3D-Convolutional Neural Network with Generative Adversarial Network and Autoencoder for Robust Anomaly Detection in Video Surveillance

Wonsup Shin∗, Seok-Jun Bu† and Sung-Bae Cho‡
Department of Computer Science, Yonsei University
50 Yonsei-ro, Seoul 133-724, South Korea
∗wshin2013@yonsei.ac.kr
†bjbhan@yonsei.ac.kr
‡sbcho@yonsei.ac.kr
Accepted 28 March 2020
Published Online 28 May 2020

As the surveillance devices proliferate, various machine learning approaches for video anomaly detection have been attempted. We propose a hybrid deep learning model composed of a video feature extractor trained by generative adversarial network with deficient anomaly data and an anomaly detector boosted by transferring the extractor. Experiments with UCSD pedestrian dataset show that it achieves 94.4% recall and 86.4% precision, which is the competitive performance in video anomaly detection.

Keywords: Video anomaly detection; machine learning; transfer learning; generative adversarial network; 3D CNN.

1. Introduction

In recent years, increasing the use of surveillance cameras with less manpower makes the automatic video surveillance systems to become more important. A crucial challenge is automatically detecting anomaly which is roughly defined as unusual, uncommon or irregular event. For example, on a highway, a stopped car or person can be defined as an anomaly. Conversely, on the sidewalk, all objects except people walking along the road can be regarded as anomalies. In other words, objects in video that are unusual or rarely occurring in a given domain are considered as anomalies. Due to this domain-dependent nature and difficulty of sample collection, the video anomaly detection (VAD) is still one of the challenging issues in the field of computer vision, and various researchers are working to solve it.

An anomaly is classified into three types, such as point anomaly, contextual anomaly and collective anomaly, according to the characteristics of occurrence factor. In video, anomalies are detected by the appearance pattern, motion pattern or both patterns of the objects within the frame sequence. Patterns of objects in video can cause contextual anomaly, which is abnormal in a specific context only, and collective anomaly, which is normal from a local point of view but is abnormal in general. Due to the nature of video, however, the point anomaly is not considered in this study.

Figure 1 shows some examples of anomaly in video. In this example, only people walking on two feet are considered as normal. The left most image in Fig. 1 shows an anomaly caused by the appearance pattern. The object in the video is a person in a wheelchair who moves similar speed to a typical pedestrian but has a different appearance. The middle image in Fig. 1 shows the anomaly caused by the motion pattern. The object in the video is a person who is riding a bicycle, and...
W. Shun, S.-J. Bu & S.-B. Cho

Fig. 1. Examples of video anomalies. The red part of each image represents anomaly objects. The anomalies from the left to the right images were generated by the appearance pattern, the motion pattern and both patterns of the object, respectively.

Fig. 2. The mapping diagram between conventional anomaly type and object patterns in video.

Appearance and speed are different from typical pedestrians. Therefore, the methods for the VAD should be able to extract high-quality spatial-temporal features of objects within the video. Figure 2 shows a diagram of mapping object patterns and anomaly types.

Deep neural network (DNN) is one of the key technologies for modeling anomaly. Especially, models like the long short-term memory (LSTM), which can learn temporal information, and the convolutional neural network (CNN), which can extract spatial information, are making tremendous achievements in the field of video feature extraction and classification. Intuitively, it seems that a model combining the two models is easy to extract the characteristics of the video. In the meantime, there are some works that the model with 3D CNN-based auto-encoder alone has shown higher performance.

Most deep VAD classifiers have been modeled in an unsupervised learning manner because it is difficult to collect labeled anomaly data enough to learn the internal variability of anomalies. However, labeled data is very important and valuable information that can guide the learning direction of the VAD classifier. Therefore, when there is even a little bit of labeled abnormal data, unsupervised learning, which does not consider any label information, has its limitations in terms of data efficiency. It is hard to think that unsupervised learning is the best way to solve the VAD. Even semi-supervised learning-based approaches in VAD exclude consideration of labeled abnormal data. These approaches are only interested in configuring a training set with normal data. In this context, research on supervised learning methods is important in terms of using available data as efficiently as possible.

