Video anomaly detection using unsupervised deep learning methods

Yan, Mengjia

2018

Yan, M. (2018). Video anomaly detection using unsupervised deep learning methods.
Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/88812>

<https://doi.org/10.32657/10220/46011>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**VIDEO ANOMALY DETECTION USING
UNSUPERVISED DEEP LEARNING METHODS**

YAN MENGJIA

**SCHOOL OF ELECTRICAL & ELECTRONIC
ENGINEERING**

2018

VIDEO ANOMALY DETECTION USING UNSUPERVISED DEEP LEARNING METHODS

YAN MENGJIA

YAN MENGJIA

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Master of Engineering

2018

Acknowledgments

My sincerest gratitude goes first to my previous supervisor, Professor Junsong Yuan, for his full support, encouragement, and guidance. He has walked me through the world of computer vision, and guided me to form the scientific research thinking. His open-mindedness and conscientiousness make a profound impact on my research work and my life attitude. I would also like to show my honest gratitude to my current supervisor, Professor Xudong Jiang, who has instructed and helped me a lot in the past few months.

Besides, I own my sincere gratitude to all of the excellent members of our research group. They offered me encouragement and inspiration not only on research work but also on my life. My thanks would also go to my beloved family for their generous considerations and confidence.

Lastly, I have much appreciated the School of Electrical & Electronic Engineering, NTU, for providing me with the wonderful environment and equipment for my research work.

Contents

Acknowledgments	i
Contents	ii
Summary	v
List of abbreviations	vii
List of tables	viii
List of figures	viii
1 Introduction	2
1.1 Background	2
1.2 Motivation	3
1.3 Objectives	4
1.4 Contributions	4
1.5 Organization of the Report	5
2 Related Work	7
2.1 Video Anomaly Detection	7
2.1.1 Trajectory-based Approaches	7
2.1.2 Dynamic Textures	11

Contents

2.1.3	Sparse Reconstruction	14
2.1.4	Deep Learning Based Methods	16
2.2	Generative Adversarial Network	18
2.3	Autoencoder	19
3	Temporal Anomaly Detection using 3D-GAN	22
3.1	Introduction	22
3.2	Proposed Method	25
3.2.1	Video Temporal Anomaly Detection	25
3.2.2	Proposed Network	26
3.3	Experiments	28
3.3.1	Results of Avenue Dataset	30
3.3.2	Results of UMN dataset.	31
4	Spatiotemporal Anomaly Detection using 3D-FCAE	34
4.1	Introduction	34
4.2	Propose Method	36
4.3	Fully 3D Convolutional Autoencoder	37
4.3.1	Autoencoder	37
4.3.2	3D Convolution and Deconvolution	38
4.3.3	Fully 3D Convolutional Autoencoder	39
4.4	Learning Fully 3D Convolutional Autoencoder	40
4.4.1	Network Architectures	42
4.5	Testing Fully 3D Convolutional Autoencoder	43
4.5.1	Global Anomaly Detection	44
4.5.2	Local Anomaly Detection	44
4.6	Experiments	45

Contents

4.6.1	Datasets and Implementation Details	46
4.6.2	Exploration Study	47
4.6.3	Evaluation on Video Anomaly Detection	50
4.6.4	Evaluation on Running Detection	54
5	Conclusions and Future Work	56
5.1	Conclusions	56
5.2	Future Work	57
5.2.1	On-line Video Anomaly Detection	58
5.2.2	Video Anomaly Detection using PU Learning	58
	List of publications	60
	Bibliography	60

Summary

Video anomaly detection has played a significant role in computer vision and video surveillance tasks. It is concerned about security applications which are much needed by academy and industry. Different from other video analysis tasks such as action detection and action recognition, the deviation between the normal and the anomaly, including appearance and motion, is the crucial measurement we use to determine the anomaly. However, different scenarios will have different normal patterns, which leads to the various definition of deviation. This yields the objective definition problem to video anomaly detection. Another challenge is the limited abnormal samples: abnormal events and behaviors are unusual temporal or spatiotemporal parts of videos. It brings difficulties when we formulate the video anomaly detection problem: some effective methods such as supervised learning methods are impractical to employ.

Much effort has been made to achieve video anomaly detection using object tracking, dynamic textures, and sparse reconstruction, for example. However, the majority of these methods employ low-level features and separate classifiers, leading to massive computational and memory cost. To address the above challenges and reduce the computational and memory cost, in this thesis, we propose unsupervised deep learning and end-to-end methods for temporal and spatiotemporal anomaly detection, respectively.

For temporal anomaly detection, we formulate it as *fake data* detection via

Summary

the discriminative framework of a designed 3D-GAN. This new formulation only employs normal videos during the training phase and detects anomalies according to the deviation estimated by the discriminator of 3D-GAN. We treat normal videos as real data and construct a 3D-GAN to learn the distribution of normal videos during the training phase. Since testing data contain abnormal videos or fake data, whose distribution is different from normal videos/real data, we employ the trained discriminator of our networks to detect temporal normal and abnormal segments. Experiments show that 3D-GANs outperforms 2D-GANs in temporal anomaly detection, and demonstrate the effectiveness and competitive performance of our approach on anomaly detection datasets.

For spatiotemporal anomaly detection, we design a 3D fully convolutional autoencoder that is trainable in an end-to-end manner to learn the spatiotemporal representation of normal visual patterns. Subsequently, spatiotemporal patterns can be detected as blurry regions that are not well reconstructed. Our approach can accurately locate temporal and spatiotemporal anomalies thanks to the 3D fully convolutional structure and the careful design of the architectures. We evaluate the proposed autoencoder for detecting abnormal spatiotemporal patterns on benchmark video datasets. Compared with state-of-the-art approaches, experiment results demonstrate the effectiveness of our approach. Moreover, the learned autoencoder demonstrates good generalizability across multiple datasets.

List of Abbreviations

GAN	Generative Adversarial Network
3D-GAN	3D Convolutional Generative Adversarial Network
2D-GAN	2D Convolutional Generative Adversarial Network
3D-FCAE	3D Fully Convolutional Autoencoder
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
EER	Equal Error Rate
TPR	True Positive Rate
FPR	False Positive Rate
PU learning	Positive and Unlabeled learning

List of Tables

3.1	Temporal anomaly detection performance of Avenue	30
3.2	Temporal anomaly detection performance of UMN	31
4.1	Global abnormal event detection performance	51
4.2	Pixel-level EER and AUC of abnormal event localization for UCSD Ped2 dataset	53
4.3	Pixel-level EER and AUC and Path-based localization (PA) of run- ning detection on the NTU-run dataset	55

List of Figures

2.1	Dynamic Hierarchical Clustering (DHC) method	9
2.2	Application of [8]. (a)-(d) is normal and (e)-(h) is abnormal	9
2.3	Particle trajectories in [10]	10
2.4	Mixture of DTs in [4]	12
2.5	Multi-scale HOF in [2]	15
2.6	The possible combination of basis in [3]	17
2.7	Conditional GAN	18
2.8	The structure of multi-scale video prediction in [10]	19
2.9	The structure of autoencoder in [27]	20
3.1	Top: training phase. Normal video segments are taken as the input of the 3D-GAN. Bottom: testing phase.	23
3.2	Top: structure of Discriminator. The input of Discriminator is $c \times$ $8 \times 64 \times 64$. c is the number of channels. Bottom: structure of Generator.	24
3.3	3D convolution procedure: 3D convolution considers temporal di- mension. (The fourth dimension channel did not show in this figure.)	27
3.4	Qualitative results of temporal anomaly detection. The X-axis: confidential score. Y-axis: the frame number. Blue line: confidential scores. Red shaded regions: ground-truth anomalies.	28

List of Figures

3.5	Visualizing feature maps (last layer, 512 dimensions) of Avenue dataset using t-sne.	31
3.6	Visualizing feature maps (last layer, 512 dimensions) of UMN dataset using t-sne.	32
3.7	Comparison of our method and 2D-GAN via ROC curve.	32
4.1	Spatiotemporal anomaly detection. Blue bounding boxes denote the abnormal objects in the original frames (1 st column). Best viewed in color.	36
4.2	Encoder's output is the hidden middle layer which is a high dimensional representation of the input; decoder tries to reconstruct the input of the whole system.	37
4.3	(a) 2D convolution procedure: feature maps convolve with a 2D convolution kernel to get one channel feature map. (b) 3D convolution considers temporal dimension.	38
4.4	Training procedure and testing procedure of our 3D-FCAE.	40
4.5	Our 3D-FCAE consists of three 3D convolutional layers and three 3D deconvolutional layers.	41
4.6	Deep 1 : 9408 units in the hidden middle layer with 18 layers in the encoder. Deep2: 12544 units in the hidden middle layer. Shallow: 12544 units in the hidden middle layer with 3 layers in the encoder.	48
4.7	Top image: the original frame. Middle row: the reconstructed images from the deep networks with different number of units in middle hidden layers. Bottom image: the reconstructed image from the shallow network. Number under images: number of units in hidden middle layers. The white car is the abnormal objects in this dataset.	49
4.8	Training loss of different depth of the encoder.	50
4.9	(a) training loss of networks with different activation function. (b) training loss of networks using different momentum during optimization.	51

LIST OF FIGURES

4.10	Qualitative results of global anomaly detection. Blue line : regularity scores. High regularity score implies high probability of being normal. Red shaded regions: ground-truth anomalies.	52
4.11	Examples of local anomaly detection results. First line: original frames. Second line: reconstruction frames. Third line: results of detection based on original frames and reconstruction frames. Red shaded region: detected abnormal parts.	53
4.12	NTU-run examples. The red rectangular boxes in the original frames (top row) show the ground truth of running person, which are poorly reconstructed (<i>i.e.</i> , blurry) in the reconstructed frames (bottom row).	54
4.13	Running detection: ROC curves of Hasan <i>et al.</i> and our method. .	55
5.1	Online video anomaly detection.	57
5.2	Training procedures of PU learning.	58

Chapter 1

Introduction

1.1 Background

Video anomaly detection, as a part of video surveillance, is of great importance in both research and industry in recent years. Anomaly defined in Oxford English Dictionary as something that deviates from what is standard, normal, or expected. This definition indicates three variables: normal, deviation, anomaly. These variables show that the anomaly can be detected by determining the normal and the deviation precisely. Subsequently, the deviation information of videos is the crucial point in video anomaly detection. Different from other video analysis tasks like action recognition which extracts complete features to perform classification of every action, video anomaly detection just focuses on the video information which illustrates the deviation.

