion model.

3.3. Training

Our proposed framework consists of two parts: video frame reconstruction and latent code prediction, so the objective function to train both components end-to-end can be formulated as:

$$\mathcal{L} = \lambda_r \sum_{q=2}^{k} \sum_{j=1}^{N} \|\hat{T}_{q,j} - T_{q,j}\|_2^2 + \lambda_p \sum_{m=1}^{M} \|\hat{z}_{t,m} - z_{t,m}\|_2^2 + \gamma \|W\|_2^2 \quad (1)$$

The first term measures the pixel-wise reconstruction loss where N is the total number of pixels per frame and k is the number of frames in a input sequence. We can only reconstruct the last $k-1$ frames if we input k frames since the reconstruction of one frame requires the features from its previous frame. The second term is the MSE between the predicted and the observed latent code where M is the number of elements in the latent code. The last term is an L2 regularization term where γ is kept to be 0.001. The model can be trained end-to-end but to bootstrap the encoder with useful features, we first focus on training the autoencoder for reconstruction ($\lambda_r = 1$ and $\lambda_p = 0.001$) until the training loss for reconstruction converges. Then we focus on finetuning the weights of the motion model ($\lambda_r = 0.001$, $\lambda_p = 1.0$) to use the motion information better.

3.4. Inference

At inference time, we discard the decoder and use the difference between predicted and actual latent code as the metric to determine whether a frame is an anomaly or not. The underlying assumption is that the model can predict the latent code for the normal frames with high accuracy but is not able to do so for anomalous frames. We measure the distance between latent codes with Mean Squared Error (MSE) as shown in Eq. (2).

$$s_t = \frac{\sum_{m=1}^{M} \|\hat{z}_{t,m} - z_{t,m}\|^2}{M} \quad (2)$$

After calculating the anomaly score for each frame, following Mathieu et al. (2015), we normalize the score for each frameset $\mathcal{G}$ to the range of [0,1] using Eq. (3). A frame that has the anomalous score

Table 1
Design choices for each evaluation dataset.

	UCSDPed1	UCSDPed2	Avenue	ShanghaiTech
Input size	128 × 192	128 × 192	128 × 224	128 × 224
Num input (k)	8	8	6	6
Encoder block	5	4	4	4

higher than a threshold is considered as anomaly. Depending on the dataset, $\mathcal{G}$ contains all frames of a video or just the frames in a sliding window for long videos.

$$s_t = \frac{s_t - min_{j\in\mathcal{G}}(s_j)}{max_{j\in\mathcal{G}}(s_j) - min_{j\in\mathcal{G}}(s_j)} \tag{3}$$

4. Experiments

In this section, we compare our methods to state-of-the-art approaches on public benchmarks. We do not only focus on detection accuracy but also compare the computational cost of the models since this is often the bottleneck that limits the performance in the real world. In addition, we also evaluate the proposed anomaly detector on multiple distorted environments and show that our method is more robust against changes in lighting and weather that are common in the real world.

4.1. Experimental setup

To evaluate the effectiveness of our proposed methods, we use the same datasets as Liu et al. (2018), including the UCSD Pedestrian dataset (Mahadevan et al., 2010a), the CUHK Avenue dataset (Lu et al., 2013) and the ShanghaiTech dataset (Liu et al., 2018). There are no anomalies in the training data for the UCSD Pedestrian dataset (Mahadevan et al., 2010a). The CUHK Avenue dataset (Lu et al., 2013) is more realistic and challenging in that sense as it contains several outliers in the training data, and some normal patterns that occur in the test data seldom appear in the training data. As is common, we report the Area Under Curve (AUC) score as the accuracy metric.

The experimental settings are shown in Table 1. We use 8 input frames for UCSDPed1 and UCSDPed2 datasets and 6 input frames for the Avenue and ShanghaiTech datasets. More frames are needed to encode the motion and appearance of the much smaller objects in the UCSD Pedestrian datasets than in the Avenue and ShanghaiTech datasets. The resolution of the frames is reduced using bilinear interpolation, keeping the original aspect ratio. As for the architecture, we use five encoder blocks for the UCSDPed1 dataset and four encoder blocks for the other datasets. This is done because the scenes in the UCSDPed1 dataset include more objects and we need to increase the model's capacity in order to encode the appearance- and motion-related features.