Previously, we developed an anomaly detection model based on supervised learning using generative adversarial network (GAN). This method dealt with the shortage of labeled anomaly data in supervised learning. The key idea used the GAN discriminator as the base model and apply transfer learning to the classifier. Since the GAN can generate data that do not exist in the dataset, the base model has learned from a lot of data that are very similar to the actual data. The problem of insufficient labeled data is addressed. However, the GAN discriminator cannot extract the features of the data effectively, because it extracts the features for classifying real and fake data. The features obtained in this way do not exactly catch up with the tasks of the classifier. It lacked consideration for the ability to extract high-level spatial-temporal features.

In this paper, we propose another VAD model that overcomes the limitations of our previous work. The proposed method can solve the lack of labeled data, while it is possible to extract features suitable for the classifier. The main idea is to convert the GAN generator into an autoencoder model based on 3D CNN and use the encoder of the autoencoder as the base model. Unlike a discriminator, and like a
classifier, it aims to extract features that can model the underlying information of the data. At the same time, because it learns in the GAN framework, it preserves the advantage of being able to learn with abundant data. The contributions of this paper are summarized as follows.

- We propose an end-to-end deep VAD model that utilizes labeled anomaly data, unlike the conventional methods.
- We propose a robust feature extraction model training method using GAN framework even when the anomaly data is insufficient.
- We propose an effective transfer learning method for the VAD classifier.

The rest of this paper is organized as follows. The related works for VAD are introduced in Sec. 2. The proposed model is presented in Sec. 3. Section 4 illustrates the experimental results for evaluating the proposed model. We conclude this paper in Sec. 5.

## 2. Related Works

Numerous approaches have been studied to solve the problem of VAD. As mentioned previously, most studies suggested methods based on unsupervised learning. Table 1 summarizes the related works introduced.

The anomaly detection model is divided into feature extraction and anomaly detection parts. Feature extraction is categorized as CNN-based model, reconstructive model and generative model. In addition, the anomaly detection part can be constructed using a classifier such as one-class SVM or an end-to-end model using fully connected structure or autoencoder structure. Since the key idea of the proposed method lies in the feature extraction, we introduce the related research by focusing on the feature extraction.

First, most feature extraction methods using CNN are based on transferring pre-trained models. Smeureanu et al. used pretrained VGG networks for feature extraction. Similarly, Sabokrou et al. used pretrained AlexNet. These methods alleviate the issue of building a high-quality feature extraction model through transfer learning. However, the domain trained by the pretrained model and the target domain are very different. As another approach