Scenarios of video anomaly detection include traffic scenario, entrance scenario, road scenario (for pedestrians), crowds behaviors scenario like fighting and running away, and so on. In the traffic scenario, cars moved in the wrong direction are the anomaly. In the entrance scenario, people entered in the wrong direction are abnormal. Pedestrians are normal in both road scenario and crowds behaviors

scenario. The anomaly in the road scenario are cars, bicycles, skateboards, and so on. In this thesis, we will focus on road scenario and crowds behaviors scenario where crowds will increase the difficulty of detection.

1.2 Motivation

Based on the background mentioned in Section 1.1, we can conclude the following challenges in video anomaly detection.

- 1 Anomalous events are hard to define. Some behaviors are normal in one situation but abnormal in the other. Formulating anomaly detection to one-class problem whose outliers are the anomalies can be a practical way to solve the objective definition problem.
- 2 Datasets and labels available are usually insufficient since the anomalies are unusual temporal segments or spatiotemporal parts of videos.
- 3 Supervised learning algorithms are impractical due to limited anomalous events.
- 4 End-to-end learning algorithms for video anomaly detection is scarce.

Many works have exploited video anomaly detection such as sparse reconstruction [2, 3], dynamic textures [4, 5] and trajectory-based approaches [6]. However, the majority of these methods employ low-level features and separated classifiers, leading to massive computational and memory cost.

To address the above challenges and reduce the computational and memory cost, we are planning to apply unsupervised deep learning methods to video anomaly detection and achieve end-to-end learning in this thesis.

1.3 Objectives

Video anomaly detection can be classified into two aspects: temporal (global) detection and spatiotemporal (local) detection [2]. In temporal video anomaly detection, we just need to predict the label of every frame or find the temporal segments of the anomalies. This is used in some crowds behaviour scenarios when the spatial localization of crowds behaviors is trivial. In spatiotemporal video anomaly detection, we need to localize the anomalies in every frame and predict the label of every pixel.

In this thesis, we focus on these two aspects of video anomaly detection and propose methods for temporal and spatiotemporal detection, respectively.

For temporal anomaly detection, we aim to propose an approach to extract the high-level features of video segments and predict the labels of video segments in an end-to-end manner. For spatiotemporal anomaly detection, we aim to propose an end-to-end learning method which can extract the high-level features of video segments and obtain the labels of every pixel. Since the anomalies are scarce in the datasets, we just consider unsupervised learning methods.

1.4 Contributions

To solve the challenges mentioned in Section 1.2 and achieve the objective mentioned in Section 1.3, we proposed two novel frameworks for temporal anomaly detection and spatiotemporal anomaly detection, respectively.

For temporal anomaly detection, we formulate it as *fake data* detection via the discriminative framework of 3D Convolutional Generative Adversarial Networks (3D-GANs). GAN introduced in [7] is used to learn generative models by integrating discriminative framework. During the training phase, the generative framework tries to generate fake data, which are similar to real data, to fool the discrimina-

tive framework, while discriminative framework focuses on distinguishing real data and fake data. Generative framework and discriminative framework will learn the distribution of real data when the training is approaching Nash equilibrium. To achieve anomaly detection, we treat normal data as real data in the training, using GAN to learn the distribution of normal data. Discriminative framework will treat abnormal data as fake data and assign lower scores to them since the distribution of the anomalies are different from normal data.

For spatiotemporal anomaly detection, we propose a 3D fully convolutional autoencoder (3D-FCAE) to learn the normal pattern and detect the spatiotemporal anomalies through the reconstruction blur during the testing phase. It's hard for trained 3D-FCAE to reconstruct the abnormal videos as good as normal videos, since it is only sensitive to the normal videos. Through the reconstruction blur, we can predict the label of each pixel. This method is end-to-end learning with little-to-no supervision.

We evaluate the proposed methods on video anomaly detection datasets. Experiments demonstrate the effectiveness and competitive performance of our approaches.

1.5 Organization of the Report

The organization of this thesis is described as follows:

Chapter 2 introduces the background of our objective tasks and previous methods. First, previous works on video anomaly detection are discussed. Then we discuss existing works using autoencoder. Finally, we introduce related works using GAN.

Chapter 3 introduces a novel formulation of temporal anomaly detection. We formulate video temporal anomaly detection as fake data detection via the dis-

criminator of 3D-GAN. We apply the unsupervised learning of GAN and integrate 3D convolution into it.

Chapter 4 proposes a 3D fully convolutional autoencoder to reconstruct the normal videos. The spatiotemporal anomalies can be detected through the reconstruction error.

Chapter 5 presents the conclusion of this thesis while the possible future works are also discussed.

Chapter 2

Related Work

2.1 Video Anomaly Detection

Video anomaly detection is a crucial application of video surveillance. Low-level features are introduced to handle video anomaly detection. [13] models a hierarchical codebook for the dominant events based on HOG. [14] employs Gaussian process regression to constructs a codebook of interaction templates based on 3DSIFT [15]. Social Force model is introduced by [11]. The authors estimate interaction forces by Social Force and detect anomalous events using the bag of words. Many existing works try to handle anomaly detection task by first learning the normal patterns from normal training videos, then detecting the anomalies as deviations from the normal patterns, such as trajectory-based approaches, dynamic textures, and sparse reconstruction.

2.1.1 Trajectory-based Approaches

Since video anomaly detection considers much information about motion of objects, tracking objects using trajectories is employed in early works of video anomaly detection [6, 8–10].

The main idea of [6] is clustering trajectories of objects in videos and treating outliers as anomalies. The new trajectory can be distinguished by comparing with cluster models in the testing. They employ single-class SVM clustering to model trajectories. SVM performs classification by maximizing the margin between support vectors and classification plane:

$$\begin{aligned} & \min_w \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) \geq 1, \quad i = 1, \dots, m \end{aligned} \quad (2.1)$$

The decision function thus is:

$$f(\mathbf{x}) = \text{sign}\left(\sum_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) - b\right) \quad (2.2)$$

In this paper, they use fixed trajectories since kernels are based on fixed-dimension spaces. They smooth trajectories with a running-average filter and evenly subsample trajectories into a list of 16 2D coordinates. Gaussian kernel is employed in SVM, since Gaussian kernel considers the Euclidean distance between features. The main application of this paper is urban roads.

In paper [8], the video anomaly detection problem is formulated as unsupervised clustering whose dominant groups are treated as normal data. They first employ Hidden Markov Models (HMMs) to characterize object trajectories. Then they computer dissimilarity of two trajectories based on Bayesian information criterion (BIC):

$$\begin{aligned} d(i, j) &= BIC(ij, \dots) - BIC(i, j, \dots) \\ &= \log L_i + \log L_j - \log L_{ij} - \frac{1}{2} K_0 \log N \end{aligned} \quad (2.3)$$

In the first term, N trajectories are model by N HMMs. While in the second

-
- 0) *Initialization*: each trajectory in the dataset forms a group and is fitted with an HMM. There are N groups with N HMMs;
 - 1) *Dissimilarity Measurement*: calculate the dissimilarity $d(i, j)$ for every two groups i and j in the dataset by the measure in (5);
 - 2) *Merging*: the two groups $\hat{i}$ and $\hat{j}$ with smallest $d(\hat{i}, \hat{j})$ ($d(\hat{i}, \hat{j}) < 0$) are merged; if there is no $d(i, j) < 0$, the clustering terminates;
 - 3) *Reclassification*: a new HMM $\theta_{\hat{i}\hat{j}}$ is trained, replacing $\theta_{\hat{i}}$ and $\theta_{\hat{j}}$; then based on the $(N - 1)$ HMMs, all trajectories are reclassified into $(N - 1)$ groups by the maximum likelihood (ML) criterion;
 - 4) *Retraining*: $(N - 1)$ HMMs are retrained based on the updated $(N - 1)$ data groups, respectively;
 - 5) *Update*: $N = N - 1$; go back to step 1).
-

Figure 2.1: Dynamic Hierarchical Clustering (DHC) method

Figure 2.2: Application of [8]. (a)-(d) is normal and (e)-(h) is abnormal

term, trajectories i and j are modeled and merged together by one HMM. To handle the overfitting problem of HMMs, This paper proposes a Dynamic Hierarchical Clustering (DHC) method, shown in Figure 2.1. To minimize BIC, a 2-depth search based on early pruning of branches is proposed. After obtaining clustering results, a probabilistic framework is employed to find dominant groups. The application of this paper is urban roads as well (Figure 2.2).

Figure 2.3: Particle trajectories in [10]

[10] extracts trajectories based on crowd flows. After computing optical flow of each clip (U_w^t, V_h^t) , they perform the particle advection by sub-pixel level optical flow interpolation. Numerically, the position vector (X_w^t, Y_h^t) can be obtained by solving evolution equations:

$$X_w^{t+1} = X_w^t + U_w^t, \quad Y_h^{t+1} = Y_h^t + V_h^t \quad (2.4)$$

which consists of particles trajectories. The particle trajectories are shown in Figure 2.3. Then k-means and iterative clustering strategy are employed to obtain final clean particles trajectories. This paper models particle trajectories as feature $F = L, D, M$ through chaotic invariants(the largest Lyspunov exponent L and the correlation dimension D , and the mean M of the x and y coordinates).

Modeling Gaussian mixture models (GMMs) based on features, the probability density function of normal is:

$$P(\Gamma|\Phi) = \sum_K \omega_k N(\Gamma; \mu_k, \sigma_k) \quad (2.5)$$

[9] tries to track the single object trajectories in multiple cameras and proposes an invariant distance measures using spectral analysis of graphs to identify the anomalies. First of all, in the single camera, objects are detected by a process based on background model called Statistical And Knowledge-Based Object deTector (SAKBOT). This process groups the difference between the current image and the background model by thresholding and overall coherence. Appearance Driven tracking with Occlusion Classification (Ad Hoc) is the tracking algorithm used to track the objects detected above. Multiple camera processing is achieved by Homography and Epipolar-based COnsistent Labeling. After obtaining trajectories of objects, this paper converts trajectories to graph and then detect anomalies by comparing new trajectory with reference trajectories from spectral clustering.

