

Anomaly Detection in Video

by

Tran Thi Minh Hanh

**Submitted in accordance with the requirements
for the degree of Doctor of Philosophy**



The University of Leeds

School of Computing

June 2018

Declarations

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some parts of the work presented in this thesis have been published in the following articles:

Hanh T. M. Tran and David C. Hogg. Anomaly Detection using a Convolutional Winner-Take-All Autoencoder. *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, September 2017.

The candidate confirms that the above jointly-authored publications are primarily the work of the first author. The role of the second author was purely supervisory.

This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Abstract

Anomaly detection is an area of video analysis that has great importance in automated surveillance. Although it has been extensively studied, there has been little work on using deep convolutional neural networks to learn spatio-temporal feature representations. In this thesis we present novel approaches for learning motion features and modelling normal spatio-temporal dynamics for anomaly detection.

The contributions are divided into two main chapters. The first introduces a method that uses a convolutional autoencoder to learn motion features from foreground optical flow patches. The autoencoder is coupled with a spatial sparsity constraint, known as Winner-Take-All, to learn shift-invariant and generic flow-features. This method solves the problem of using hand-crafted feature representations in state of the art methods. Moreover, to capture variations in scale of the patterns of motion as an object moves in depth through the scene, we also divide the image plane into regions and learn a separate normality model in each region. We compare the methods with state of the art approaches on two datasets and demonstrate improved performance.

The second main chapter presents a end-to-end method that learns normal spatio-temporal dynamics from video volumes using a sequence-to-sequence encoder-decoder for prediction and reconstruction. This work is based on the intuition that the encoder-decoder learns to estimate normal sequences in a training set with low error, thus it estimates an abnormal sequence with high error. Error between the network's output and the target output is used to classify a video volume as normal or abnormal. In addition to the use of reconstruction error, we also use prediction error for anomaly detection.

We evaluate the second method on three datasets. The prediction models show comparable performance with state of the art methods. In comparison with the first proposed method, performance is improved in one dataset. Moreover, running time is significantly faster.

Acknowledgements

First of all I would like to express my utmost gratitude to my supervisor Prof. David Hogg. His guidance, feedback and encouragement have inspired me to pursue novel ideas and he have helped me to develop the research skills. I would also like to thank him and the department for funding me for various events such as a summer school and conferences during my PhD.

My sincere gratitude goes to Project 911 - Vietnam International Education Department (VIED) Scholarship which funded me for three years of my PhD. Without VIED scholarship, I would not have had the opportunity to come to UK and do my PhD at University of Leeds.

Also a big thank you to all current and ex PhD students, Aryana Tavanai, Christiana Panayi, Leo Pauly, Rebecca Stone and so many others who have discussed ideas and helped me over the last four years. My special thanks go to Duane Carey and Fouzhan Hosseini who have advised and encouraged me through the hardest times. I would also like to thank the staff in the School of Computing for a lot of fruitful discussions and support, and for providing a conducive environment to research. There are many others, too many to list here, but my gratitude goes to everyone who has helped me or been a friend to me over the years. I thank you all.

Most importantly however I would like to thank my parents and family in Vietnam. Without the never ending support and encouragement of my parents, I would not be here now. I would also like to thank my husband, Nguyen Van Quyen, who has supported me at the toughest times, understanding when I was so caught up in my work and encouraging me when things went wrong. My special thanks go to my lovely son, who has brought so much joy laughter and happiness into my PhD.