<table>
<thead>
<tr>
<th>Category</th>
<th>Author</th>
<th>Feature extraction method</th>
<th>Learning category</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-based model</td>
<td>Zhou et al.</td>
<td>2D CNN + 3D CNN</td>
<td>Supervised</td>
</tr>
<tr>
<td></td>
<td>Smeureanu et al.</td>
<td>Multi-task Fast RCNN</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Himami et al.</td>
<td>Pretrained VGG net</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Sabokrou et al.</td>
<td>Pretrained AlexNet</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Chu et al.</td>
<td>3D CNN + similarity tree</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Sulhani et al.</td>
<td>3D-ConvNet</td>
<td>Supervised</td>
</tr>
<tr>
<td></td>
<td>Wang and Bi</td>
<td>Parallel CNN</td>
<td>Supervised</td>
</tr>
<tr>
<td></td>
<td>Sun et al.</td>
<td>2D CNN</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Ullah et al.</td>
<td>3D CNN</td>
<td>Supervised</td>
</tr>
<tr>
<td>Reconstructive model</td>
<td>Xu et al.</td>
<td>SDAE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Chen et al.</td>
<td>Trajectory clustering + AE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Hassan et al.</td>
<td>HOOG, HOF + AE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Medel and Savakis</td>
<td>ConvLSTM-AE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Chong and Tan</td>
<td>Spatiotemporal-AE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Zhao et al.</td>
<td>3D ConvAE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Generative model</td>
<td>Ahmadlou and Adeli</td>
<td>Probabilistic NN</td>
<td>Supervised</td>
</tr>
<tr>
<td></td>
<td>Dimokranitou</td>
<td>AAE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Sun et al.</td>
<td>VAE</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Huang et al.</td>
<td>CDBN</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Sabokrou et al.</td>
<td>AVID (GAN)</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Lee et al.</td>
<td>STAN (GAN + LSTM)</td>
<td>Unsupervised</td>
</tr>
<tr>
<td></td>
<td>Shin et al. (2018)</td>
<td>GAN + LRCN</td>
<td>Supervised</td>
</tr>
<tr>
<td></td>
<td>Ravanbakhsh et al. (2015)</td>
<td>GAN Unsupervised</td>
<td></td>
</tr>
</tbody>
</table>
for CNN-based feature extraction, end-to-end models based on supervised or unsupervised learning have also been developed. For example, there have been attempts to do supervised learning by designing a structure that connects a fully connected network (FCN)22 These methods are intuitive, but do not consider the issues of supervised learning in the VAD domain mentioned in the previous section. Besides, many models using 3D CNN-based feature extraction have been proposed. This is because Tran et al. demonstrated that 3D CNNs can outperform RNNs in the video domain.23 For example, Chu et al. proposed a 3D CNN-based method for VAD problem.23 The method showed excellent performance for pixel-unit detection.

The reconstruction models mainly extract features using auto encoder (AE). Often, end-to-end anomaly detection is performed using reconstruction loss as a measure. Xu first applied the AE-based unsupervised learning method to the VAD domain.24 For each frame, AEs composed of 2D convolutional networks were applied to extract video features.25 Chen et al. combined hand-crafted feature extraction methods with AE. Many hand-crafted feature extraction methods such as trajectory clustering, histogram of gradient (HoG) and histogram of optical flow (HoF) were combined with AE.26 Thereafter, the methods of extracting features using only the AE have been proposed consequently. Some researchers devised a feature extraction method of using AE composed of ConvLSTM.27 On the other hand, the AE structure using a 3D convolutional network has been studied steadily.28 Zhao et al. proposed a 3D convolutional network-based AE.29 This method outperformed the LSTM-based methods.

The generative model learns the probability distribution of normal data using models such as deep belief network (DBN),30 variational auto encoder (VAE),31 adversarial auto encoder (AAE),32 and generative adversarial network GAN. The GAN is currently being studied most actively in anomaly detection domain.33,34 But, to the best of our knowledge, all the GAN-based methods operate in an unsupervised learning manner except our previous work.35 Ravanbakhsh et al. suggested a way to use GAN.35 The key idea is to use the discriminator as an ideal classifier immediately after learning the GAN. Since learning proceeds only to normal data, it uses the idea that abnormal data is relatively classified as fake data. Inspired by the idea of Ravanbakhsh et al., Sabokrou et al. proposed an ensemble method to detect anomalies using both GAN’s generator and discriminator.36 Lee devised a GAN model with bi-directional anomalies.37 Our previous study constructed a VAD classifier based on supervised learning and uses transfer learning to solve the issue of constructing high-quality feature extractor.38 It built a base model for transfer learning through GAN trained with the target dataset. The discriminator of GAN learned from the large number of generated data and the actual data is the same as the one learned from a large number of data having the same distribution theoretically as the target domain. It used the feature extraction part of the discriminator as the base model. This study has shown that learning the base model through GAN is promising. However, it lacked the consideration of network structure and transferring strategy.

3. The Proposed Method

This paper proposes a classification model for VAD that learns in supervised manner. The overall structure of the proposed model is shown in Fig. 1. It consists of the preprocessing step and two learning phases. The video clip \( x \) is created by windowing in the preprocessing step and used as input. The first learning phase trains GAN that receives a video clip as input and outputs a fake video clip of the same size. The generator aims to generate video clip as real as possible for deceiving the discriminator of GAN. At the same time, the discriminator aims to distinguish between the generated data and the real data. In this process, the decoder of generator learns the ability to generate data to trick the discriminator better, and the encoder of generator learns more informative feature extraction to help the decoder generate data. In addition, each component of GAN updates its weights not only with the actual data, but also with a large number of fake data generated by the generator.