From above we can see that trajectory-based approaches are hard to tackle video anomaly detection problem if there are occlusions and crowded scenes with complex background. Moreover, procedures using trajectories to detect anomalies are complex with low efficiency, which are not suitable to real-time speed applications.

2.1.2 Dynamic Textures

Within the crowded scenes, the appearance and dynamics are of the same importance for anomaly detection. As a result, [4] proposes a mixture of DTs to model multiple components of different appearance and dynamics, as shown in Figure 2.4. The probability of a sequence $\mathbf{y}_i^\tau = [y_1 \dots y_\tau]^T$ of K dynamic textures is

Figure 2.4: Mixture of DTs in [4]

$p(\mathbf{y}_i^\tau) = \sum_{i=1}^K \pi_i p(\mathbf{y}_i^\tau | z = i)$. The MDT is:

$$\begin{aligned} x_{t+1} &= A_z x_t + v_t \\ y_t &= C_z x_t + w_t \end{aligned} \tag{2.6}$$

where x_t refers to hidden state variable, y_t refers to observed video measurement, A_z represents transition matrices and C_z represents observation matrices. $x_1 \sim \mathcal{N}(\mu_z, S_z)$, $v_t \sim \mathcal{N}(0, Q_z)$, $w_t \sim \mathcal{N}(0, R_z)$. Maximum likelihood and expectation-maximization algorithm are employed to learn parameters. Temporal anomaly in location l is estimated according to the patch centered at l :

$$A_{temporal}(l) = -\log(p_{X|Y}(\mathbf{x}_1^\tau | \mathbf{y}_1^\tau); \theta_l) \tag{2.7}$$

The spatial abnormal detection in paper [4] is based on saliency detection:

$$S(l) = \sum_{c=0}^1 p_{C(l)}(c) KL[p_{\mathbf{Y}|C(l)}(\mathbf{y}|c) || p_{\mathbf{Y}}(y)] \quad (2.8)$$

The overall abnormal detection is the sum of temporal and spatial abnormal maps:

$$A_{total}(l) = A_{temporal}(l) + S(l) \quad (2.9)$$

To extend the work of [4], [5] employs a hierarchical mixture of dynamic textures $\{\{\mathbf{M}_i^1\}_{i=1}^{n_1}, \dots, R^L\}$ learned from the hierarchy of support windows $\{\{R_i^1\}_{i=1}^{n_1}, \dots, R^L\}$ to achieve multiscale anomaly maps.

The method in [12] uses spatiotemporal context and probability model to detect and localize anomalies. They learn local patterns using Sparse Semi-nonnegative Matrix Factorization(SSMF) and these local patterns are constructed to local features for anomaly detection. In detail, they concatenate the grey level of small regions into a vector $a \in \mathbb{R}^d$ ($d = r_x \times r_y \times r_t$) and centralize and normalize it to make it satisfies:

$$\sum_{i=1}^d a_i = 0, \sum_{i=1}^d a_i^2 = 1 \quad (2.10)$$

These vectors are used to learn a set of basis(dictionary) $\mathbf{B} \in \mathbb{R}^{d \times k}$ and non-negative coefficients $\mathbf{c} \in \mathbb{R}_+^k$ by Sparse Semi-nonnegative Matrix Factorization, k is the number of basis:

$$\min_{\mathbf{B}, \mathbf{X}} O = \|\mathbf{A} - \mathbf{B}\mathbf{C}\|_F^2 + \lambda \sum_{i=1}^n |\mathbf{c}_i|, s.t., c_{i,j} \geq 0 \quad (2.11)$$

where $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{d \times n}$ is the few frames from initialization. And it can be rewrite as:

$$O = tr(\mathbf{A}\mathbf{A}') - 2tr(\mathbf{A}'\mathbf{B}\mathbf{C}) + tr(\mathbf{C}'\mathbf{B}'\mathbf{B}\mathbf{C}) + \lambda \sum_{ij} c_{ij} \quad (2.12)$$

Eq. 2.12 can be optimized iteratively. Local features are obtained by proposed method Histogram of Nonnegative Coefficients (HNC) :

$$\mathbf{h}(R) = \sum_{x_i, y_i, t_i \in R} \mathbf{c}_i \quad (2.13)$$

The detection model is constructed below:

$$P(C) = \prod_{l=1}^n p(\mathbf{r}_l | R_0, R_l) = \prod_{l=1}^n \sum_{i=1}^k p(R_0 = \mathbf{b}_i) p(\mathbf{r}_l | \mathbf{b}_i, R_l) \quad (2.14)$$

2.1.3 Sparse Reconstruction

Alternatively, sparse reconstruction is used in [2, 3]. They construct sparse codebook using normal events from training data and detect outlier based on the reconstruction error.

In [2], the authors extract and concatenate video features by MHOF (Multi-scale Histogram of Optical Flow), as shown in Figure 2.5. The two scale refer to 8 directions with motion magnitude smaller of bigger than the threshold. The basic

Figure 2.5: Multi-scale HOF in [2]

units of MHOF are extracted by spatiotemporal bricks or image patches. Using the features of normal data, the initial feature pool $\mathbf{B}$ can be constructed. To obtain the optimal feature pool $\mathbf{B}'$, the authors employ reconstruction:

$$\min_{\mathbf{X}} : \frac{1}{2} \|\mathbf{B} - \mathbf{B}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_{2,1} \quad (2.15)$$

To keep the sparsity, they use $l_{2,1}$ norm for $\mathbf{X}$. Nesterov's method is employed to solve this formulation.

Considering the deformation and unpredicted situation in the testing the dictionary learned above converts to $\Phi = [\mathbf{B}', I_{m \times m}]$, and weight matrix $\mathbf{W} = \text{diag}[w_1, w_2, \dots, w_n, 1, \dots, 1]$ is used to represent the prior that the cost to use a basis appears frequently in the training dataset should be lower. In a sum, the SRC (Sparsity Reconstruction Cost) predicts the abnormal data when SRC is higher than a threshold:

$$S_w = \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{x}^*\|_2^2 + \lambda_1 \|\mathbf{W} \mathbf{x}^*\|_1 \quad (2.16)$$

where

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{x}\|_2^2 + \lambda_1 \|\mathbf{W} \mathbf{x}\|_1 \quad (2.17)$$

To improve the efficiency of [2], [3] changes the Eq. 2.17 from finding s basis combination to find the most suitable combination from a set of possible combination, as shown in Figure 2.6. In detail, the small reconstruction error t is computed by:

$$\begin{aligned} t \triangleq \min_{\mathbf{S}, \gamma, \beta} \sum_{j=1}^n \sum_{i=1}^K \gamma_j^i \|\mathbf{x}_j - \mathbf{S}_i \beta_j^i\|_2^2 \\ s.t. \sum_{i=1}^K \gamma_j^i = 1, \gamma_j^i = 0, 1 \end{aligned} \quad (2.18)$$

where γ_j^i is the indicator for the usage of i^{th} combination $\mathbf{S}_i$.

2.1.4 Deep Learning Based Methods

To summarize, the methods we discussed above have following shortages:

1. They employ handcraft features which cannot capture abstract and meaningful representations.
2. They cannot perform end-to-end learning. They need to spend time on separated steps: feature extraction and model buildings.
3. When they perform anomaly localization, they need to divide images to image patches.
4. For different scenarios, they need to build different models.

Figure 2.6: The possible combination of basis in [3]

Deep learning methods are employed in video anomaly detection [16, 17]. [16] employs a two stream autoencoder to learn the appearance and motion features of image patches. After extracting high level features, they use a one-class classifiers to detect the anomalies.

However, due to small datasets available, many works divide images into patches. They usually use deep learning networks to extract features and use one-class classifiers such as one-class SVM and one-class Gaussian classifiers to find the anomalies. Our work also falls into the category of deep learning which captures high-level features, but we employ end-to-end learning, and our input is video segments rather than frames or patches.

Figure 2.7: Conditional GAN

2.2 Generative Adversarial Network

Generative adversarial network is introduced by [7] to generate images without Markov chains or unrolled approximate inference. This algorithm is relevant to a minimax problem of game theory. In the training phase of GAN, the discriminator is learning to distinguish real data from fake data generated by the generator, while the generator is learning to fool the discriminator by minimizing the probability of the discriminator distinguishing correctly. Considering the generator, GAN is a generative approach used for sample generation. Considering the discriminator, GAN is a discriminative approach that can be applied to classification or feature extraction. Since the pretty generation ability, GAN is popularly used in recent years. To control the attributes of the different generation, conditional GAN is proposed by [18] to generate images according to class label, as shown in Figure 2.7. They perform condition on to both the generator and discriminator. In [19], it employs conditional GAN to achieve face generation. They can perform face generation with specific attributes by varying the conditonal information.

Figure 2.8: The structure of multi-scale video prediction in [10]

Besides image generation, GAN is also applied to other generation applications such as video prediction [20], image super-resolution [21], semantic segmentation [22], and image translation [23]. For example, [20] predicts video frames by integrating GAN into a multi-scale architecture, the structure of multi-scale video prediction as shown in Figure 2.8. Condition on the previous frames, it achieves frames prediction effectively. SRGAN is proposed in [21] to achieve image super-resolution via two perceptual loss: content loss and adversarial loss. It can infer photo-realistic natural images for $4\times$ upscaling factors. However, the instability of GAN training remains a problem. In [24], it proposes DCGAN, which focuses on addressing this problem through architecture design. Moreover, [25] theoretically discusses the instability of GAN, indicating the loss function of GAN is the main reason.

3D-GAN has been used in [26]. The authors employ a 3D generative model of GAN to generate 3D CAD objects. In our work, we design a new 3D-GAN framework and use discriminative model for video temporal anomaly detection.