Contents

1	Introduction	1
1.1	Challenges	2
1.2	Motivation	3
1.3	Aims and Objectives	4
1.4	Novelty and Significance	6
1.5	Limitations and Constraints	7
1.6	Outline	7
2	Related Work	9
2.1	Introduction	9
2.2	Non-deep learning methods for anomaly detection	10
2.2.1	Feature descriptors	11
2.2.2	Generative models of descriptors	17
2.2.3	Discriminative models of descriptors	20
2.3	Deep learning in anomaly detection	24
2.3.1	Autoencoder and its variants	25
2.3.2	Deep feature learning for anomaly detection	32
2.3.3	End-to-end deep network	34
2.4	Experimental validation	37
2.4.1	Evaluation measure	37
2.4.2	Datasets	39
2.5	Summary	43
3	Convolutional Winner-Take-All Autoencoder for anomaly detection	45
3.1	Introduction	45
3.2	Our method	47
3.2.1	Extracting foreground patches	47
3.2.2	Convolutional Winner-Take-All autoencoder	48

3.2.3	Max pooling and temporal averaging for motion feature representation	50
3.2.4	Convolutional autoencoder	52
3.2.5	One class SVM modelling	55
3.3	Experimental evaluation	57
3.3.1	Dataset and Evaluation measures.	57
3.3.2	Experimental Settings	57
3.3.3	Quantitative Analysis	58
3.3.3.1	Comparison with the state of the art	58
3.3.3.2	Varying patch size.	62
3.3.3.3	Varying max-pooling size.	62
3.3.3.4	Smoothness over number of frames	63
3.3.3.5	Efficiency of the Winner-Take-All sparsity constraint	64
3.3.4	Qualitative Analysis	65
3.3.5	Number of parameters and running time	67
3.4	Conclusions	68
4	Convolutional Long Short-Term Memory for anomaly detection	69
4.1	Introduction	69
4.2	Architecture	70
4.2.1	Input data layer	72
4.2.2	Convolutional and Deconvolutional layer	73
4.2.3	Recurrent Neural Network using Long Short-term Memory	75
4.2.3.1	Long Short Term Memory	76
4.2.3.2	Convolutional Long Short Term Memory	77
4.3	Optimization and Initialization	78
4.4	Regularity score for anomaly detection	80
4.5	Experiments	80
4.5.1	Datasets	81
4.5.2	Anomalous event detection	81
4.5.3	Reconstruction and Prediction	90
4.5.4	Number of model parameters and tesing time	94
4.6	Conclusion	95
5	Conclusion and Future Work	96
5.1	Contributions	97
5.1.1	Convolutional Winner-Take-All autoencoder	97

5.1.2	Convolutional Long Short-Term Memory	97
5.2	Limitations	98
5.3	Future Work	99
5.4	Closing Remarks	100
6	Annex	102
6.1	Convolutional WTA autoencoder for anomaly detection	102
6.1.1	One-class SVM kernels	102
6.1.2	Normalization as a preprocessing step for OCSVM.	102
6.1.3	A convolutional WTA autoencoder with different number of convolutional layers.	103
6.2	Convolutional Long Short-Term Memory for anomaly detection	105
6.2.1	Regularity score of different models.	105
	Bibliography	110

List of Figures

1.1	Overview of our method that uses a convolutional autoencoder to learn motion representations for anomaly detection. The details of this method will be described in Chapter 3.	5
1.2	Overview of our method that uses prediction error for anomaly detection. The details of this method will be described in Chapter 4.	6
2.1	An ensemble of patches in video, c is an origin of the ensemble. Figure from [1].	12
2.2	(A) Multi-scale Histogram of Optical Flow [2] is extracted from a basic unit. (B) The selection of spatial-temporal basis used for anomaly detection. Figure reproduced from [2].	13
2.3	AMHOF descriptors of three regions of interest. Figure from [3].	14
2.4	The interaction force (red) of two sampled frames is calculated based on optical flow (yellow). Figure from [4].	15
2.5	(a) Distribution based HMM and (b) coupled HMM for capturing temporal and spatial relationships. Figure from [5].	17
2.6	(a) Temporal and (b) Spatial anomaly detection with MDT models. Figure from [6].	20
2.7	SVM for binary classification, where support vectors are the training points that lie near the hyperplane defining the margin.	23
2.8	Taxonomy: most popular autoencoders classified according to the characteristics they induce in their encodings [7].	25
2.9	A general autoencoder structure.	25
2.10	Common activation functions.	26
2.11	Under-complete and over-complete auto-encoders.	27
2.12	Architecture of the shift-invariant unsupervised feature extractor applied to the two bars dataset. Figure from [8].	29
2.13	2D and 3D convolution operations [9].	30