In the second phase, we transfer the learned encoder of the generator to the VAD classifier and fine-tune the VAD classifier in supervised manner. In this process, the classifier obtains a feature extractor learned with a rich dataset that is very similar to the target data. The data generated by the GAN
Fig. 3. Architecture of the proposed method composed of a preprocessing step and two learning phases. In the phase 1, we train GAN so that the encoder of generator can extract useful features. In the phase 2, we transfer the encoder of generator to VAD classifier and fine-tune it. Here, Enc, Dec and Dis mean encoder and decoder of generator, and discriminator, respectively.

follow the similar distribution of real data. For more details about transfer learning, see Sec. 3.2. After both phases are completed, the performance of classifier is verified using the test dataset.

3.1. GAN for video generation

This section describes the structure of the GAN model used in this paper. For the input video clip $x$, the generator $G$ outputs $\hat{x}$. That is, the generator has a relation like $G(x) = \hat{x}$. The discriminator $D$ receives $x$ and $\hat{x}$ as inputs and predicts whether the given input is real or generated data. At the same time, the generator tries to generate the data so that the discriminator has the difficulty to classify them in the original mechanism of GAN. The learning loss between the two models is given by Eq. (1). Through Eq. (1), each component of GAN updates the weights that guarantee the Nash equilibrium between generator $G$ and discriminator $D$.

$$
\min_G \max_D V(D,G) = \min_G \max_D \left[\mathbb{E}_{x \sim p_d(x)} [\log D(x)] + \mathbb{E}_{\hat{x} \sim p_{\hat{x}}(x)} [\log (1 - D(G(x)))\right], \tag{1}
$$

where $p_d$ is a probability distribution of real data, and $V(D,G)$ is an objective function.

3.1.1. Generator

In order for the encoder $Enc$ to learn properly, it is essential to be able to generate high-level fake data in the decoder $Dec$. To improve the quality of production data, we construct a generator as the U-net structure using a 3D convolution. U-net is an AE-based network designed to minimize the energy loss defined with the distance between original and reconstructed data, and weights are directly given between layers in symmetric positions. According to the original GAN formulation, the similarity between the distribution of data generated by the generator and actual data is a critical factor in the convergence of the loss function $V(D,G)$. From the practical point of view, the U-net can be a partial solution to the problem of the instability associated with convergence. Intuitively, it can be understood that GAN is trained so that the sampled data follow the auto-encoded vectors from the actual data, rather than the random normal vectors. We formulate the generator function $G(\cdot)$ with encoding-decoding process with convolution-pooling operations and up-sampling function $\phi_C(\cdot)$, $\phi_P(\cdot)$ and $\phi_P^{-1}(\cdot)$ as follows:

$$
G(x) = Dec(Enc(x)), \tag{2}
$$

$$
Enc(x) = \phi_P(\phi_C(x)) \tag{3}
$$

$$
Dec(x) = \phi_P^{-1}(Enc(x)) \tag{4}
$$
The 3D convolution demonstrates the performance of spatial-temporal feature extraction and the CNN structure is highly parallel. In detail, the 3D convolution denoted by the weight in the $i$th node from convolution filter $(\alpha, \beta, \gamma)$ in the $l$th layer as $a_{l,i}^{\alpha \beta \gamma}$ is calculated by Eq. (5).

$$a_{l,i}^{\alpha \beta \gamma} = f \left( \sum_{k_1} \sum_{k_2} \sum_{k_3} \sum_{h_1} \sum_{w_1} \sum_{d_1} w_k \Theta_{l,k} \delta^{\alpha \beta \gamma}(h+k_1 \beta + k_2 \gamma + d_1) ight) + b_i,$$  

where $\delta_{l,k}^{\alpha \beta \gamma}$ shows the weight of neural network at index $(h, w, d)$ of the $k$th channel in the $l$th kernel connected to the $i$th layer. Each of $K_l, H_l, W_l$ and $D_l$ denotes the kernel number in the $(l-1)$th layer, height, width and depth, respectively.