2.3 Autoencoder

Autoencoder is an unsupervised learning method originally employed to reduce dimensionality. [27] proposes a deep autoencoder initialized by Restricted Boltz-

Figure 2.9: The structure of autoencoder in [27]

mann machines (RBMs) that uses gradient descent for optimization and shows better performance than Principal Component Analysis (PCA), as shown in Figure 2.9. To extract robust features, other variants of autoencoder are proposed such as sparse autoencoder [28], denoising autoencoder [29] and contractive autoencoder [30]. Sparse autoencoder employs the sparsity penalty to force the representation (output of middle hidden layer) to be sparse. Denoising autoencoder tries to reconstruct the original input from a corrupted version (add Gaussian noise for example). Contractive autoencoder applies an analytic contractive penalty to the loss function.

Convolutional autoencoder has been presented in [31], where the authors consider the 2D images as input and construct stacked convolutional autoencoders for initializing Convolutional Neural Networks (CNNs). Autoencoder has been used to 3D shape matching and retrieval [32, 33]. [32] uses autoencoder to learn 3D

shape features by projecting 3D shapes to 2D space. [33] describes 3D shape by a multiscale shape distribution and employs autoencoder to learn features from this distribution.

[1] applies 2D convolutional autoencoder to detect temporal regularity of videos. Extending their work, we propose a 3D fully convolutional autoencoder specific to spatiotemporal anomaly detection. We directly take a 3D video segment as input and learn representations from it.

Chapter 3

Temporal Anomaly Detection using 3D-GAN

3.1 Introduction

Temporal anomalies are unusual segments of videos which are essential in video applications such as action detection [34], video summarization [35] and video anomaly detection [2]. It is a challenging task since the reasons we mentioned in Section 1.2. To address those challenges, anomaly detection is often regarded as the one-class classification problem, treating the video anomalies as outliers [1–5,16,17,36]. They first employ normal video data to train a model or dictionary, then use trained model to detect the outliers such as high reconstruction cost samples. These methods subtly avoid using abnormal data in the training phase. However, most of them perform inefficiently. They spend much computational and memory costs to extract features of videos and use separate classifiers or procedures to detect anomalies.

To enhance the efficiency of video temporal anomaly detection, in this work we propose an end-to-end approach. Recently, generative adversarial network (GAN) [7] has been proposed to integrate discriminative framework with generative modeling. This algorithm is relevant to a minimax problem of game theory. In the training phase of GAN, the discriminator is learning to distinguish real

Figure 3.1: Top: training phase. Normal video segments are taken as the input of the 3D-GAN. Bottom: testing phase.

data from fake data generated by the generator, while the generator is learning to fool the discriminator by minimizing the probability of the discriminator distinguishing correctly. Considering the generator, GAN is a generative approach used for sample generation. Considering the discriminator, GAN is a discriminative approach that can be applied to classification or feature extraction. In [36], authors employ generative modeling of GAN to generate images and optical flows. They implement abnormal event detection by detecting the reconstruction error. Conversely, we propose to apply the discriminative approach of GAN to achieve video temporal anomaly detection directly. The benefits of our method are 1) we achieve the temporal anomaly detection with only one 3D-GAN framework by leveraging its discriminative capability ([36] used two GAN frameworks and one CNN); 2) to capture the motion information, we integrate 3D convolution with GAN framework. Our method is end-to-end with little-to-no supervision.

We present a video temporal anomalies detection approach based on the dis-

criminator of 3D convolutional generative adversarial networks (3D-GANs). As shown in Fig. 3.1, we formulate the video temporal anomaly detection as *fake data* detection using the trained discriminator of GAN. Since the objective of the generator is generating fake data to fool the discriminator, the generator must learn the distribution of real data and try to generate data similar to real data. We formulate normal videos as *real data* and let the generator and the discriminator to learn the distribution of normal videos. When the discriminator learns the distribution of normal videos, abnormal videos, whose distribution is different from *real data*, will be treated as *fake data* and assigned lower scores. Considering the importance of motion information of videos and the difficulty of extracting motion information by the traditional GAN, we design a 3D-GAN. The convolutional layers of 3D-GANs are all 3D convolution, which is suitable to capture motion information inside the videos. Experiments on video anomaly detection datasets show the effectiveness and efficiency of our method.

Figure 3.2: Top: structure of Discriminator. The input of Discriminator is $c \times 8 \times 64 \times 64$. c is the number of channels. Bottom: structure of Generator.

3.2 Proposed Method

3.2.1 Video Temporal Anomaly Detection

To detect temporal anomalies in videos, we adopt generative adversarial frameworks [7] and integrate 3D convolutional layers with them. The original loss function of GAN is:

$$\begin{aligned} \min_G \max_D V(D, G) = & E_{x \sim p_{data}} [\log(D(x))] \\ & + E_{z \sim p(z)} [\log(1 - D(G(z)))] \end{aligned} \quad (3.1)$$

where p_{data} is the distribution of real data, and $p(z)$ is the noise distribution. Discriminator D maximizes this loss to differentiate real data from generated data. Generator G minimizes this value to learn the distribution of real data and generate fake data to fool D .

In theory, there is one Nash equilibrium in the end: G recover the distribution of normal videos and D equal to 0.5 everywhere [7]. However, previous work shows that the Nash equilibrium is hard to approach and GAN is hard to train because of some instability [25]. Therefore, ensuring the generated data distribution to recover abnormal data distribution is difficult when GAN approaches normal video distribution. To address this, we formulate the testing procedure as outlier detection, which just focuses on normal data.

In our formulation, we treat normal video data as real data. Thus, during the training phase, the loss function becomes:

$$\min_D L(D) = -E_{x \sim p_{normal}} [\log D(x)] - E_{z \sim p(z)} [\log(1 - D(G(z)))] \quad (3.2)$$

$$\min_G L(G) = E_{z \sim p(z)} [\log(1 - D(G(z)))] \quad (3.3)$$

where p_{normal} is the distribution of normal videos. Inspired by [24, 25], we use

another version of Eq. 3.3:

$$\min_G L(G) = E_{z \sim p(z)} [-\log(D(G(z)))] \quad (3.4)$$

which is proven to suffer less from vanishing gradients [25].

Since labels for normal data and generated data are binary, Eq. 3.3 and Eq. 3.4 can be the forms of expectation of binary cross entropy:

$$\begin{aligned} \min_D L(D) = & E_{x \sim p_{normal}} [BCE(D(x), 1)] \\ & + E_{z \sim p(z)} [BCE(D(G(z)), 0)] \end{aligned} \quad (3.5)$$

$$\min_G L(G) = E_{z \sim p(z)} [BCE(D(G(z)), 1)] \quad (3.6)$$

where

$$BCE(Y, \hat{Y}) = -(\hat{Y} \log(Y) + (1 - \hat{Y}) \log(1 - Y)) \quad (3.7)$$

In our work, x is video segment, and z is drawn from the Gaussian distribution.

We use the output of Discriminator D as our confidential scores since D will assign higher scores to normal data:

$$S(x) = D(x) \quad (3.8)$$

3.2.2 Proposed Network

To avoid the instability in training, we follow some settings in [24] and adopt 3D convolution to extract motion information.

To learn spatiotemporal representations, 3D convolutional networks are investigated by [37]. Typical 2D convolution uses a 3-dimensional kernel to convolve

Figure 3.3: 3D convolution procedure: 3D convolution considers temporal dimension. (The fourth dimension channel did not show in this figure.)

3-dimensional input feature maps. The output of 2D convolution is a 2-dimensional feature map. Temporal information will collapse quickly if a video segment is taken as a 3-dimensional input block which is convolved with 3-dimensional kernels for a 2D convolutional neural network [37]. Different from 2D convolution, 3D convolution has one more dimension: temporal dimension. A kernel of a 3D convolutional layer is 4-dimensional, and the feature maps output from a 3D convolutional layer are 3-dimensional, as shown in Fig. 3.3. 3D convolution is performed spatiotemporally and is more suitable for spatiotemporal feature learning compared with 2D convolution [37].

We stack six 3D convolutional layers in the discriminator and six 3D deconvolutional layers in the generator. On top of every convolutional/deconvolutional layer (except for the final layers) is batch normalization and leakyReLU/ReLU layers (leakyReLU layer in the discriminator and ReLU layer in the generator). There is no pooling layer and fully connected layers. The height and width of all convolutional/deconvolutional kernels is 4 with stride 2. Moreover, to avoid temporal information decreasing fast, the time dimension of kernels in the first convolutional layers of the discriminator is set to 3 with stride 1. The time dimension of kernels in other convolutional layers are set to 4 with stride 2. The

Figure 3.4: Qualitative results of temporal anomaly detection. The X-axis: confidential score. Y-axis: the frame number. Blue line: confidential scores. Red shaded regions: ground-truth anomalies.

architecture of the generator and the discriminator are similar, but reversed. The structures of Discriminator and Generator is shown in Fig. 3.2.

Our network’s input is a video segment rather than an image with RGB channels. We use color images which contain three channel. A video segment is constructed by stacking L successive frames of resolution 64×64 in a video. We set $L = 8$ for our experiments. The dimensions of the input of our network are $3 \times 8 \times 64 \times 64$, where the first dimension is the channel number. In spatial dimension, we do not use any data augmentation techniques such as mirror, cropping, and jittering. We also do not augment the training data in the temporal dimension. Since the structure of our 3D-GAN is new, we train the network from scratch.

3.3 Experiments

We implement our network on torch. All experiments are conducted on NVIDIA Titan Xp GPUs. We initialized our network weights from a Gaussian distribution, and we used Adam to optimize our network. The momentum is set to 0.5 following [24]. The batch size is set to 256. The initial learning rate of D is 0.0001, and that

of G is 0.002. The latent vector z is set to 300-dimension. Our frame classification time (on one Titan Xp GPU) is 0.00626(second/frame).

Datasets. We perform experiments on two video anomaly detection datasets: CUHK Avenue dataset [3] and UMN dataset [38]. The Avenue dataset consists of 12 training video samples which only contain normal events and 16 testing video samples which contain both normal and abnormal video events. Each frame in this dataset has a resolution of 640×320 pixels. Running, throwing and the wrong direction are abnormal behaviors in Avenue dataset. The UMN dataset contains three different scenes. People are walking in the majority frames, which are treated as normal. Running away is the anomalies in UMN dataset. The abnormal behaviors in those two datasets are not very local and suitable for video temporal anomalies detection.

