Learning Not to Reconstruct Anomalies

Marcella Astrid[1,2]
marcella.astrid@ust.ac.kr
Muhammad Zaigham Zaheer[1,2]
mzz@ust.ac.kr
Jae-Yeong Lee[1,2]
jylee@etri.re.kr
Seung-Ik Lee[1,2]
the_silee@etri.re.kr

[1] University of Science and Technology
Daejeon, South Korea
[2] Electronics and Telecommunications
Research Institute
Daejeon, South Korea

Abstract

This is the supplementary material for the BMVC submission "Learning Not to Reconstruct Anomalies". Section 1 describes the SmoothMixS function used to generate patch based pseudo anomalies. Section 2 explains additional experimental details including implementation settings, dataset used, additional qualitative results, equal error rate results, discussion regarding pixel-level performance, and comparisons with GAN-based pseudo anomaly generation, which were omitted from the main manuscript due to limited space.

1 Details of Patching Techniques

In this section, we describe the patching technique function utilized in patch based pseudo anomalies (Section 3.3.1 of manuscript) in more details.

To generate the i-th frame of pseudo anomaly X_i^P of the input sequence, we combine an arbitrary image I^A from an intruding dataset and the i-th normal frame X_i^N of the sequence as follows:

$$X_i^P = G_i \circ I^A + (1 - G_i) \circ X_i^N, \tag{1}$$

where $\circ$ is pixel-wise multiplication in the spatial dimension, G_i is mask for the i-th frame, and I^A is an arbitrary image taken from the intruding dataset and transformed to have the same size as X_i^N, i.e., $C \times H \times W$. The overall process can be seen in Fig. 1(a) with example if SmoothMixS [8] is utilized. By default, we utilize SmoothMixS [8], however, we also experiment with other techniques, such as CutMix, SmoothMixC, and MixUp-patch. Mask G_i generated from each of the technique can be seen in Fig. 1(b). Details of each mask generation are explained below:

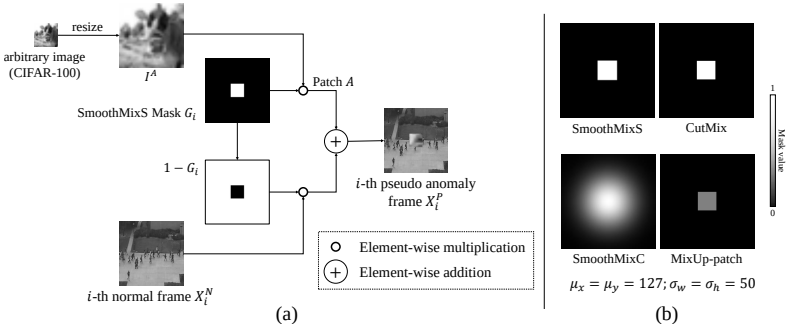

Figure 1: (a) The overall process of patch based pseudo anomaly generation using SmoothMixS. (b) Different types of mask generated from different patching techniques having the same μ_x, μ_y, σ_w, and σ_h values.

1.1 SmoothMixS

The SmoothMixS [8] mask is generated by utilizing two one dimensional masks, i.e., width and height dimensions. SmoothMixS mask in the width dimension is given as:

$$G_i^w = \begin{cases} \max(0, \frac{m-(\mu_i^x-(1+k)\frac{\sigma^w}{2})}{k\frac{\sigma^w}{2}}) & \text{if } m \leq \mu_i^x - \frac{\sigma^w}{2}, \\ \max(0, -\frac{m-(\mu_i^x+(1+k)\frac{\sigma^w}{2})}{k\frac{\sigma^w}{2}}) & \text{if } m \geq \mu_i^x + \frac{\sigma^w}{2}, \\ 1 & \text{otherwise,} \end{cases} \tag{2}$$

where k is the smoothness hyperparameter, a higher value of which represents more slope (set to 0.1). Furthermore, σ^w is the span of the mask in width dimension, μ_i^x is the center x-coordinate of the patch in the i-th frame, and m represents the x-coordinate over the mask. Similar calculation is also done for mask in height dimension G_i^h using μ_i^y (y-coordinate of the center) and σ^h (span of the mask in height dimension):