### 3.1.2. Discriminator

The discriminator must also be meticulously considered to provide a more beneficial loss to the generator. We construct a discriminator based on the long recurrent convolutional network (LRCN) model. The LRCN consists on CNN and LSTM which models the spatial feature using convolutional operations $\psi_S(\cdot)$, pooling operations $\psi_P(\cdot)$ and LSTM which models the temporal feature using gate operation $\psi_T(\cdot)$. For every video clip $V = (x_1, \ldots, x_T)$ and window size $p$, the convolutional encoding function $\psi_P(\cdot)$ is defined as follows with step $t$:

$$\psi_P(x_t) = \psi_P(\psi_T(x_t)).$$  

The convolution operation, which preserves the spatial relationship between features by learning filters that extract correlations, is known to reduce the translational variance between features. Given the $t$th input state, the encoding network performs the convolution operation $\psi_S(\cdot)$ using $m \times m \times m$ sized filter $w$:

$$\psi_S(x_t) = \sum_{m=1}^{m-1} \sum_{j=1}^{m-1} w_{mk} x_{t+m(j+1)+h}.$$  

The summary statistic of nearby outputs is derived from $\psi_P(\cdot)$ by max-pooling operation. Because the dimension of the output vectors from the convolutional layer is increased by the number of convolutional filters, it must be carefully controlled. Pooling refers to a dimension reduction process used in CNN in order to impose a capacity bottleneck and facilitate faster computation. The max-pooling operation has effects on feature selection and dimension reduction under $k \times k$ sized area with pooling stride $s$:

$$\psi_P(x_t) = \max_{x_{t-s+1}}^{x_t} x_{t-s+1}.$$  

The purpose of convolution-pooling operation is to extract the spatial features from the video clips and deliver it to the following LSTM.

In the LSTM, in order to make the network to easily figure out the relationship of image sequence of large time scale, we use the gate operation defined in Eqs. (8)–(13) the definition of input gate $i_t$, forget gate $f_t$, cell state $c_t$, output gate $o_t$ and hidden value $h_t$ at time step $t$ is as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i),$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f),$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c),$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c + b_o),$$

$$h_t = o_t \odot \tanh(c_t),$$

where $\odot$ denotes the element-wise product and $b$ denotes bias term. After the time-series features are extracted by LSTM, the typical DNN is used to complete the mapping function $\psi(\cdot)$:

$$\psi(x_t) = \psi_L^{-1}(\psi_s^{-1}(\psi_T(x_t))),$$

where a softmax activation function is used in the last layer of DNN so that the output vector is encoded as a probability of real or fake and $i$ denotes to the index of the video clip from overall dataset. The weights of the discriminator are optimized using the backpropagation algorithm based on gradient descent optimization, by minimizing the cross-entropy loss function $\mathcal{L}_{CE}$:

$$\mathcal{L}_{CE} = - \sum_i y_i \log(\hat{y}_i).$$
3.2. Transfer learning

This section describes how to learn a VAD classifier for problem solving. VAD classifier $C$ is composed of CNN-based feature extraction part $Enc_C$ and DNN-based classification part $Cls_C$, like general classifier for solving image domain classification problem.

$$C(x) = Cls_C(Enc_C(x)). \quad (18)$$

The feature extraction part of VAD classifier has the same structure as the encoder of the generator. Through this, it is possible to transfer the encoder learned in the previous phase to the VAD classifier. The strategy of transferring the base model to the target model is also an important issue of transfer learning. Shin and Cho guided the transfer learning method of the DNN model consisting of the feature extractor and the classifier for the four major scenarios as shown in Table 2. Table 2 categorizes in which transfer learning can be used according to several scenarios. The discriminator of GAN is a good base model of transfer learning for VAD classifier, because it is a well-formed model using rich fake data generated by generator and some real data. Even the datasets used for learning discriminators and VAD classifiers are very similar. Finally, since the amount of labeled data in the VAD is usually small, we fine-tune the VAD classifier according to the left-most scenario in Table 2.