2.14	Two examples of a LSTM autoencoder.	31
2.15	(a) A Generative Adversarial Network and (b) an example of a conditional GAN for mapping edges to photos (Figure from [10]).	32
2.16	The fully convolutional network with a trainable layer on top of pretrained layers. (Figure from [11]).	33
2.17	The overview of frameworks using three stacked denoising autoencoders to learn appearance, motion and joint representations. (Figure from [11]).	34
2.18	Stacked convolutional auto-encoders used for anomaly detection. (a) Spatial-temporal information is learned on a sequence of 10 frames with a convolutional auto-encoder (Figure reproduced from [12]) and (b) a convolutional LSTM is applied on top of convolutional layer's feature maps to learn temporal information (Figure reproduced from [13]).	35
2.19	Architecture of a 3D convolutional auto-encoder for anomaly detection with reconstruction and prediction branches. Figure from [14].	36
2.20	The ROC curve, where the blue area is AUC and the intersection between the line ($FPR = 1 - TPR$) and the curve is EER. A better method gives higher AUC and lower EER.	39
2.21	Sample normal/abnormal frames in UCSDPed1 (top row) and UCSDPed2 (bottom row). Anomalous pixels are shown in red.	40
2.22	Examples of abnormal frames in Avenue dataset, where red boxes correspond to abnormal events.	41
2.23	Examples of anomalous frames in Subway Entrance (top row) and Subway Exit dataset (bottom row). Red boxes correspond to abnormal events. . .	42
3.1	Overview of the method using a spatial sparsity Convolutional Winner-Take-All autoencoder for anomaly detection.	46
3.2	(a) The flow field color coding [15] used in this chapter, where flow-vector angle and magnitude are represented by hue and saturation; (b) Examples of training patches.	48
3.3	Foreground patches extraction using a sliding window and thresholding of the accumulated optical flow squared magnitude. (a) Video frame at time t . (b) Map of the flow magnitude (from frames t and $t + 1$) with overlapping foreground patches superimposed; the red square delineates a single (24×24) foreground patch.	49
3.4	The architecture for a Conv-WTA autoencoder with spatial sparsity for learning motion representations.	49

3.5	Learned deconvolutional filters of the Conv-WTA autoencoder trained on the UCSDPed1 and UCSDPed2 optical flow foreground patches: (a) visualisation of 128 filters, and (b) displacement vector visualisation of 25 filters.	51
3.6	The convolutional WTA feature extractor.	52
3.7	Convolutional autoencoder for learning motion representations.	53
3.8	The convolutional feature extractor.	53
3.9	Training error of the convolutional autoencoder and the convolutional WTA autoencoder.	53
3.10	Learned deconvolutional filters of the convolutional autoencoder trained on the UCSDPed1 and UCSDPed2 optical flow foreground patches: (a) visualisation of 128 filters, and (b) displacement vector visualisation of 25 filters.	54
3.11	Examples of one-class SVM regions.	56
3.12	Frame-level and pixel-level evaluation on the UCSDPed1. The legend for the pixel-level (right) is the same as for the frame-level (left).	59
3.13	Frame-level and pixel-level evaluation on the UCSDPed2.	59
3.14	Frame-level comparison on the Avenue dataset.	61
3.15	Detection results on the UCSDPed1 (first 2 rows), the UCSDPed2 (third row) and the CUHK Avenue dataset (fourth row).	65
3.16	False detection results on the UCSDPed1 (first row), the UCSDPed2 (second row) and the Avenue dataset (third row).	66
4.1	Regularity scores obtained from an extract from the CUHK Avenue dataset[16]. The regularity score drops when an abnormal event appears.	70
4.2	The convLSTM encoding-decoding structure used for future prediction with $\tau = 5$	71
4.3	The convolutional encoding-decoding structure with skip connections used for future prediction with $\tau = 5$	71
4.4	A diagram of an Long Short-term Memory cell, an activation function can be tanh or ReLU.	76
4.5	The validation error of the convLSTM encoder-decoder trained on each dataset.	79
4.6	The train and validation errors of the convLSTM encoder-decoder trained on UCSDPed1 and UCSDPed2.	79