$$G_i^h = \begin{cases} \max(0, \frac{n-(\mu_i^y-(1+k)\frac{\sigma^h}{2})}{k\frac{\sigma^h}{2}}) & \text{if } n \leq \mu_i^y - \frac{\sigma^h}{2}, \\ \max(0, -\frac{n-(\mu_i^y+(1+k)\frac{\sigma^h}{2})}{k\frac{\sigma^h}{2}}) & \text{if } n \geq \mu_i^y + \frac{\sigma^h}{2}, \\ 1 & \text{otherwise,} \end{cases} \tag{3}$$

where n represents the y-coordinate over the mask. Illustrations on G_i^w and G_i^h values can be seen in Fig. 2.

The final mask to be used in Eq. (1) is then given as:

$$G_i = G_i^w \otimes G_i^h, \tag{4}$$

where $\otimes$ is the outer product.

1.2 CutMix

The CutMix [34] mask is a rectangular shape mask with (μ_x, μ_y) as the center coordinate of mask and (σ_w, σ_h) as the width and height of the mask.

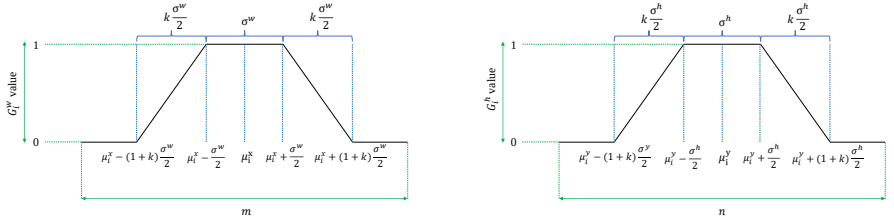

Figure 2: Illustrations of SmoothMixS [8] mask values (a) G_i^w (Eq. (2)) and (b) G_i^h (Eq. (3)).

1.3 SmoothMixC

The SmoothMixC [8] mask is generated using Gaussian distribution with (μ_x, μ_y) as the mean and (σ_w, σ_h) as the standard deviation.

1.4 MixUp-patch

The MixUp-patch mask is a combination of MixUp [37] and CutMix [34]. MixUp mixes two images with linear combination at pixel-level. However, to be consistent with the aforementioned patch based techniques, we utilize the patch shape of CutMix instead of overlapping the whole image. Moreover, instead of mask with value of 1 used in the CutMix, the mask in the MixUp-patch used in our experiments is set to the value of 0.5. Hence, the final patch is a combination of the input frame and the patch from intruder dataset.

2 Experiments

In this section, we discuss experimental details, including implementation details, details of each datasets, additional qualitative results, equal error rate results, regarding pixel-level performance, and comparisons with GAN-based pseudo anomaly, which have been omitted from the manuscript because of space limitation.

2.1 Implementation Details

For the autoencoder models used in our experiments, we adopt a generative architecture recently proposed by Gong *et al*. [4] that takes a grayscale input sequence of size $16 \times 1 \times 256 \times 256$ and produces its reconstruction of the same size. All the 16 frames of the sequence are used for computing the reconstruction loss during training, while only the 9th frame of the sequence is considered for anomaly score calculations at test time. However, differently from the original implementation in [4], we remove the memory network and utilize only the autoencoder part. Furthermore, we normalize each input image to the range $[-1, 1]$, as well as add a Tanh output layer to have the similar output range. Architectural details of the autoencoder used in our experiments can be seen in Table 1. The training is carried out using Adam optimizer [7] with a learning rate of 10^{-4} and the mini batch size is set to 4. The implementation is done in PyTorch [16].

	Layer	Output Channels	Filter Size	Stride	Padding	Negative Slope
Encoder	Conv3D	96	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	Conv3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
Decoder	ConvTranspose3D	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	96	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
	BatchNorm3D	-	-	-	-	-
	LeakyReLU	-	-	-	-	0.2
	ConvTranspose3D	1	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
	Tanh	-	-	-	-	-

Table 1: Autoencoder architecture used in our work. Each number in the tuple represents time, height, and width dimensions, respectively.