Table 2. A guide to transfer learning methods for four major scenarios.

<table>
<thead>
<tr>
<th>Category</th>
<th>Text</th>
<th>Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Checkup</td>
<td>Is there a lot of new training data?</td>
<td>No Yes No Yes</td>
</tr>
<tr>
<td></td>
<td>Is the new data similar to the existing data?</td>
<td>Yes Yes No No</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Learn only the classifier</td>
<td>$++$ $+$ $+$ $+$ $+$</td>
</tr>
<tr>
<td></td>
<td>Learn the classifier and some other layers</td>
<td>$+ + +$</td>
</tr>
<tr>
<td></td>
<td>Learn full-network</td>
<td>$+ +$ $+$ $+$</td>
</tr>
</tbody>
</table>

Algorithm 1: Proposed Algorithm

Phase 1: GAN-based base model training


for $j = 1, \ldots, ep_k$:

for $i = 1, \ldots, n$:

sample a mini-batch of video clips $X_i$; $X$

get a result of $D(x_i)$ and $D(\hat{x}_i)$

update $\theta_G, \theta_D$ through equation (1) with adam optimizer

Phase 2: Transfer learning

Input: video clip dataset $X$, data label $L = [0, 1]$.

VAD classifier $C = (Enc_C, Cls_C)$, number of epochs $ep_{\tau}$, parameter of $C$ $\theta_C$.

Output: anomaly score $A$.

for $j = 1, \ldots, ep_{\tau}$:

for $i = 1, \ldots, n$:

sample a mini-batch of video clips $X_i$; $X$

calculate the anomaly score $A = A(x)$

update the $\theta_C = \theta_C + \Delta \theta_C - \Delta \theta_C$ with adam optimizer

Fig. 4. The pseudocode of the proposed VAD method.

3.3. Anomaly detection

The last fully connected layer of the VAD classifier is designed to have sigmoid activation. The final output value $A$ has a value between 0 and 1, and we treat this value as anomaly score. Thus, when $A$ exceeds the threshold $T$, the given input is treated as an anomaly. Equal error rate (EER) is used as the threshold. EER means the point where the false positive rate is equal to the false negative rate. Equation (19) represents the result of prediction $\tau$ on the input data. Figure 4 shows the training algorithm of the proposed method.

$$\tau = \begin{cases} \text{Normal} & \text{if } A < T \\ \text{Abnormal} & \text{otherwise}. \end{cases} \quad (19)$$

4. Experiments

4.1. Datasets and experimental settings

To verify that the proposed method is useful, we conduct the experiments using UCSD pedestrian-1, 2 dataset and Subway entrance and exit dataset. UCSD dataset consists of a surveillance camera video that captures the sidewalk. It defines everything except the person who is walking sidewalk as abnormal. Subway dataset consists of a surveillance camera video that captures the images in the subway.
ticket gates. This defines no payment, wrong direction movement and loitering as abnormal.

Figure 5 shows the data preprocessing process to determine the ground truth (label). Here, \( p \) is the size of window, which should be larger than 1, and \( n \) is the number of video clips \( V \). As we said in Sec. 1, we do not consider point anomaly. In other words, an anomaly occurs over several frames. A few frames-delayed alarms for anomaly are unnoticeable by humans. Thus, we simply label anomaly if the video clip contains more than half anomaly frames. Additionally, due to the limitations in learning speed and memory storage capacity, each image is resized to 180 pixels horizontally and 120 pixels vertically.

Figure 6 shows the specification of all the experiments. In experiment 1, we use machine learning methods, our previous works and supervised learning-based state-of-the-art methods for comparison. The results labeled “GAN + LRCN” represent the performance in Ref. 33 and the results marked “AE + LRCN” are the performance of the model in which the base model is replaced by AE rather than GAN. The results labeled “LRCN” are from the experiments using the LRCN model from the beginning.