We train the model for 50 epochs in an end-to-end fashion with initial learning rate $1e{-}4$ which decays by 0.1 every 20 epochs. The ShanghaiTech dataset contains data from multiple cameras. We trained individual models per camera and observed no significant performance difference with a model that is trained on data from all cameras. We train the model with the Adam optimizer (Kingma and Ba, 2015) for all our experiments.

For the evaluation, we use feature-wise MSE (Eq. (2)) to calculate the anomaly score for all the datasets. We normalize the anomaly score in UCSDPed1 and UCSDPed2 dataset with Eq. (3) using all the frames in a test video. For the ShanghaiTech datasets, we use the same sliding-window approach as Abati et al. (2019).

4.2. Results

Table 2 compares our results with those of other unsupervised deep learning based methods for anomaly detection. Our approach achieves similar and outperforms the existing methods in terms of frame-level AUC score. The decoupled mechanism and the combined learning of appearance and motion information improves the training process and allows the extracted and predicted latent codes to be more representative for the frequently seen events and thus improve the anomaly detection accuracy. We further investigate the role of decoupling, the use of Conv3D layers in the motion model as well as different anomaly metrics in Section 6.

The initial goal of our approach was to develop an architecture that is more efficient than previous approaches. We benchmark our method with the approaches that have publicly available code on an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz with an GeForce GTX 1080 Ti. We reimplemented the methods described by Hasan et al. (2016) in Tensorflow to allow for a fair comparison. The results are displayed in Table 3.

Our method outperforms the other approaches by a large margin in terms of the number of frames that can be processed per second. Compared to the *Latent-Auto* approach (Abati et al., 2019) that also detects anomalies using latent code, we can process 16~45 times more frames per second. The main reason our method is more efficient is because the model independently encodes appearance and motion information. We extract the appearance information using a relatively efficient 2D convolutional network and process the combined features using a small 3D convolutional network, whereas other approaches build the entire network around 3D convolutions making it much more expensive. Because we extract latent codes from individual frames, each frame only needs to be processed once. We can save each of the last k latent codes that are needed in the motion model and reuse them for the next k predictions. In contrast, models that use 3D convolutions process each frame k times, each time at a different position in the stack, predicting a different frame, making it much more computationally expensive. Since we predict future latent codes and use the prediction error in latent space as our anomaly metric, we do not need the decoder part at inference, again reducing the computational cost. Also, compared to other models that use anomaly scoring metrics based on latent codes, we do not impose any distribution constraint on the latent code, giving the model the freedom to fit the data as best as possible.

A disadvantage of working with latent codes is that it is harder to interpret the model. It is however possible to also use the decoder at inference time to generate predicted frames and to measure the pixel wise reconstruction metrics. This allows us to localize the part of the frame that contains the anomaly. We show these results in Fig. 2. The red boxes in each frame are the regions that have the prediction error larger than a threshold. The green boxes show the ground-truth annotations for the Avenue and ShanghaiTech dataset (the UCSD pedestrian dataset only has frame-level labels). These results empirically confirm that our model can detect motion- and appearance-related anomalies, such as the *skater, cyclist, car, running and gymnastics* events (first four columns). The last two columns of Fig. 2 show false positives, frames that were labeled as normal but that were flagged as anomalies by our model. Two of these show noise or camera movements that were not seen during training. It also shows that the model is more likely to incorrectly flag objects as anomalies if they are closer to the camera.

5. Robustness of the model

The datasets we used in the previous section are commonly used datasets that allow us to compare anomaly detection techniques quantitatively. They however all contain relatively clean data, recorded at similar times during the day and under clear weather conditions. These datasets are therefore not necessarily representative of real world

Table 2
Frame-level AUC score with 95% confidence interval (4 runs) on UCSDPed1, UCSDPed2, Avenue and ShanghaiTech datasets. We outperform most of the existing approaches on all the datasets.