2.2 Datasets

The details of each dataset used in our experiments are as follows:

Ped2 [9]. This dataset consists of 16 training and 12 test videos. Pedestrians dominate most of the normal frames, whereas anomalies include people riding bicycles, vehicles, or skateboards.

Avenue [10]. This dataset contains 16 training and 21 test videos. Anomalies include bicycles, people running, and people throwing stuff.

ShanghaiTech [11]. This is by far the largest one-class anomaly detection dataset. It consists of 330 training and 107 test videos. The dataset is recorded at 13 different scenes and various camera angles. In total, the test videos contain 130 anomalous events including running, riding bicycle, and fighting.

2.3 Additional Qualitative Results

To investigate the score distribution performance, Fig. 3 shows the comparisons of anomaly score distributions between the baseline and our models on the complete test split of Ped2 dataset. Training using pseudo anomalies increases the reconstruction error of the anomalous regions hence improving the capability of AEs to discriminate anomaly scores between the normal and anomalous frames resulting in a noticeable AUC performance improvement. In addition to the qualitative visualizations in Fig. 4 of the manuscript, we also provide a demonstration video (demonstration.mp4) along with this Supplementary document.

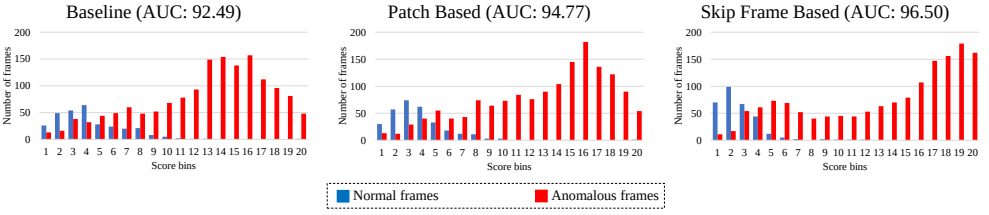

Figure 3: Anomaly score distributions of normal and anomalous frames using a complete test split of Ped2. The anomaly scores are divided into 20 equal bins. Compared to the baseline (a), our models trained using patch based pseudo anomalies (b) and skip frame based pseudo anomalies (c) show better separability between normal and abnormal scores, resulting in a superior performance as evident from the improved AUC.

	Method	Ped2 [9]		Method	Ped2 [9]
Non deep learning	MDT [32]	24	Deep learning	Sabokrou et al. [23]	12.6
	MPCCA [6]	30		AMDN [33, 34]	17
	MPCCA+SF [32]	36		Sabokrou *et al.* [22]	19
	IBC [8]	13		Deep-Cascade [24]	**8.2**
	Bertini *et al.* [3]	30		Deep-Anomaly [25]	11
	Xu *et al.* [31]	21		Ravanbakhsh *et al.* [20]	11
	Adam *et al.* [1]	42		ALOCC [26]	13
	Mehran *et al.* [13]	42		Sabokrou *et al.* [28]	12.5
	Reddy *et al.* [21]	20.0		Abnormal GAN [18]	14
	Saligrama and Chen [29]	18		Plug-and-Play [19]	18
	Xiao *et al.* [30]	<u>10</u>		AVID [27]	14
	Zaharescu and Wildes [35]	17		Baseline	15.33
	Li *et al.* [9]	18.5		**Ours**: Patch based	11.73
	iHOT [14]	**8.59**		**Ours**: Skip frame based	<u>8.39</u>

Table 2: EER comparisons with SOTA. The best and the second best performance of each category are marked as bold and underlined, respectively.

2.4 Equal Error Rate

Besides AUC used in our manuscript, Equal Error Rate (EER) is also a metric to measure the performance of anomaly detection system used in the literature, especially for Ped2 dataset. EER is the error at the point where false positive and false negative rates are equal. Table 2 shows the EER results of our models and other approaches in Ped2 dataset. To the best of our knowledge, EER is typically reported only on Ped2 dataset. Nevertheless, for completeness, our EER results on Avenue are 26.96, 22.46, and 22.14 for the baseline, our model trained using patch based pseudo anomalies, and our model trained using skip frame based pseudo anomalies, respectively. Similarly, in the same order of models, the results on ShanghaiTech are 34.08, 33.78, and 29.08. Similar to using AUC metric, our models trained using pseudo anomalies achieve better performance compared to the baseline using EER metric. Moreover, compared to the other methods, our models are competitive.