4.1.1. Model structures

The generator of the proposed method consists of two 3D convolutional networks and two 3D deconvolutional networks with \( 5 \times 5 \) and \( 3 \times 3 \) filter sizes, respectively. Each layer gives stride 2 for pooling. Along the U-net structure, symmetrical layers have skip connections. The number of channels is 16 and 32 in order. Also, batch normalization is performed for each layer to ensure learning stability. The
The discriminator consists of three 2D time-distributed convolutional networks with $7 \times 7$, $5 \times 5$, and $3 \times 3$ filter sizes, an LSTM layer that receives the output of the last convolutional network, and two fully connected layers. In order to prevent the number of parameters from becoming too large, we set the channel of all convolutional networks to 16 and the output size of LSTM is fixed to 1024. The numbers of nodes in the fully connected layer is set to 256 and 2, respectively. The fully connected layer of the VAD classifier of the proposed method is also designed to be identical to that of the discriminator. We set the activation function of the output layer to softmax and the activation function of all the other layers to ReLU. We use the same network structure for all the experiments.

### 4.2. Performance of VAD

Figure 7 shows the comparison of 10-fold cross-validation results of the proposed method and various comparable models on UCSD pedestrian-1 dataset. Experimental results show that the proposed method yields the highest performance. The proposed model has about 94.4% recall and about 86.4% precision performance, where the precision represents the probability of no false alarm and the recall represents the probability of no anomaly missing. Figure 8 shows the result of the detailed comparison with the previous works. The proposed method shows similar recall performance with the previous works. However, it achieves about 8% higher precision and about 5% higher accuracy. It implies that the proposed method is more robust in anomaly detection than the previous works.

Table 3 shows the mean, variance and t-test results to show whether the performance between the proposed method and the comparable methods is significantly different. The values of the third column are the results of a t-test between the proposed method and the comparable methods, respectively. As a result of the experiment, the t-value is less than 0.05 in both methods, supporting that the performance improvement of the proposed method is statistically significant. Moreover, the proposed method has the highest average accuracy among all the methods, but at the
same time has the highest standard deviation. This implies that the proposed model may be somewhat unstable. However, the absolute magnitude of the standard deviation is small to be regarded as a disadvantage compared to the base model. In addition, we conduct a performance comparison with the conventional supervised learning-based VAD models on four datasets. For this comparative study, we adopt the supervised learning models mentioned in Sec. 2. We use EER and area under curve (AUC) for the performance metrics. Table 4 shows the experimental results. As can be seen, the proposed method produces significantly better performance in all the four datasets. In particular, the proposed method yields higher performance in all the cases than our previous study.

### 4.3. Ability of data generation

To demonstrate that the generator can produce a realistic video clip, we visualize and analyze a fake video clip generated by the trained generator. Figure 9 shows four generated video clips plotted frame by frame. A row means a video clip. If you follow the sequence of each line, you can see that the object moves little by little every frame. Figure 10 shows the result of one generated video clip every three frames. In this figure, the motion of the object appears more clearly. Table 6 shows the experimental results of whether the generated data are similar to the actual data quantitatively. The value in the row “Actual” is the result of comparing the similarity between randomly sampled real data, and the value in the row “Generated” is the result of comparing the similarity between randomly sampled real data and generated data. We have repeated the comparisons 30 times and averaged them for fair comparison. They show that the fake data is very similar to the actual data. Interestingly, the similarity of the row “Generated” is slightly higher than the row “Actual”. This result is caused because the generated data is blurry compared to the actual data. The generator might be learned to follow the distribution of the actual data rather than the detailed characteristics of the actual data.

### 4.4. Case analysis of misclassification

In this section, we analyze the UCSD pedestrian-1 dataset using the $t$-SNE method and analyze the false-positive cases to more closely examine the anomaly detection performance of the proposed model. The $t$-SNE is a method of clustering high-dimensional data on a two-dimensional plane, so that
Fig. 9. Plotting the generated video clips. One row means one image sequence. You can see that the object moves slightly in every frame.