	UCSD Ped1	UCSD Ped2	Avenue	Shanghai-Tech
MDT (Mahadevan et al., 2010b)	81.8	82.9	–	–
ConvAE (Hasan et al., 2016)	81.0	90.0	70.2	–
ConvLSTM (Luo et al., 2017a)	75.5	88.1	77.0	–
Unmasking (Ionescu et al., 2017)	68.4	82.2	80.6	–
Hinami (Hinami et al., 2017)	–	92.2	–	–
StackRNN (Luo et al., 2017b)	–	92.2	81.7	–
FFP-MC (Liu et al., 2018)	83.1	95.4	84.9	72.8
LatentAuto (Abati et al., 2019)	–	95.4	–	72.5
AnomalyNet (Zhou et al., 2019)	83.5	94.9	86.1	–
AM-Corr (Nguyen and Meunier, 2019)	–	96.2	86.9	–
MEMAE (Gong et al., 2019)	–	94.1	83.3	71.2
DOR (Pang et al., 2020)	71.7	83.2	–	–
MNAD (Park et al., 2020)	–	**97.0**	**88.5**	70.5
Ours	**85.0** ± **0.3**	95.1 ± 0.4	**88.8** ± **0.3**	**73.9** ± **0.1**

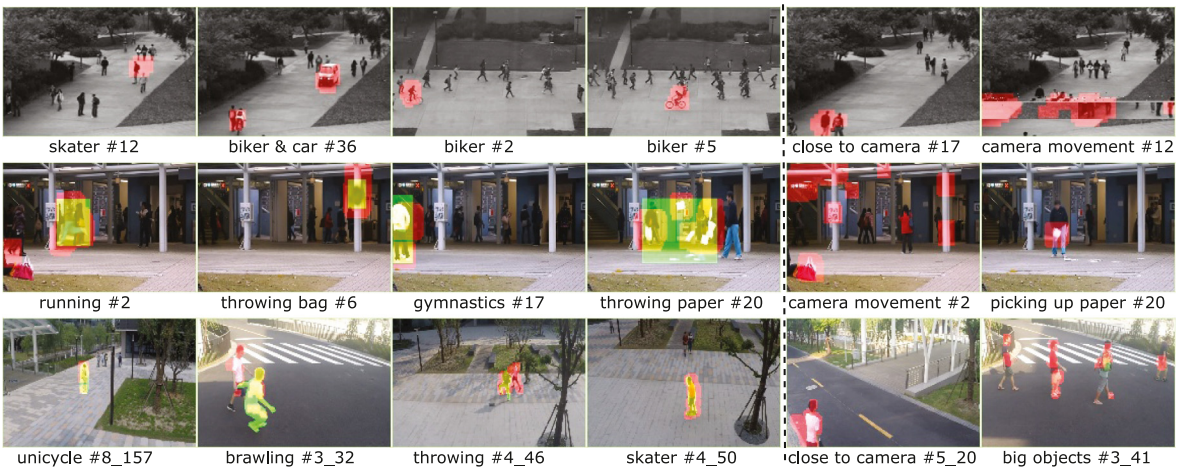

Fig. 2. True positive (first four columns) and false positive (last two columns) detections of our framework. Examples are selected from the UCSD pedestrian dataset (first row), Avenue dataset (second row) and ShanghaiTech dataset (last row). # indicates the testing video index. The red boxes are the regions that have highest pixel-wise prediction error and the green box are the ground truth bounding boxes for the anomalous event. We can successfully detect the motion- and appearance-related anomalies and tend to incorrectly flag objects as anomalies if they are closer to camera. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
FPS for different methods. Our method is more efficient than the existing approaches.