Evaluation Metrics. We use AUC (Area Under Curve) and EER (Equal Error Rate) as the measurement to evaluate our methods. AUC is the area under ROC (Receiver Operating Characteristic) curve. The x-axis of ROC curve refers to FPR (false positive rate), and the y-axis of ROC curve refers to TPR (true positive rate). For different thresholds, experiments will obtain different TPR and FPR. Higher AUC indicates the better performance. EER is the point $FPR = FNR$, the smaller the better.

We first used Avenue dataset to train our 3D-GAN. Since the number of frames in UMN dataset is small, we just extract 800 frames from every scene of UMN dataset and finetune 3D-GAN based on those frames.

To better evaluate our method, we built a 2D-GAN which has a similar architecture with our 3D-GAN except for the input. The input of 2D-GAN is one frame with RGB channel: $3 \times 64 \times 64$.

3.3.1 Results of Avenue Dataset

As shown in Table. 3.1, we compare our results on Avenue dataset with that of state-of-the-art [1] and 2D-GAN. Our method achieves higher AUC ($\sim 9.5\%$) and lower EER (1%) than the state-of-the-art. The AUC of 2D-GAN is higher than that of the state-of-the-art, which confirms the discriminative ability of GAN framework.

Table 3.1: Temporal anomaly detection performance of Avenue

Method	AUC (%)	EER (%)
Hasan <i>et al.</i> [1]	70.2	25.1
2D-GAN	76.95	35.0
Ours	79.68	24.1

Our method is also better than 2D-GAN, indicating the effectiveness of 3D convolution. Fig. 3.7(d) illustrates the ROC curve of our approach and 2D-GAN: FPR of our method in the range of $0.3 \sim 4.4$ corresponds to a higher TPR. The points of our ROC curve are closer to the upper left corner than theirs, indicating our method can achieve higher TPR when FPR is small. This also verifies the stability of 3D convolution in videos. Fig. 3.4 shows qualitative results of our approach. The abnormal behaviors such as running away, throwing and wrong direction obtain lower scores than normal behaviors. Moreover, we visualize the feature maps (512 dimensions) of the last layer of Discriminator using tsne [?], as shown in Figure 3.5. The blue points in Figure 3.5 are ground-truth normal data and the yellow points are abnormal data. We can see that features of videos are divided to some continuous line (every point in this figure suggests one sample). The abnormal samples also form some lines which are different from lines of normal samples. It illustrates that the features of abnormal and normal samples are grouped into different lines in feature space.

Figure 3.5: Visualizing feature maps (last layer, 512 dimensions) of Avenue dataset using t-sne.

3.3.2 Results of UMN dataset.

Table 3.2: Temporal anomaly detection performance of UMN

Method	Scene 1 (%)		Scene 2 (%)		Scene 3 (%)	
	AUC	EER	AUC	EER	AUC	EER
2D-GAN	70.03	29.5	82.21	25.5	67.78	42.1
Ours	97.83	6.1	90.83	13.7	96.83	6.8

Table. 3.2 compares our method and 2G-GAN in different scenes of UMN. Our 3D-GAN obtains a higher AUC and a smaller EER in all of three scenes. The AUC of our method exceeds the AUC of their method in scene 1 and scene 2 by more than 20%, and the EER of our method is more than 20% lower than the EER of 2D-GAN in those two scenes. As can be seen in Fig. 3.7, our network performs

Figure 3.6: Visualizing feature maps (last layer, 512 dimensions) of UMN dataset using t-sne.

Figure 3.7: Comparison of our method and 2D-GAN via ROC curve.

more stable than 2D-GAN. In these three scenes, the upper left points of ROC curves of our method are all closer to $(1, 0)$, showing the better discrimination of our methods. The majority of FPR in our approach corresponds to a higher TPR than 2D-GAN. Fig. 3.4 also illustrates the qualitative results of our method on UMN dataset. The visualization of feature maps of the last layer is shown in Figure 3.6. Basically, the abnormal data can be distinguished from normal data.

Chapter 4

Spatiotemporal Anomaly Detection using 3D-FCAE

4.1 Introduction

Detecting spatiotemporal anomalies in videos is a challenging problem for two reasons. First, visual anomalies are generally not well defined. Second, as abnormal patterns are uncommon and hence seldom appear in videos, it is challenging to use supervised approaches due to the lack of sufficient training samples.

Most existing works try to address spatiotemporal anomaly detection by first learning the normal patterns from normal training videos and then detecting the anomalies as deviations from the normal patterns [2–6, 8–11, 39–41]. Sparse coding based methods [2, 3] first construct a codebook using normal training examples and then detect outliers based on reconstruction errors. In [4, 5], dynamic textures are proposed to model normal patterns, and abnormal patterns are identified via discriminative saliency. Mehran *et al.* [11] propose a social force model to estimate interactive forces which are represented by a bag of words for anomalous event classification. In addition, deep learning methods [16, 17] have also been employed for video anomaly detection, which represent image regions using features learned by deep neural networks and apply one-class classifier to detect anomalies. Recently, Hasan *et al.* [1] propose to learn temporal anomalies in videos using a 2D

convolutional autoencoder. It is able to learn both the feature representation and the temporal visual characteristics of normal patterns in an end-to-end manner, and is computationally more efficient than sparse coding on video datasets [1]. Although this approach can temporally locate anomalies implied by reconstruction errors of video clips, it is not as effective in locating irregularities both spatially and temporally (Fig. 4.1). This can be attributed to the loss of temporal information, especially local motion, due to the use of 2D convolution in the autoencoder [37]. However, accurate spatiotemporal localization of anomalies is crucial in many real-world applications such as anomaly detection in crowds from surveillance videos [2].

To achieve more accurate spatiotemporal localization of abnormal visual patterns, we develop a 3D fully convolutional autoencoder (3D-FCAE) that is trainable in an end-to-end manner. Given normal videos as training data, the proposed 3D-FCAE performs unsupervised learning of the spatiotemporal representation of normal patterns and reconstruct them with low errors. Consequently, spatiotemporal anomalies can be localized as regions with high reconstruction errors (*e.g.*, yellow regions in column 5, Fig. 4.1). Comparing with the work of Hasan *et al.* [1], our 3D-FCAE can more accurately reconstruct and locate normal spatiotemporal patterns, while discriminating abnormal patterns, thanks to its 3D convolutional layers. Reconstruction error maps in Fig. 4.1 demonstrate the accuracy of our reconstruction in comparison with [1], where prominent yellow regions indicate a high reconstruction error. As can be seen, our 3D-FCAE results in apparently poorer (*i.e.*, blurrier) reconstructions of abnormal objects and motions, while Hasan *et al.*'s method [1] does not demonstrate much discrimination.

In summary, our proposed 3D-FCAE has three benefits. First, it can accurately locate spatiotemporal anomalies in videos with weak supervision (*i.e.*, it only requires normal videos as training data, while regular patterns and reconstructions are learned in an unsupervised manner). Second, the proposed 3D-FCAE can be implemented in an end-to-end manner and achieves promising performance for

Figure 4.1: Spatiotemporal anomaly detection. Blue bounding boxes denote the abnormal objects in the original frames (1^{st} column). Best viewed in color.

both temporal anomaly detection and spatiotemporal anomaly detection thanks to the use of 3D convolution. Third, the model shows good generalizability across multiple video datasets.

4.2 Propose Method

In this section, we introduce our 3D fully convolutional autoencoder (3D-FCAE). We first describe the basic concepts of 3D-FCAE. Second, we present the framework of the whole system. Then, we detail the descriptions of performing anomaly detection using 3D-FCAE. After that, we explore the good practices in training 3D-FCAE. Finally, we evaluate the effectiveness of our approach on two spatiotemporal localization of anomalies applications: video anomaly detection and running detection. We perform local anomaly detection and global anomaly detection for video anomaly detection, and local anomaly detection for running detection. We

provide both quantitative and qualitative results and compare our approach with the baseline method [1] and other state-of-the-art approaches.

Figure 4.2: Encoder’s output is the hidden middle layer which is a high dimensional representation of the input; decoder tries to reconstruct the input of the whole system.

4.3 Fully 3D Convolutional Autoencoder

4.3.1 Autoencoder

The basic autoencoder is composed of two components: encoder and decoder, as shown in Figure 4.2. The encoder is mapping input $x \in [0, 1]^d$ to the hidden middle layer $h \in [0, 1]^{d'}$

$$h(x) = h(W_e x + b_e) \quad (4.1)$$

using function of encoder $h(\cdot)$. The hidden middle layer is the encoder’s output which can be used as the latent representation of the input. The decoder $d(\cdot)$ takes this latent representation as input to obtain the reconstruction

$$r(x) = d(W_d h(x) + b_d) \quad (4.2)$$

Thus, every input can obtain its latent representation and reconstruction. Typically we optimize basic autoencoder by minimizing the average squared reconstruction

tion error:

$$L = \frac{1}{N} \sum_i ||r(x) - x||_2^2 \quad (4.3)$$

(a) 2D convolution

(b) 3D convolution (without channel dimension)

Figure 4.3: (a) 2D convolution procedure: feature maps convolve with a 2D convolution kernel to get one channel feature map. (b) 3D convolution considers temporal dimension.

4.3.2 3D Convolution and Deconvolution

To learn spatiotemporal representations, 3D convolutional networks are introduced by [37]. Typical 2D convolution uses a 3-dimensional kernel to convolve 3-dimensional input feature maps. The output of 2D convolution is a 2-dimensional feature map. Temporal information will collapse quickly if a video clip is taken as a 3-dimensional input block which is convolved with 3-dimensional kernels for a 2D convolutional neural network [37]. Different from 2D convolution, 3D convolution has one more dimension: temporal dimension. A kernel of a 3D convolutional layer is 4-dimensional, and the feature maps output from a 3D convolutional layer are 3-dimensional, as shown in Figure 4.3. 3D convolution is performed spatio-

temporally and is more suitable for spatiotemporal feature learning compared with 2D convolution [37].