4.7	Prediction error in the timestamp area affects the regularity score. A blue-green-red color map shows error from low to high.	84
4.8	Regularity score of video sequence #1, 5, 24, 17, 23 (from top to bottom) of UCSDPed1 dataset.	86
4.9	Regularity score of video sequence #2, 4, 5, 7 (from top to bottom) of UCSDPed2 dataset.	87
4.10	Regularity score of each sequence of video sequence #5, 7, 15, 12 (from top to bottom) of CUHK Avenue dataset.	88
4.11	Regularity score of frames #115, 000 – 120, 000 of Subway entrance dataset.	89
4.12	Regularity score of frames #52, 500 – 64, 000 of Subway exit dataset (without and with masking the timestamp).	89
4.13	Reconstruction and prediction on sample irregular frames of UCSDPed1 which contains a car. Best viewed in color.	91
4.14	Reconstruction and prediction on sample irregular frames of UCSDPed1 which contains a wheelchair. Best viewed in color.	92
4.15	Reconstruction and prediction on sample irregular frames of UCSDPed2 which contains a biker. Best viewed in color.	93
4.16	Reconstruction and prediction on sample irregular frames of CUHK Avenue which contains a running person. Best viewed in color.	94
5.1	A proposed encoder-decoder with the presence of negative samples from the generator	100
6.1	Learned deconvolutional filters of the Conv-WTA autoencoder with 6 convolutional layers trained on optical flow foreground patches (UCSD dataset).	104
6.2	Regularity score of video sequence #1, 5, 24, 17, 23 (from top to bottom) of UCSDPed1 dataset.	106
6.3	Regularity score of video sequence #2, 4, 5, 7 (from top to bottom) of UCSDPed2 dataset.	107
6.4	Comparison of regularity scores deriving from prediction and reconstruction errors on video sequence #5, 7, 15, 12 (from top to bottom) of CUHK Avenue dataset.	108
6.5	Regularity score of frames #115, 000 – 120, 000 of Subway entrance dataset (with masking the timestamp).	109
6.6	Regularity score of frames #52, 500 – 64, 000 of Subway exit dataset (with masking the timestamp).	109

List of Tables

2.1	Composition of abnormal events in the UCSD dataset.	41
2.2	Groundtruth of Avenue dataset.	41
2.3	Groundtruth of Subway dataset.	43
3.1	Performance comparison on UCSDPed1 and UCSDPed2. (* the results in [17] include replicated results for MPPCA [18] and Social Force Model [4] methods.)	60
3.2	Performance comparison on the Avenue dataset. (* the results from [19] replicated SCL method [16])	61
3.3	Detection accuracy (%) with IoU threshold v on CUHK Avenue dataset. OCSVM[8 × 12] is used.	61
3.4	Performance comparison on UCSDPed1 and UCSDPed2 with different patch sizes.	62
3.5	Performance comparison on UCSDPed1 with different kernel sizes and strides of max-pooling and different subdivisions.	63
3.6	EER/AUC for different temporal smoothing windows.	64
3.7	Impact of the WTA constraint.	64
3.8	The details of the number of parameters for each autoencoder.	67
3.9	Testing time (second/frame) without and with GPU.	67
4.1	The details of the convolutional LSTM encoder-decoder model with 12 layers. The two dimensions in “Kernel”, “In Res” and “Out Res” represent for height and width.	74
4.2	The details of the convolutional LSTM encoder-decoder model with 8 layers.	74
4.3	The details of the 2D convolutional encoder-decoder with skip connections.	75
4.4	Number of training data and training epochs correspond to each dataset.	82