	ConvAE Hasan et al. (2016)	FFP+MC Liu et al. (2018)	LatentAuto Abati et al. (2019)	Ours
UCSDPed1	75	63	2	**81**
UCSDPed2	75	63	2	**90**
Avenue	71	48	5	**77**
ShanghaiTech	71	48	5	**77**

surveillance footage where external factors such as weather and time of day will severely influence the performance of the model. We argue that in addition to their anomaly detection performance and computational cost, we should also compare the robustness and generalization of the models to these external factors. In this section we investigate three types of robustness and show that by working with latent code anomaly metrics we are more robust than other approaches that use pixel-wise metrics.

5.1. Modeling long term temporal information

In Section 4.2, we showed that our approach is by design much more efficient than existing techniques. To reduce the computational cost even further we could reduce the number of times we activate the model. In surveillance video, anomalies are typically in view of the camera during multiple seconds. It should be enough to process only a few frames of this window to detect the anomaly. If instead of running our model every 40 ms, we run it every 200 ms, then this obviously results in a lower total computational cost but this also makes the task much harder for the network since we now need to predict five times further into the future. In this way, the model is forced to encode longer term temporal information and the prediction task is more challenging since the future frame will differ substantially from the previous frames.

To explore this trade-off, we subsample the video sequence and only keep every d_{th} frame in our training and test data. Fig. 3 shows how sensitive the latent code metric (red line) is compared to the pixel-wise metric (green line). Both techniques follow the same trend but the latent code metric consistently performs better than the pixel-wise metric, especially when the gap between input frames becomes large. This illustrates the power of latent codes and their capability of modeling longer term temporal information.

5.2. Generalization to other lightning conditions

The performance of anomaly detection in surveillance video is impacted severely by factors such as varying illumination, multiple weather conditions, on- and off-peak traffic profiles, degradation of the camera and so on. Therefore, in this section, we investigate the robustness of our proposed method to these distortions. We train a model using original frames from the Avenue dataset and then analyze the performance on distorted test set frames. We adjust brightness, blur

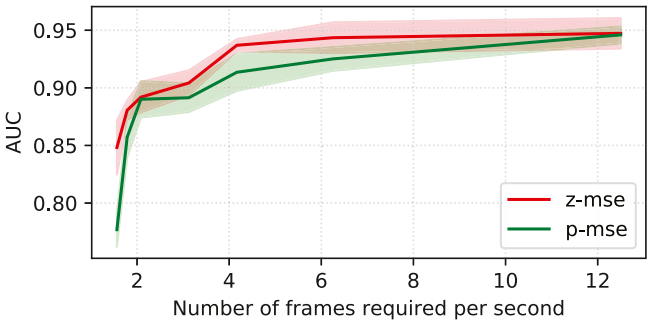

Fig. 3. The anomaly detection accuracy (frame-level AUC score with 95% confidence interval) when we have low fps input. **z-mse** is calculated using Eq. (2) on the latent code and **p-mse** is the MSE between the predicted frames and actual frames. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

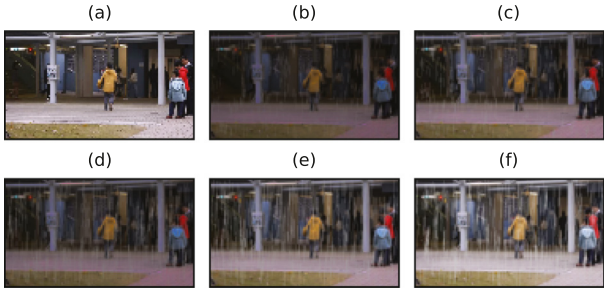

Fig. 4. The distorted frames using Avenue dataset. (a) original frame, (b) and (c) have heavy rain with brightness degree 0.5 and 0.7 respectively and (d), (e) and (f) show torrential rain with the brightness 0.6, 0.8 and 1.0 respectively. The figure is best viewed in color.

the image and add rain to the test frames using the Automold toolkit.[1] Fig. 4 shows some examples of the distorted frames using different levels of rain and brightness.