3D deconvolution, which is an inverse process of 3D convolution, is used in the decoder of our autoencoder. Deconvolution is also called fractionally strided convolution or transposed convolution. Fractionally strided convolution means interpolating 0s to feature maps and then performing convolution. Transposed convolution refers to convolving the transpose of convolution kernels. They explain deconvolution from two different angles.

4.3.3 Fully 3D Convolutional Autoencoder

To detect video anomalies, we focus on finding outliers of normal spatiotemporal patterns. We propose an end-to-end training technique called 3D fully convolutional autoencoder (3D-FCAE) to learn the normal spatiotemporal patterns with weak supervision. Specifically, our 3D-FCAE aims to employ the normal visual information of video clips to perform video clip reconstruction, as illustrated in 4.4(a). The 3D-FCAE model can be exploited for detecting both global anomalies and local anomalies in videos, as shown in 4.4(b).

Motivated by [42], we apply a fully convolutional framework to our autoencoder. Without fully connected layers, a fully convolutional network reserves location information which is essential for pixel-level detection. The training procedure of 3D-FCAE is unsupervised without using labels. The architecture of our 3D-FCAE is shown in Figure 4.5. Using fully convolutional layers to preserve location information and 3D convolution to capture motion and appearance information, our 3D-FCAE learns essential spatiotemporal information for efficiently reconstructing video clips.

(a) Training phase

(b) Testing phase

Figure 4.4: Training procedure and testing procedure of our 3D-FCAE.

4.4 Learning Fully 3D Convolutional Autoencoder

In the training phase, 3D-FCAE takes video clips as input and outputs reconstructed video clips, as shown in Figure 4.4(a). In general, given the i th video clip V_i , the autoencoder reconstructs the video clip using $R_{\tilde{W}}(H_W(V_i))$. $H_W(\cdot)$ is a stacked 3D fully convolutional network with weight parameters W and $R_{\tilde{W}}(\cdot)$ is a stacked 3D fully deconvolutional network with weight parameters $\tilde{W}$. $H_W(V_i)$ is the encoder utilized to extract the latent representation from V_i . The decoder $R_{\tilde{W}}(H_W(V_i))$ takes the output of the encoder as input to reconstruct V_i . To learn

this autoencoder, we use the Euclidean loss as the objective function:

$$R_{\tilde{W}}^*, H_W^* = \arg \min_{R_{\tilde{W}}, H_W} \frac{1}{2N} \sum_i \|V_i - R_{\tilde{W}}(H_W(V_i))\|_2^2 + \lambda \|R_{\tilde{W}}\|_2^2 + \lambda \|H_W\|_2^2 \quad (4.4)$$

where λ is a hyperparameter to balance the loss and the regularization and N is the number of training video clips. We use stochastic gradient descent (SGD) to learn weight parameters W and $\tilde{W}$.

Different from other visual tasks, video anomaly detection needs proper training over normal videos and bad performance on abnormal videos. It implies that good performance on both training data and testing data is not suitable for anomaly detection. It's important to find a trade-off between good performance on training data and bad performance on specific testing data. To tackle this issue, we investigate a series of good practices in training a well-suited autoencoder for video anomaly detection.

Figure 4.5: Our 3D-FCAE consists of three 3D convolutional layers and three 3D deconvolutional layers.

4.4.1 Network Architectures

Some work has shown that deeper structures can improve the performance of several visual tasks such as object detection and images segmentation [43]. In contrast, autoencoder as an unsupervised learning method is not suitable for using deeper networks, because deeper networks enhance the capacity of the encoder and decoder. Increasing the capacity of encoder and decoder may block autoencoder to learn an adequate representation of the input. In our autoencoder, we choose a relatively shallow network structure, which consists of three 3D convolutional layers in the encoder and three 3D deconvolutional layers in the decoder. we define a network with less than 5 convolutional layers as a shallow network. There are no pooling layers and fully connected layers in our autoencoder. We use strided convolution with stride 2 to allow the network to learn spatial downsampling [24].

We use Tanh as the activation function. For autoencoder, using Tanh as the activation function achieves better performance than using ReLU because of the symmetry property of Tanh. It is justified by experiments in next section. We also employ batch normalization [44] as regularization which turns out to improve the training efficiency.

Considering the property of our 3D-FCAE that the type of training exhibits good reconstruction and other types display poor reconstruction, our autoencoder is trained from scratch without employing any pre-train. Pre-train models may include the distribution of both training data and testing data. It will cause negative effects on the performance of the autoencoder. There is also no any point to train a strong autoencoder which can perform the one-to-one mapping.

Network inputs. As in [37], our autoencoder’s input is a video clip rather than an image with RGB channels. We use grayscale images which only contains one channel. A video clip is constructed by stacking L successive frames of resolution 112 x112 in a video. We set $L = 8$ in our experiments. The dimensions of

the input of our autoencoder are $1 \times 8 \times 112 \times 112$, where the first dimension is the channel number.

In spatial dimension, we do not use any data augmentation techniques such as mirror, cropping, and jittering. We also do not augment the training data in the temporal dimension. In [1], the authors sample video sequences with strides of 2 and 3 to augment the training data in the temporal dimension. We only use the original video sequences for training.

Besides grayscale images, we also considered using optical flow as the input for our network. Optical flow computed between two consecutive frames is used as the motion representation of the two frames. Formally, optical flow consists of x and y displacement vectors for every position in the frame which indicate the horizontal and vertical components of the vector field. We stack the optical flow of L consecutive frames as the input for our autoencoder. We find that the performance is worse than using grayscale images as the input because 3D-FCAE forces the reconstructed optical flow to fit the same distribution which leads to the loss of angle information between two displacement vectors. It is also pointed out in [1] that stacking the optical flow with grayscale video frames does not improve the overall performance.

4.5 Testing Fully 3D Convolutional Autoencoder

The proposed 3D-FCAE enables the detection of two types of anomaly detection: the global detection and local detection. The global detection often applies to scenarios that involve the behaviour of crowds, for example crowded running and fighting. In those situations, detecting individual actions is tough and not critical. The local detection is often applied to individual anomalous behavior detection. In testing phase 4.4(b), reconstructed video clips from 3D-FCAE can be applied to perform global anomaly detection using frame-wise reconstruction errors and local

anomaly detection using reconstructed frames (Algorithm 1).

4.5.1 Global Anomaly Detection

For global detection, we compute regularity scores for every frame following [1].

First, we estimate the reconstruction error e for every pixel of frame t :

$$e(x, y, t) = ||I(x, y, t) - R(H(I(x, y, t)))||_2 \quad (4.5)$$

where $H(\cdot)$ and $R(\cdot)$ represent encoder and decoder respectively, $I(x, y, t)$ is the pixel value at location (x, y) in t -th frame, $e(x, y, t)$ is the reconstruction error of location (x, y) in the t -th frame. Then we can compute a temporal regularity score, $g(t)$, by summing up the pixel-wise reconstruction errors and normalizing it:

$$g(t) = 1 - \frac{e(t) - \min_t e(t)}{\max_t e(t)} \quad (4.6)$$

where

$$e(t) = \sum_{(x,y)} e(x, y, t) \quad (4.7)$$

Following [1], we can use persistence1D algorithm [45] based on temporal regularity score to detect global abnormal events.

4.5.2 Local Anomaly Detection

Due to high reconstruction errors by our autoencoder, abnormal regions in a reconstructed frame become blurry. Therefore, we can treat the problem of local anomaly detection as the problem of detecting the blurry regions in reconstructed frames. To this end, we first compute the magnitude of gradients for each frame and that of its corresponding reconstructed frame. Denote by V_t and R_t the t -th frame in an input video and its reconstructed frame with $t \leq T$, where T is the total number of frames in the input video. We compute two gradient magnitude maps G_t^v and G_t^r for the two frames, respectively. For the gradient magnitude map

G_t^r , we eliminate regions of high gradient magnitudes which usually correspond to clear borders due to good reconstruction by a threshold $T1$. Specifically, the locations in G_t^r are filled with 0s, if their gradient magnitudes are larger than $T1$. Let $G_t^{r'}$ be the gradient magnitude map obtained by thresholding G_t^r . Then, we compute element-wise products of G_t^v and $G_t^{r'}$ to get $G_t = G_t^v \otimes G_t^{r'}$. Finally, we apply the max-path search algorithm with multi-scale extension [46] to $\{G_t\}_{i=1}^T$ to find spatiotemporal paths in the video that have the highest path-wise reconstruction errors. These paths correspond to the spatiotemporal abnormal events. Specifically, max-path search algorithm treats an entire video as a spatiotemporal trellis, and applies dynamic-programming algorithms to efficiently find an optimal path (*i.e.*, a sequence of bounding box locations) that has the highest path score. To find the top-K paths, one just need to remove the current best path and repeat the search. Algorithm 1 summarizes the procedure of our local anomaly detection approach.

Algorithm 1 Anomaly Localization for 3D fully convolutional autoencoder

Input: T frames video clips $\{V_i\}_i^T = 1$, T frames reconstructed video clips $\{R_i\}_i^T = 1$

Output: Local anomaly detection maps $\{S_i\}_i^T = 1$

```

for  $i = 1 \dots T$ 
     $G_i^r = ||\text{gradient}(R_i)||$ 
     $G_i^v = ||\text{gradient}(V_i)||$ 
     $G_i^{r'} = G_i^r$ 
     $G_i^{r'}(G_i^r > T1) = 0$ 
     $G_i = G_i^v \otimes G_i^{r'}$  (element-wise product)
end
 $\{S_i\}_i^T = \text{Max-path}(\{G_i\}_i^T)$  [46]
```

4.6 Experiments

In this section, we give the detailed descriptions of experiments focus on video anomaly detection using our 3D-FCAE. We first introduce datasets and implementation details for video anomaly detection. Then, we set up some explo-

ration experiments to explore a 3D fully convolutional autoencoder specific to video anomaly detection. Finally, we provide both quantitative and qualitative results and compare our approach with the baseline method [1] and other state-of-the-art approaches. To evaluate the cross-dataset generalizability of our model, we carry out running detection using the same 3D-FCAE model as video anomaly detection.