2.5 Pixel-level Performance

In anomaly detection systems, frame-level performance metrics are widely popular [4, 5, 15, 36]. A pixel-level metric could be another option to evaluate such systems. In pixel-level evaluation, a true positive is counted when at least 40% of the anomaly ground truth pixels in the frame are predicted as anomalous, otherwise predicted as false positive [17]. However, it was subsequently proposed by Li *et al*. [9] to count false positive only for the frames that do not contain any anomaly annotation. These inconsistent interpretations of the pixel-level evaluations make the anomaly detection systems incomparable [17]. Therefore, although our approach is capable of localizing anomalies within frames, we do not provide pixel-level AUC comparisons.

2.6 Comparisons with the Pseudo Anomaly Generated by GAN

To provide more extensive analysis, we also experiment with pseudo anomalies generated by Generative Adversarial Network (GAN) using Ped2. Following [36], we obtain a bad generator by keeping an adversarial AE undertrained. The architecture of the generator is same with our AE (Table 1) and the architecture of the discriminator can be seen in Table 3. To train the generator $\mathcal{G}$, we utilize both reconstruction loss L_{recon} and adversarial loss L_{adv} as:

$$L_{\mathcal{G}} = L_{adv} + \lambda L_{recon}, \tag{5}$$

where L_{adv} is binary cross entropy loss to fool the discriminator to predict the reconstructions as real, L_{recon} is L^N (Eq. (2) of our main manuscript), and λ is the weighting factor. To train the discriminator, binary cross entropy loss to differentiate between real sequences and fake sequences (reconstructed sequences) is used. The adversarial training is conducted for only few first epochs before the generator starts producing good reconstructions. Then, we utilize only the generator to generate bad reconstructions as pseudo anomalies. As seen in Table 4, we experiment with several bad generators trained using different settings, i.e., λ, stopping epoch, and discriminator learning rate. We achieve a maximum of 93.66% AUC, which is better than the baseline (92.49%) but still inferior to our proposed pseudo anomalies based methods. The lower performance may be caused by non-optimal bad generator, since it is difficult to optimize the GAN-based pseudo anomaly generation without any particular objective function. Therefore, the GAN-based pseudo anomalies used with our method may improve the baseline performance. However, the proposed pseudo anomaly generation is more robust and yields superior results.

References

[1] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):555–560, 2008.

[2] Marco Bertini, Alberto Del Bimbo, and Lorenzo Seidenari. Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3):320–329, 2012.

[3] Oren Boiman and Michal Irani. Detecting irregularities in images and in video. *International journal of computer vision*, 74(1):17–31, 2007.

Layer	Output Channels	Filter Size	Stride	Padding	Negative Slope
Conv3d	96	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
BatchNorm3D	-	-	-	-	-
LeakyReLU	-	-	-	-	0.2
Conv3d	128	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
BatchNorm3D	-	-	-	-	-
LeakyReLU	-	-	-	-	0.2
Conv3d	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
BatchNorm3D	-	-	-	-	-
LeakyReLU	-	-	-	-	0.2
Conv3d	256	(3, 3, 3)	(2, 2, 2)	(1, 1, 1)	-
BatchNorm3D	-	-	-	-	-
LeakyReLU	-	-	-	-	0.2
Conv3d	256	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
BatchNorm3D	-	-	-	-	-
LeakyReLU	-	-	-	-	0.2
Conv3d	256	(3, 3, 3)	(1, 2, 2)	(1, 1, 1)	-
BatchNorm3D	-	-	-	-	-
LeakyReLU	-	-	-	-	0.2
Flatten	-	-	-	-	-
Linear	1	-	-	-	-
Sigmoid	-	-	-	-	-

Table 3: Architecture of discriminator used to with bad generator.

[4] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1705–1714, 2019.

[5] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019.

[6] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2928. IEEE, 2009.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Jin-Ha Lee, Muhammad Zaigham Zaheer, Marcella Astrid, and Seung-Ik Lee. Smoothmix: a simple yet effective data augmentation to train robust classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 756–757, 2020.