4.5	Comparison of EER/AUC with different architectures and setups of the convLSTM encoder-decoder. τ is the number of frames in an input volume and a target volume.	82
4.6	Comparison of EER/AUC with different types of the encoder-decoders (the convolutional encoder-decoder and the convLSTM encoder-decoder) for reconstruction and prediction, aug2 is used for data augmentation, $\tau = 5$ is used.	83
4.7	Performance comparison with the state of the art.	83
4.8	Performance comparison in Subway Entrance/Exit datasets with and without masking the timestamp.	85
4.9	Comparison on number of parameters.	95
4.10	Testing time (second/frame) without and with GPU.	95
6.1	Performance comparison on UCSDPed2 with different kernels for OCSVM.	102
6.2	Performance comparison on UCSDPed2 with different normalization methods for OCSVM.	103
6.3	Performance comparison on UCSDPed1 and UCSDPed2 with different network's architectures.	104

List of Notations

The following is a list of important math notations used in the thesis. In general, the following rules are used for numbers and arrays:

- Bold capital letters (e.g. \mathbf{W}) denote matrices.
- Bold small letters (e.g. \mathbf{w}) denote column vectors. A row vector is denoted by its transpose, e.g. \mathbf{w}^T .
- Non-bold letters (e.g. x, l, C) are for scalars.

Latin

- a - Negative slope in leaky ReLU layer
- \mathbf{b} - Network bias
- \mathbf{b}_l - Network bias of layer l
- C_l - The depth of output tensor of layer l
- C_0 - The depth of input tensor
- \mathbf{d} - Feature representation of a patch
- \mathbf{E} - Output tensor
- e - Prediction/Reconstruction error
- \mathcal{F}_t - A video frame at time t
- \mathbf{P} - Foreground patch
- \mathbf{P}_n - n -th foreground patch
- $\hat{\mathbf{P}}$ - Estimation of the foreground patch
- H_l - The height of output tensor of layer l
- H_0 - The height of input tensor
- N - Batch size
- \mathbf{w} - Decision hyperplane normal vector

r - Regularity score

s - Anomaly score

thr - A threshold for anomaly score

W_l - The width of output tensor of layer l

W_0 - The width of input tensor

\mathbf{W}_l - Network weights at layer l

\mathbf{W} - Network weights

(x, y, c) - The row, column and channel indices of an element in the tensor

Greek

λ - Regularization term or weight decay

α_i, β_i - Lagrangian multipliers

ξ_i - Slack variables in one-class Support Vector Machine

ν - One-class Support Vector Machine parameter

ρ - Bias

γ - Radial basis kernel function parameter

v - Intersection over Union threshold

τ - Temporal window

θ - A video volume

Functions

g, f - Activation function

$\sigma(x)$ - Logistic sigmoid, $\frac{1}{1+\exp(-x)}$

Φ - Feature projection function

k - Kernel function

\mathcal{L} - Loss function

$\|\mathbf{W}\|_F^2$ - Frobenius norm of \mathbf{W}

$\|\mathbf{W}\|_2^2$ - L_2 norm or Euclidean norm of \mathbf{W}

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ - Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b}$ - Dot product between column vector \mathbf{a} and \mathbf{b}

List of Acronyms

AMHOF - Adaptive Multi-scale Histogram of Optical Flow

AE - Autoencoder

CAE - Convolutional Autoencoder

CNN - Convolutional Neural Network

Conv-WTA - Convolutional Winner-Take-All

ConvLSTM - Convolutional Long Short Time Memory

CRF - Conditional Random Field

GMM - Gaussian Mixture Model

HOF - Histogram of Optical Flow

HMM - Hidden Markov Model

KL - Kullback-Leibler

LDA - Latent Dirichlet Allocation

LSTM - Long Short Time Memory

MHOF - Multi-scale Histogram of Optical Flow

MPPCA - Mixture of Probabilistic Principal Component Analyser

MDT - Mixture of Dynamic Textures

MRF - Markov Random Field

OCSVM - One Class Support Vector Machine

PCA - Principal Component Analysis

ReLU - Rectified Linear Unit

RNN - Recurrent Neural Network