4.6.1 Datasets and Implementation Details

Different from other visual tasks using deep learning methods, datasets of video anomaly detection are relatively small. Considering this issue, we train our models based on multiple datasets: UCSD pedestrian [4] and Avenue [3].

UCSD pedestrian datasets are static camera datasets in crowded scenes. They consist of pedestrians and other vehicles such as cars, bicycles, skateboarders, wheelchairs. UCSD pedestrian datasets have two different scenes: Ped1 and Ped2. Training data provided only contain pedestrians which are defined as normal. Vehicles appeared on screen are defined as the anomalies. UCSDped1 has 34 normal video samples and 36 abnormal video samples for training and testing respectively. Each video sample involves 200 frames images with the resolution of 158×238 pixels. UCSDped2 is composed of 16 training samples and 12 testing samples with the resolution of 240×360 pixels.

The Avenue dataset is a static camera dataset collected in front of a subway station. It consists of 12 training video samples that only contain normal events, and 16 testing video samples containing both normal and abnormal video events. Each frame in this dataset has a resolution of 640×320 pixels.

We implement our 3D-FCAE on modified Caffe [47]. All experiments are conducted on NVIDIA Tesla K80 GPUs. We initialize our network weights using MSRA method as [37], and we use SGD with momentum to optimize our network.

The momentum is set to 0.5. The batch size is set to 64. Weight decay is set to 0.0005. The initial learning rate is 0.00001, and decrease by a factor of 1/10 every 3000 iterations. The learning rate of the encoder is 10 times that of the decoder.

4.6.2 Exploration Study

In this section, we focus on investigating the good practices used in Section 4.4.1. All experiments in this section performed on UCSDped1 dataset. Specifically, we compare (1) deep layers with shallow layers; (2) hidden middle layers with small middle layers; (3) learning rates setting; (4) other settings such as ReLU with Tanh, momentum 0.5 with momentum 0.9. And we perform some experiments to explore abilities of our autoencoder.

With an undercomplete hidden middle layer, autoencoder seems suitable to use deeper networks to compress well for the training data. However, the capacity of encoders and decoders needs to be constrained to an appropriate range to ensure the reconstruction ability. In other words, capacity leads to a one-to-one mapping without any semantics and contains redundant representations. There are two aspects which affects the capacity of encoders and decoders: depth of autoencoder and number of hidden middle layers.

To analyze how the two factors affect the performance, we construct two networks. The first one is a deep network with more than eighteen 3D convolutional layers in the encoder and three 3D deconvolutional layers in the decoder. The second one is a shallow network with three 3D convolutional layers in the encoder and three 3D deconvolutional layers in the decoder. Figure 4.6 shows that the deeper network perform worse than shallow network: the training loss of shallow network is much smaller than the training loss of deep network after 1000 iterations. This may be caused by the limited training data and the vanishing gradient problem in deeper networks.

Figure 4.6: Deep 1 : 9408 units in the hidden middle layer with 18 layers in the encoder. Deep2: 12544 units in the hidden middle layer. Shallow: 12544 units in the hidden middle layer with 3 layers in the encoder.

We also illustrate this finding by some reconstructed frames from the two networks in Figure 4.7. The shallow network obtains the bad reconstruction of the anomalies while deep networks reconstruct all parts of frames to the same extent. In addition, with more units in the hidden middle layer, better reconstruction can be achieved. However, the one-to-one mapping can be caused by clearer reconstruction.

We can see from above experiments that a very deep network is hard to train with limited training data. To ensure this, we perform another experiments on depth 3, 5, 7 of the encoder. Fig. 4.8 shows the training loss of depth 3, 5 and 7 with a batch size of 48 and depth 3 with a batch size of 64. With batch size 48, different depths have similar performance which illustrates that limited training data restrict the performance of the autoencoder. The learning rate can increase when the batch size increases, this can lead to a better local optimum. Constrained by the memory of the GPU used in our experiments, a shallow network (e.g. of

Figure 4.7: Top image: the original frame. Middle row: the reconstructed images from the deep networks with different number of units in middle hidden layers. Bottom image: the reconstructed image from the shallow network. Number under images: number of units in hidden middle layers. The white car is the abnormal objects in this dataset.

depth 3) with a higher batch size can achieve better performance.

The encoder of the autoencoder is essential since the decoder has the ability to reconstruct the input only if the encoder captures enough key information of the input. In our experiment, the encoder comprises bottom layers of our autoencoder. Thus, The gradients of the encoder’s layers tend to be smaller than those of the decoder’s layers. To avoid gradient vanishing, we set the learning rates of encoder’s layers bigger than decoder’s layers in every iteration as [48]. This technique can help accelerate training convergence.

Other results of exploration study are in Figure 4.9. We can see that the training loss is lower when use Tanh rather than ReLU as activation function.

Figure 4.8: Training loss of different depth of the encoder.

To sum up, our exploring study has revealed that:

1. 3D convolutional autoencoder with deeper layers may not help due to the limited data.
2. Resolution of reconstructed frames is proportional to the number of units in the hidden middle layer.
3. Use Tanh as activation function and set momentum to 0.5 is helpful.
4. Set learning rate of the encoder bigger than learning rate of the decoder in every iteration will increase the network's convergence.
5. Our autoencoder is sensitive to motion information.

4.6.3 Evaluation on Video Anomaly Detection

For global abnormal event detection, we focus on the anomalies of temporal. Following [1], we compare our approach with [1] for temporal anomalies detection on

4.6. EXPERIMENTS

Figure 4.9: (a) training loss of networks with different activation function. (b) training loss of networks using different momentum during optimization.

UCSDped dataset and Avenue dataset. [1] and our work both use autoencoders and regularity scores to detect global anomalies. Following [1], we compute the temporal regularity score of every frame and use the persistence1D algorithm [45] to detect the temporal irregularities (*i.e.*, anomalies). Table 4.1 shows that our network achieves comparable performance to the 2D convolutional autoencoder proposed in [1].

Dataset	# Anomalous Event	Correct Detection / False Alarm	
		3D-FCAE	state-of-the-art [1]
UCSD ped1	40	37/2	38/6
UCSD ped2	12	12/0	12/1
CUHK Avenue	47	46/5	45/4

Table 4.1: Global abnormal event detection performance

Figure 4.10 shows regularity scores of frames in UCSD pedestrian and Avenue. The regularity score of those frames where abnormal events occur are relatively low. Overall, the regularity scores of the majority of normal events are higher than those of abnormal events. For a small portion of normal events, they have relatively low regularity scores since they are poorly constructed due to high level of crowdedness.

Figure 4.10: Qualitative results of global anomaly detection. Blue line : regularity scores. High regularity score implies high probability of being normal. Red shaded regions: ground-truth anomalies.

For local abnormal detection, we evaluate our method on UCSDped2 dataset and detect anomaly regions in reconstructed video clips. Following the video anomaly detection setting [2], a frame is detected as a true positive if 40% or more pixels of true anomalies are detected in this frame; if a frame without true anomalies is detected as containing anomalies, this frame is treated as a false positive. We compute Equal Error Rate (EER) and Area Under Curve (AUC) according to true positive rate and false positive rate to quantitatively analyse the performance. Table 4.2 shows the EER and AUC comparison of our method and state-of-the-art methods. As can be seen, our method outperforms 2D convolutional autoencoder [1] significantly on local anomaly detection. On UCSDped2, the EER of our approach is 19.4% lower than that of [1]. And the AUC of our network is 21.9% higher than that of [1] on UCSDped2. These results demonstrate the advantage of 3D convolution over 2D convolution for extracting spatiotemporal information. Our

Figure 4.11: Examples of local anomaly detection results. First line: original frames. Second line: reconstruction frames. Third line: results of detection based on original frames and reconstruction frames. Red shaded region: detected abnormal parts.

method also outperforms the other state-of-the-art methods on UCSDped2, which validates the effectiveness of our approach.

Algorithm	Ped2(%)	
	EER	AUC
MPPCA [49]	71	77.4
Social force [50]	80	62.3
Social force + MPPCA [4]	72	71.0
Mixture dynamic texture [4]	54	84.8
Hasan <i>et al.</i> [1]	30	67.3
Sabokrou [17]	19	88.4
Xiao [12]	17	88.2
Ours	10.6	89.2

Table 4.2: Pixel-level EER and AUC of abnormal event localization for UCSD Ped2 dataset

Figure 4.11 shows qualitative results of local anomaly detection on UCSDped2. We can observe that abnormal objects such as cars and bicycles, which move much faster than normal objects in this dataset (*i.e.*, pedestrians), appear blurry in the

reconstructed frames, due to high reconstruction errors by our autoencoder.

4.6.4 Evaluation on Running Detection

For running detection, we use the running sequences in the NTU-run dataset [46]. There are 15 running sequences in crowded scenes with many partial and full occlusions. The resolution of each frame in this dataset is 640×320 .

Figure 4.12: NTU-run examples. The red rectangular boxes in the original frames (top row) show the ground truth of running person, which are poorly reconstructed (*i.e.*, blurry) in the reconstructed frames (bottom row).

As the majority of training data in UCSD pedestrian and Avenue datasets only contain walking pedestrians. Therefore, we directly test the model trained earlier for abnormal event detection using these two datasets on the NTU-run dataset. As can be seen in Figure 4.12, our model can reliably reconstruct the background and foreground scenes of NTU-run dataset from the normal training videos in UCSDped2 and Avenue datasets. This experiment demonstrates the cross-dataset generalizability of our model.

In the testing phase, image regions containing running persons are poorly reconstructed and therefore become blurry, as they deviate from the normal training data.

Figure 4.12 shows qualitative results, where the blurry running person can be easily distinguished from the rest of the scene.

4.6. EXPERIMENTS

Algorithm	NTU-RUN(%)		
	EER	AUC	PA
Hasan <i>et al.</i> [1]	46	52.36	-
Video event detection [46]	-	-	29.14
Ours	25.6	74.51	32.17

Table 4.3: Pixel-level EER and AUC and Path-based localization (PA) of running detection on the NTU-run dataset

Figure 4.13: Running detection: ROC curves of Hasan *et al.* and our method.