[9] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and local-

λ	Disc. LR	Pretrained Epochs	Average	Max
1	10^{-5}	1	**92.92**	93.49
0.1	10^{-5}	1	92.70	93.59
0.01	10^{-5}	1	92.32	93.24
10	10^{-6}	1	92.60	93.31
10	10^{-6}	2	92.17	92.97
10	10^{-6}	3	92.25	93.03
10	10^{-6}	4	92.68	93.04
10	10^{-6}	5	92.67	93.08
10	10^{-6}	6	92.69	92.96
10	10^{-6}	7	92.24	92.47
10	10^{-6}	8	92.38	92.88
10	10^{-6}	9	92.19	92.60
10	10^{-6}	10	92.55	93.20
1	10^{-6}	1	92.91	**93.66**
1	10^{-6}	5	92.40	92.98
1	10^{-6}	10	92.88	93.56
Baseline			92.28	92.49
Ours: Patch based			93.94	94.77
Ours: Skip frame based			**95.59**	**96.50**

Table 4: Average and maximum AUC (%) performance out of five repeated experiments on different bad generators trained on various λ, discriminator learning rate (Disc. LR), and number of epochs during pretraining. Experiments are conducted in Ped2.

ization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):18–32, 2013.

[10] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[11] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 341–349, 2017.

[12] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE, 2010.

[13] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 935–942. IEEE, 2009.

[14] Hossein Mousavi, Moin Nabi, Hamed Kiani Galoogahi, Alessandro Perina, and Vittorio Murino. Abnormality detection with improved histogram of oriented tracklets. In *International conference on image analysis and processing*, pages 722–732. Springer, 2015.

[15] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381, 2020.

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learn pdf.

[17] Bharathkumar Ramachandra, Michael Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[18] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1577–1581. IEEE, 2017.

[19] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto, and Nicu Sebe. Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1689–1698. IEEE, 2018.

[20] Mahdyar Ravanbakhsh, Enver Sangineto, Moin Nabi, and Nicu Sebe. Training adversarial discriminators for cross-channel abnormal event detection in crowds. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1896–1904. IEEE, 2019.

[21] Vikas Reddy, Conrad Sanderson, and Brian C Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *CVPR 2011 WORKSHOPS*, pages 55–61. IEEE, 2011.

[22] Mohammad Sabokrou, Mahmood Fathy, Mojtaba Hoseini, and Reinhard Klette. Real-time anomaly detection and localization in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 56–62, 2015.

[23] Mohammad Sabokrou, Mahmood Fathy, and Mojtaba Hoseini. Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters*, 52(13):1122–1124, 2016.

[24] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, and Reinhard Klette. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, 26(4):1992–2004, 2017.

[25] Mohammad Sabokrou, Mohsen Fayyaz, Mahmood Fathy, Zahra Moayed, and Reinhard Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018.

[26] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018.

[27] Mohammad Sabokrou, Masoud Pourreza, Mohsen Fayyaz, Rahim Entezari, Mahmood Fathy, Jürgen Gall, and Ehsan Adeli. Avid: Adversarial visual irregularity detection. In *Asian Conference on Computer Vision*, pages 488–505. Springer, 2018.

[28] Mohammad Sabokrou, Mahmood Fathy, Guoying Zhao, and Ehsan Adeli. Deep end-to-end one-class classifier. *IEEE transactions on neural networks and learning systems*, 2020.

[29] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2112–2119. IEEE, 2012.

[30] Tan Xiao, Chao Zhang, and Hongbin Zha. Learning to detect anomalies in surveillance video. *IEEE Signal Processing Letters*, 22(9):1477–1481, 2015.

[31] Dan Xu, Rui Song, Xinyu Wu, Nannan Li, Wei Feng, and Huihuan Qian. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing*, 143:144–152, 2014.

[32] Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. In *BMVC*, 2015.

[33] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127, 2017.

[34] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[35] Andrei Zaharescu and Richard Wildes. Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing. In *European Conference on Computer Vision*, pages 563–576. Springer, 2010.

[36] Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, and Seung-Ik Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2020.

[37] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.