Fig. 5 shows the frame-level anomaly detection accuracy for different distortion levels. The x-axis shows the relative brightness compared to the original frame. The different curves show the performance of using Mean Squared Error (MSE) in pixel space (p) and in latent space (z) as our anomaly metrics for different levels of rain added to the image. As expected the model performance drops when the brightness decreases, but our model is consistently more robust than the baseline model that uses pixel wise metrics. Adding rain to the test frames also reduces the detection performance but again, our feature-wise latent code MSE performs significantly better as anomaly scoring metric than the pixel-wise MSE. These results verify that our proposed methods using feature-wise MSE in the latent code to identify anomalies is more robust to different outdoor situations than pixel-wise MSE measurements.

5.3. Robustness to already seen anomalies

Most anomaly detection techniques assume that there are no or very few anomalies during training. To determine the impact of anomalies in the training data, we performed an additional experiment where we randomly sample anomalies into the training data. We have compared our results with *FFP* (Liu et al., 2018) and *MEMAE* (Gong et al., 2019). We adopted the official public available codes for methods *FFP* and *MEMAE* and trained them on the dataset which contains both normal frames and randomly sampled anomalies. The result is shown

[1] https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library.

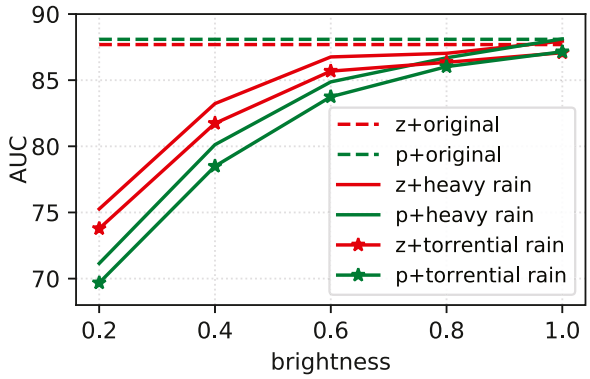

Fig. 5. The averaged anomaly detection accuracy (frame-level AUC score) on the augmented frames where **z** means the latent code feature-wise MSE and **p** means the prediction pixel-wise MSE. The latent code anomalous score measurement is more robust on unseen weather conditions compared to other approaches. The figure is best viewed in color.

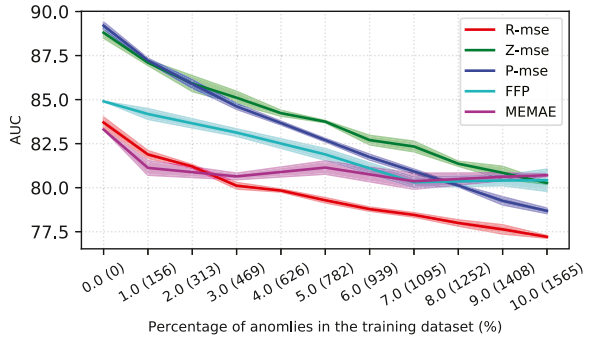

Fig. 6. Anomaly detection accuracy (AUC ± 95% confidence interval) on the CUHK Avenue dataset. The number in the bracket means the number of added anomalous frames. There are 1565 anomalous frames in total. Our approach (*Z-MSE*) achieves better performance and is more robust than other SToAs as the number of anomalies during training increases.

in Fig. 6. On the x-axis, we show the percentage of anomalies in the training data, and on the y-axis, the obtained AUC score. As the number of anomalies seen during training increases, the AUC score goes down. Our approaches outperform the existing works when there are increasing numbers of anomalies in the training dataset. Using the latent code seems more robust to a noisy training dataset than using the pixel-wise difference between reconstructions or predictions.

6. Ablation study

In Section 3 we introduced our model together with some design choices. In this section we look closer to these choices and investigate how these contribute to our results.