For evaluation, in addition to EER and AUC, we also compute path localization accuracy (PA), which is the overlapping volume of two paths divided by their union volume following [46] to facilitate comparison with [46]. The ROC curves of our approach and [1] of running detection are also shown in Figure 4.13. Table 4.3 shows quantitative results of running detection. Our approach has a higher PA than [46] since our approach learns features by the proposed 3D-FCAE while [46] uses hand-crafted features. Compared with [1], our approach again achieves a lower EER and a higher AUC, which demonstrates the advantage of 3D convolution over 2D convolution for detecting local anomalies.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, a systematic study on video anomaly detection has been presented, which is summarized in the following.

Firstly, we investigate the properties of video anomaly detection. The critical point of video anomaly detection is the deviation. We can find the anomaly according to the normal pattern and the deviation. However, there are two main challenges: objective definition problem and limited datasets. These two challenges lead to some limitation when we solve the tasks. Thus, we conclude that the better formulation for video anomaly detection is unsupervised learning method in an end-to-end manner.

Secondly, we present a novel formulation to achieve video temporal anomaly detection. We formulate this task as a fake data detection in GAN. We just use normal video data to train GAN and employ the discriminative framework of GAN to detect temporal anomalies. Since traditional GAN framework is designed for image generation, we integrate 3D convolution with GAN and develop a new framework to approach video temporal anomaly detection. Comparisons of our method against 2D-GAN and the state-of-the-art show the effectiveness and stability of our approach.

Then, we develop a 3D fully convolutional autoencoder for reconstruction-based spatiotemporal anomaly detection in videos, which can achieve both temporal anomaly detection and spatiotemporal anomaly detection in a single model. Our experiments on benchmark datasets show that the proposed method can generalize well to perform cross-dataset anomaly detection. The comparisons with existing methods validate the effectiveness of the proposed 3D-FCAE.

To summarize, in this thesis, we address the main challenges of video anomaly detection through some novel formulations. Efficient and effective methods such as GAN and autoencoder are employed to support these formulations. These formulations are all unsupervised and end-to-end learning, which can be easily implemented to research and industry.

5.2 Future Work

There are two research aspects we can focus on, the first is online video anomaly detection, the second is a new formulation of video anomaly detection: PU (positive-unlabeled) learning.

Figure 5.1: Online video anomaly detection.

5.2.1 On-line Video Anomaly Detection

Even though our work achieves real-time in video anomaly detection, online detection is still necessary for video surveillance. To achieve online video anomaly detection, we will formulate anomaly detection as a prediction cost problem. In the training phase, we will use normal data to train a video predictor. This predictor can predict the next frame according to previous (4-8) frames. Since only normal data appears in the training, we assume that this predictor will not predict abnormal frames well. Blur or noise will be shown in the abnormal predicted frames, as shown in Figure 5.1. Then we can employ some online detection methods to detect the blur or noise in the predicted frames. These procedures are all online.

5.2.2 Video Anomaly Detection using PU Learning

Figure 5.2: Training procedures of PU learning.

In our work, we only employ the normal videos in the training. The networks we construct learn the normal patterns, while they are insensitive to the abnormal patterns. However, this formulation needs carefully design, since the boundary between normal patterns and abnormal patterns share some characteristics between normal patterns and unknown patterns. Thus, we will employ positive-unlabeled (PU) learning methods in the future. Unlabelled data in PU learning methods contain negative (abnormal) samples without labeling. We can make use of those abnormal samples in the unlabelled data during training phase. This could lead to

a better boundary between normal patterns and abnormal patterns. In detail, we can first employ our FCAE to predict the abnormal data in unlabelled data, then we can use predicted labels to train a two-class classifier. The training procedures are shown in Figure 5.2. When we obtain the two-class classifier, we can use it to detect the abnormal data. Although we use abnormal data as a part of unlabelled data in the training, we still employ no labels in whole training procedures.

List of Publications

- [1] Mengjia Yan, Xudong Jiang, Junsong Yuan "3D Convolutional Generative Adversarial Networks for Detecting Temporal Irregularities in Videos", accepted by *ICPR*, 2018.
- [2] Mengjia Yan, Jingjing Meng, Chunluan Zhou, Zhigang Tu, Tap-Peng Tan, Junsong Yuan "Detecting Spatiotemporal Irregularities in Videos via a 3D Fully Convolutional Autoencoder", submitted to *IEEE Transactions letter on Circuits and Systems for Video Technology*, 2018.
- [3] Tu, Zhigang and Guo, Zuwei and Xie, Wei and Yan, Mengjia and Veltkamp, Remco C and Li, Baoxin and Yuan, Junsong "Fusing disparate object signatures for salient object detection in video", *Pattern Recognition*, 2018.

Bibliography

- [1] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016, pp. 733–742.
- [2] Y. Cong, J. Yuan, and J. Liu, “Sparse reconstruction cost for abnormal event detection,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3449–3456.
- [3] C. Lu, J. Shi, and J. Jia, “Abnormal event detection at 150 fps in matlab,” in *Computer Vision (Computer Vision (ICCV), IEEE International Conference on)*, 2013, pp. 2720–2727.
- [4] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, “Anomaly detection in crowded scenes,” in *Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1975–1981.
- [5] W. Li, V. Mahadevan, and N. Vasconcelos, “Anomaly detection and localization in crowded scenes,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 18–32, 2014.
- [6] C. Piciarelli, C. Micheloni, and G. L. Foresti, “Trajectory-based anomalous event detection,” *IEEE Transactions on Circuits and Systems for video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
- [8] F. Jiang, Y. Wu, and A. K. Katsaggelos, “A dynamic hierarchical clustering method for trajectory-based unusual video event detection,” *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 907–913, 2009.
- [9] S. Calderara, U. Heinemann, A. Prati, R. Cucchiara, and N. Tishby, “Detecting anomalies in people’s trajectories using spectral graph analysis,” *CVIU*, vol. 115, no. 8, pp. 1099–1111, 2011.
- [10] S. Wu, B. E. Moore, and M. Shah, “Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes,” in *Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2054–2060.
- [11] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 935–942.
- [12] T. Xiao, C. Zhang, and H. Zha, “Learning to detect anomalies in surveillance video,” *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1477–1481, 2015.
- [13] M. Javan Roshtkhari and M. D. Levine, “Online dominant and anomalous behavior detection in videos,” in *Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2611–2618.
- [14] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Video anomaly detection and localization using hierarchical feature representation and gaussian process regression,” in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015, pp. 2909–2917.

- [15] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *ACM international conference on Multimedia*. ACM, 2007, pp. 357–360.
- [16] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, “Learning deep representations of appearance and motion for anomalous event detection,” in *BMVC*, 2015.
- [17] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, “Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes,” *IEEE Transactions on Image Processing*, 2017.
- [18] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [19] J. Gauthier, “Conditional generative adversarial nets for convolutional face generation,” *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester*, vol. 2014, no. 5, p. 2, 2014.
- [20] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” *arXiv preprint arXiv:1511.05440*, 2015.
- [21] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” *arXiv preprint arXiv:1609.04802*, 2016.
- [22] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, “Semantic segmentation using adversarial networks,” *arXiv preprint arXiv:1611.08408*, 2016.
- [23] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *arXiv preprint arXiv:1611.07004*, 2016.

- [24] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [25] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” *arXiv preprint arXiv:1701.04862*, 2017.
- [26] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling,” in *Advances in neural information processing systems*, 2016, pp. 82–90.
- [27] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [28] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, “Efficient learning of sparse representations with an energy-based model,” in *Advances in neural information processing systems*, 2006, pp. 1137–1144.
- [29] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *International Conference on Machine Learning*, 2008, pp. 1096–1103.
- [30] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive autoencoders: Explicit invariance during feature extraction,” in *International Conference on Machine Learning*, 2011, pp. 833–840.
- [31] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” *ICANN*, pp. 52–59, 2011.
- [32] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, “Deep learning representation using autoencoder for 3d shape retrieval,” *Neurocomputing*, vol. 204, pp. 41–50, 2016.

- [33] J. Xie, Y. Fang, F. Zhu, and E. Wong, “Deepshape: Deep learned shape descriptor for 3d shape matching and retrieval,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1275–1283.
- [34] G. Yu and J. Yuan, “Fast action proposals for human action detection and search,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1302–1311.
- [35] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, “From keyframes to key objects: Video summarization by representative object proposal selection,” in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1039–1048.
- [36] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, “Abnormal event detection in videos using generative adversarial nets,” *arXiv preprint arXiv:1708.09644*, 2017.
- [37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Computer Vision (Computer Vision (ICCV), IEEE International Conference on)*, 2015, pp. 4489–4497.
- [38] “Unusual crowd activity dataset made available by the university of minnesota at: <http://mha.cs.umn.edu/movies/crowdactivity-all.avi>.”
- [39] Y. Cong, J. Yuan, and Y. Tang, “Video anomaly search in crowded scenes via spatio-temporal motion context,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 10, pp. 1590–1599, 2013.
- [40] F. Jiang, J. Yuan, S. A. Tsafaris, and A. K. Katsaggelos, “Anomalous video event detection using spatiotemporal context,” *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 323–333, 2011.

- [41] L. Kratz and K. Nishino, “Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models,” in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1446–1453.
- [42] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.
- [43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
- [44] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [45] Y. Kozlov and T. Weinkauff, “Persistence1d: Extracting and filtering minima and maxima of 1d functions,” *http://people.mpi-inf.mpg.de/weinkauff/notes/persistence1d.html*, accessed, pp. 11–01, 2015.
- [46] D. Tran, J. Yuan, and D. Forsyth, “Video event detection: From subvolume localization to spatiotemporal path search,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 404–416, 2014.
- [47] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *ACM international conference on Multimedia*, 2014, pp. 675–678.
- [48] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.
- [49] J. Kim and K. Grauman, “Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates,” in *Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 2921–2928.

Bibliography

- [50] R. Raghavendra, A. Del Bue, M. Cristani, and V. Murino, “Abnormal crowd behavior detection by social force optimization,” in *International Workshop on Human Behavior Understanding*, 2011, pp. 134–145.