Fig. 10. Plotting a generated video clip by 3 frame units. You can see the object change over time more clearly.
Table 6. Comparison of similarity between generated data and actual data.

<table>
<thead>
<tr>
<th></th>
<th>SSIM</th>
<th>Cosine Sim.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td>0.6201</td>
<td>0.9800</td>
</tr>
<tr>
<td>Generated</td>
<td>0.6341</td>
<td>0.9806</td>
</tr>
</tbody>
</table>

the neighboring data on a two-dimensional plane share similar features. The left side of Fig. 11 shows the result of clustering the test data used in this experiment. The red dots indicate abnormal data, and the blue dots indicate normal data.

Since a number of clusters have been created, data are considered to have a wide variety of characteristics. Besides, in the middle of the t-SNE, we can see that the abnormal data and the normal data are mixed together. This indicates that the anomaly detection in the dataset used in our experiments is very difficult. And it supports the superiority of the proposed model, because it detects about 97% of the anomalies in this dataset. The right side of Fig. 11 also shows the data for the top four miss-classification cases. Each image represents the first frame of each video. All misclassified cases are the false negative; that is, incorrectly classified normal data as abnormal.

However, in all the cases, the objects defined as anomaly tend to be temporarily included in the start or end of a video block. This means that the classification performance of the model is greatly reduced in the video block where the anomaly starts or ends. In the right images of Fig. 11, the objects in the red box are classified as abnormal. Because of this fact, it is presumed that the misclassification causes not only from the classification.

Above all, the error of several frame units occurring at the boundary between the abnormal case and the normal case in the image captured by the modern equipment is not at a human level. Therefore, the misclassification case of the proposed model is not a problem in practice.

4.5. Discussion

As most of the CNN-based models for anomaly detection assume a situation in which anomaly types and characteristics are clearly defined, they simply classify the object inside the data, and decide the abnormality with predefined rules. They target to detect anomaly in specific domains such as identification of thermal infrared face, estimation of concrete, and detection of vehicle type whereas the proposed model attempts to detect abstract anomalies as the unsupervised learning-based models.

![Fig. 11. t-SNE results in the last layer and the analysis of top four false positive cases. When false positives occur, the anomaly object is occluded by another object, or the boundary point where the data label changes.](image_url)
do. That is to say, we focus on the supervised approach of anomaly detection, unlike the previous works based on unsupervised and semi-supervised approaches which cannot use the label information on anomaly.

However, general anomaly detection basically requires additional mechanism to cope with the unseen anomalies. We can tackle this problem by extending the proposed supervised approach or adopting unsupervised and semi-supervised approaches. In the future, we will strive hard to address the both directions.

5. Concluding Remarks

It is an important issue to develop a model based on supervised learning to work out the problem of VAD. Previously, we developed a method to address the shortage problem of labeled anomaly data that occurs in supervised learning. The key idea was to create a base model using the GAN and use it for transfer learning. In this paper, we have proposed another novel supervised learning-based method to address the problems of the data shortage as well as the high-level feature extraction. The proposed method attempts to construct a source domain as similar to the target domain for higher performance. Experimental results show that it is the best among the competitive models in accuracy, and the amount of improvement is statistically significant. In addition, we have discussed the advantages and disadvantages of the model through careful investigation of the video generation and false positive error.

As a future work, we will explore ways to address the unseen data problem, one of the main limitations of supervised learning, in VAD.

Acknowledgment

This work was supported by Defense Acquisition Program Administration and Agency for Defense Development Under the Contract (UD190016ED).

References

1. S. Fleck and W. Straßer, Privacy sensitive surveillance for assisted living — A smart camera approach, in Handbook of Ambient Intelligence and Smart Environments (Springer, 2010), pp. 985–1014.
15. Z. Chen, Y. Tian, W. Zeng and T. Huang, Detecting abnormal behaviors in surveillance videos based on fuzzy clustering and multiple Autoencoders, IEEE Int. Conf. on Multimedia and Expo (2015), pp. 1–6.