6.1. Reconstruction loss

We first look at different loss functions for learning reconstructions. We used pixel-wise Mean Square Error (MSE) as the objective function in our experiments to learn reconstructions as it is commonly used and widely studied (Hasan et al., 2016; Xu et al., 2017; Vincent et al., 2008). It is, however, also known that MSE loss tends to generate blurry frames (Larsen et al., 2016). Therefore, we conducted additional experiments to investigate whether more advanced reconstruction loss functions can further improve the anomaly detection accuracy. We used Gradient Difference Loss (GDL) and Structure Similarity Index Measure (SSIM), following the work from Larsen et al. (2016) and applied them on the CUHK Avenue datasets. Table 4 shows the results. Applying

Table 4
Influence of different reconstruction loss functions on the anomaly detection accuracy using the CUHK Avenue dataset (AUC score ± 95% confidence interval).

	MSE	GDL	SSIM	MSE+GDL	MSE+SSIM	MSE+GDL+SSIM
R-MSE	83.7 ± 0.3	85.7 ± 1.2	80.1 ± 1.0	84.7 ± 0.7	81.1 ± 0.9	85.5 ± 0.8
Z-MSE	89.2 ± 0.2	84.2 ± 0.1	85.2 ± 0.7	86.0 ± 0.3	85.0 ± 0.2	84.4 ± 0.5
P-MSE	88.8 ± 0.3	88.4 ± 0.3	88.7 ± 0.3	88.6 ± 0.2	88.7 ± 0.1	88.0 ± 0.8

Table 5
Abnormal event detection results (in %) with 95% confidence interval. We achieved better performance using latent-code prediction error (*Z-MSE*) as anomalous score and using Conv3D layers in the motion model. Bold numbers correspond to the best performance on each dataset.

	Conv3D				ConvLSTM				Autoregressive
	Ped1	Ped2	Avenue	ShanghaiTech	Ped1	Ped2	Avenue	ShanghaiTech	Ped2
R-MSE	80.6 ± 0.2	91.3 ± 0.5	83.7 ± 0.3	55.8 ± 0.2	74.8 ± 0.4	87.5 ± 0.1	83.1 ± 0.2	52.6 ± 0.4	81.37 ± 0.2
Z-MSE	**85.0 ± 0.3**	**95.1 ± 0.4**	**88.8 ± 0.3**	**73.9 ± 0.2**	82.8 ± 0.6	94.1 ± 0.1	**88.9 ± 0.3**	71.3 ± 0.4	92.31 ± 0.3
P-MSE	82.2 ± 0.4	90.9 ± 0.5	**89.2 ± 0.2**	72.7 ± 0.2	81.2 ± 0.4	90.3 ± 0.8	**89.0 ± 0.3**	70.2 ± 0.2	–

these more advanced reconstruction loss functions can significantly increase the anomaly detection accuracy when we only use reconstruction error to detect anomalies (*R-MSE*). However, it diminishes the performance of using the latent code difference to detect anomalies (*Z-MSE*).

6.2. Reconstruction vs. prediction

To understand the impact of each modules on the detection accuracy, we report the frame level AUC score for the model that is without and with motion learning module in Table 5. Adding the motion model highly improves the anomaly detection accuracy for all the datasets since it encodes spatiotemporal information better. Compared to the performance using pixel-wise prediction error, the use of latent code prediction error tends to be better and more stable since it is more robust to the noise in the image as indicated by Section 5. To qualitatively understand how the model differentiates between normal and abnormal frames, we conduct an experiment on the moving-mnist dataset.

We train the same model as shown in Fig. 1 using the video sequences that are created by letting randomly selected digits 4 and 7 from the training set of MNIST (Lecun et al., 1998) move horizontally or vertically with a speed of 2 following Luo et al. (2017b) (see Fig. 7(a)). This model is then tested on video sequences that include all types of digits from the test set of MNIST dataset and two new shapes (circle and square) that are moving also horizontally or vertically with a speed of 2 or speed of 4. The input, reconstruction, prediction and prediction error are shown in Fig. 7(b) and (c).

Fig. 7(b) shows the model output for input objects that move with a speed of 2. The model can make good reconstructions and predictions for already seen digits 4 and 7 but tend to predict the unseen digits to be one of the already seen digits. For example, it predicts the circle and square to be similar to 4 and digit 8 to be similar to 7. The difference between the reconstruction and prediction indicates that the model cannot make a good prediction of the latent code for the unseen digits and this allows us to detect the appearance related anomalies. For the objects that are moving with a higher speed as shown in Fig. 7(c), the model produces a prediction that falls behind the actual input. This is because the designed motion model can further exploit the speed mismatch of the objects during training and testing and is thus able to detect the motion related anomalies.

6.3. Design of the motion model

The motion model is an important factor in the design of our architecture since it is required to encode the typical appearance and motion information of the frequently seen events. Therefore, we also replaced the Conv3D layers in the motion model with ConvLSTM and autoregressive density estimation layers (Abati et al., 2019) to study the impact of the design of the motion model on anomaly detection performance. The anomaly detection accuracy using pixel-wise reconstruction, prediction error and feature-wise latent code error for different datasets are shown in Table 5. We achieved better or similar performance on all the datasets using Conv3D layers in the motion model. One of the possible reasons is that the Conv3D layers can fit the data better and thus extract more representative features. In addition, we observed that the model that uses ConvLSTM layers have delayed detection results such that it fails to detect the beginning of anomalous events and it also reports more false alarms after the anomalous events due to the slowly response.

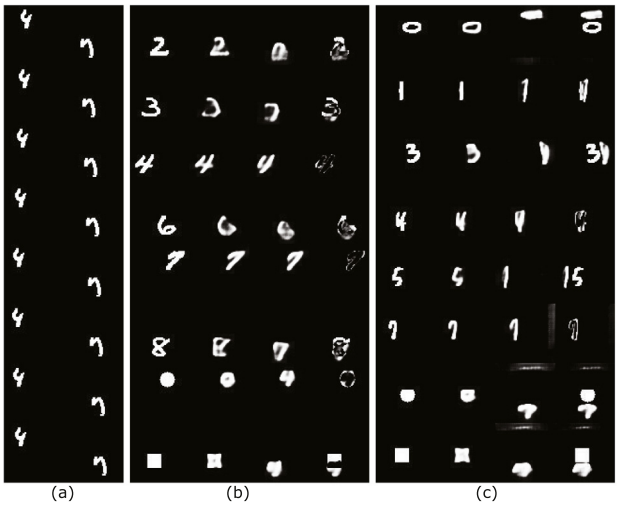

Fig. 7. Experimental results on moving-mnist dataset. (a) training digits 4 and 7 are moving horizontally or vertically randomly with speed 2. (b) and (c) show the testing digits and shapes that are moving in a similar fashion but with speed 2 (b) and speed 4 (c). From left to right, the columns in (b) and (c) are input, reconstruction, prediction and prediction error. The model cannot accurately reconstruct and predict for both appearance-related (unseen digits) and motion-related (unseen moving speed) anomalies.

7. Conclusion and future work

In this paper we introduced a novel architecture that is able to detect anomalies in real world surveillance footage using only unsupervised training. The model consists of two parts where the first part extracts appearance features from individual frames and the second part uses these features to predict the latent code for a future frame. In contrast to previous works, our model uses a prediction in latent space as a metric to detect anomalies. We showed that is able to outperform other techniques that use reconstruction or pixel based prediction metrics. Because of the decoupled appearance and motion feature learning, our model is also much more efficient than related approaches. Where other techniques use expensive 3D convolutions to

analyze a stack of frames, we process each frame individually and then combine the information with a much smaller 3D convolutional model. This allows us to process 16 to 45 times more frames using the same computational budget. Finally, we show that using latent space features makes the model more robust against distortions such as changing lighting or weather conditions.

Anomaly detection in real world surveillance data is a very challenging topic with many useful applications. For future work, we argue that more research is needed to deal with changing environments, weather and lighting conditions as well as with camera degradation.

CRediT authorship contribution statement

Bo Li: Methodology, Software, Formal analysis, Writing - original draft. **Sam Leroux:** Conceptualization, Methodology, Writing - review & editing. **Pieter Simoens:** Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from imec under the CityFlows AAA programme.

References

Abati, D., Porrello, A., Calderara, S., Cucchiara, R., 2019. Latent space autoregression for novelty detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition.

Bouindour, S., Snoussi, H., Hittawe, M.M., Tazi, N., Wang, T., 2019. An on-line and adaptive method for detecting abnormal events in videos using spatio-temporal convnet. Appl. Sci. 9 (4).

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Comput. Surv. 41 (3).

Chong, Y.S., Tay, Y.H., 2017. Abnormal event detection in videos using spatiotemporal autoencoder. In: Cong, F., Leung, A., Wei, Q. (Eds.), Advances in Neural Networks - ISNN 2017. Springer International Publishing.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., van den Hengel, A., 2019. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019. IEEE, pp. 1705–1714.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in Neural Information Processing Systems 27. pp. 2672–2680.

Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S., 2016. Learning temporal regularity in video sequences. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 733–742.

Hinami, R., Mei, T., Satoh, S., 2017. Joint detection and recounting of abnormal events by learning deep generic knowledge. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3619–3627.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T., 2017. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37. pp. 448–456.

Ionescu, R.T., Khan, F.S., Georgescu, M.-I., Shao, L., 2019. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Ionescu, R.T., Smeureanu, S., Alexe, B., Popescu, M., 2017. Unmasking the abnormal events in video. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2914–2922.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O., 2016. Autoencoding beyond pixels using a learned similarity metric. In: Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. pp. 1558–1566.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86 (11), 2278–2324.

Liu, W., Luo, W., Lian, D., Gao, S., 2018. Future frame prediction for anomaly detection – A new baseline. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Lu, C., Shi, J., Jia, J., 2013. Abnormal event detection at 150 FPS in MATLAB. In: 2013 IEEE International Conference on Computer Vision. pp. 2720–2727.

Luo, W., Liu, W., Gao, S., 2017a. Remembering history with convolutional LSTM for anomaly detection. In: 2017 IEEE International Conference on Multimedia and Expo (ICME).

Luo, W., Liu, W., Gao, S., 2017b. A revisit of sparse coding based anomaly detection in stacked RNN framework. In: 2017 IEEE International Conference on Computer Vision (ICCV).

Mahadevan, V., Li, W.-X., Bhalodia, V., Vasconcelos, N., 2010a. Anomaly detection in crowded scenes. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. pp. 1975–1981.

Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N., 2010b. Anomaly detection in crowded scenes. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Mathieu, M., Couprie, C., LeCun, Y., 2015. Deep multi-scale video prediction beyond mean square error.

Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S., 2019. Learning regularity in skeleton trajectories for anomaly detection in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11996–12004.

Nguyen, T., Meunier, J., 2019. Anomaly detection in video sequence with appearance-motion correspondence.

Pang, G., Yan, C., Shen, C., van den Hengel, A., Bai, X., 2020. Self-trained deep ordinal regression for end-to-end video anomaly detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. IEEE, pp. 12170–12179.

Park, H., Noh, J., Ham, B., 2020. Learning memory-guided normality for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14372–14381.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI). In: LNCS, vol. 9351, Springer, pp. 234–241.

Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Netw. 61, 85–117.

Song, X., Wu, M., Jermaine, C., Ranka, S., et al., 2007. Conditional anomaly detection. IEEE Trans. Knowl. Data Eng. 19 (5), 631–645.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497.

Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. pp. 1096–1103.

Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N., 2015. Learning deep representations of appearance and motion for anomalous event detection.

Xu, D., Yan, Y., Ricci, E., Sebe, N., 2017. Detecting anomalous events in videos by learning deep representations of appearance and motion. Comput. Vis. Image Underst. 156, 117–127, Image and Video Understanding in Big Data.

Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 586–595.

Zhou, J.T., Du, J., Zhu, H., Peng, X., Liu, Y., Goh, R.S.M., 2019. AnomalyNet: An anomaly detection network for video surveillance. IEEE Trans. Inf. Forensics Secur. 14 (10), 2537–2550.