Video Anomaly Detection via Sequentially Learning Multiple Pretext Tasks – Supplementary Material –

Chenrui Shi[1,2], Che Sun[2,1]*, Yuwei Wu[1], Yunde Jia[2]
[1]Beijing Key Laboratory of Intelligent Information Technology,
School of Computer Science & Technology, Beijing Institute of Technology, China
[2]Guangdong Laboratory of Machine Perception and Intelligent Computing,
Shenzhen MSU-BIT University, China
{shichenrui,sunche,wuyuwei,jiayunde}@bit.edu.cn

1. Overview

In this supplementary material, we present following additional information:

- We introduce some technical details of our method in Sec. 2.
- We showcase some failure object detection results in crowded scenes in Sec. 3.
- We demonstrate improvements of our model after leveraging optical-flow images in Sec. 4.
- We report the results of the ablation study in ShanghaiTech dataset in Sec. 5.
- We show more visualizations in Sec. 6.

2. Technical Details

2.1. Model Architecture

The detailed network architecture of our model is shown in Tab. 1 and Tab. 2.

2.2. Hyper-parameters

The initialized learning rates of our method in different datasets from each phase are shown in Tab. 3. We adopt Adam optimizer[2] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ to optimize our model and the learning rate decayed after every 50 epochs. In each training phase, minimizing the weighted sum of losses $\mathcal{L}$ is the training objective, given by

$$\mathcal{L} = \omega_{\text{pre}}\mathcal{L}_{\text{pre}} + \omega_{\text{grd}}\mathcal{L}_{\text{grd}} + \omega_{\text{rec}}\mathcal{L}_{\text{rec}} + \omega_{\text{cls}}\mathcal{L}_{\text{cls}} + \omega_{\text{enc}}\mathcal{L}_{\text{enc}}. \tag{1}$$

Weights for different losses in different datasets from each phase are shown in Tab. 4.

*Corresponding author: Che Sun

Layer		Output Shape	Param Nums
inconv	double conv	[-1, 32, 32, 32]	12864
block1	conv2d	[-1, 64, 16, 16]	18496
	double conv	[-1, 64, 16, 16]	74112
block2	conv2d	[-1, 128, 8, 8]	73856
	double conv	[-1, 128, 8, 8]	295680
block3	conv2d	[-1, 256, 4, 4]	296168
	double conv	[-1, 256, 4, 4]	1181184
Total trainable parameters			1951360

Table 1. Network architecture of our encoder. "conv2d" denotes 2D-convolution layer and "double conv" denotes two consecutive convolution layers. Numbers in the output shape $[b, w, h, c]$ mean batchsize, width, height and channel respectively.

Layer		Output Shape	Param Nums
block1	deconv2d	[-1, 128, 8, 8]	295040
	d-deconv	[-1, 128, 8, 8]	295040
block2	deconv2d	[-1, 64, 16, 16]	73792
	d-deconv	[-1, 64, 16, 16]	74112
block3	deconv2d	[-1, 32, 32, 32]	18464
	d-deconv	[-1, 32, 32, 32]	18624
PreHead	conv2d	[-1, 3, 32, 32]	99
RecHead	conv2d	[-1, 3, 32, 32]	396

Table 2. Network architecture of the prediction head and the reconstruction head. "deconv2d" denotes 2D-deconvolution layer and "d-deconv" denotes two consecutive deconvolution layers.

2.3. Computation Time

We list the training time of our method in Tab. 5. Additional information about testing is also shown in Tab. 6. We conduct experiments on an NVIDIA GeForce GTX 1080 Ti

	Ped2	Avenue	ShTech
Phase1	0.0001	0.0001	0.00001
Phase2	0.0001	0.001	0.0001
Phase3	0.001	0.001	0.0001

Table 3. Learning rate initialization in our method.

Dataset	Phase	ω_{pre}	ω_{grd}	ω_{rec}	ω_{cls}	ω_{enc}
Ped	1	1	0	-	-	-
	2	1	0	1	-	-
	3	1	0	1	1	1
Avenue	1	1	0.0001	-	-	-
	2	1	0	0.01	-	-
	3	1	0	0.01	0.001	0.001
ShTech	1	1	1	-	-	-
	2	1	0.01	0.001	-	-
	3	1	0.01	0.001	0.001	0.001

Table 4. Loss weights from each phase in different datasets.

	Ped2	Avenue	ShTech
Phase1	0.5	1.8	33.0
Phase2	0.6	6.0	40.0
Phase3	1.0	8.0	50.0

Table 5. Training time (GPU hour) of our method.

	Ped2	Avenue	ShTech
ObjectNum	34.3k	105.1k	206.5k
FrameNum	1.9k	15.2k	40.4k
Time (FPS)	122.6	127.0	72.4

Table 6. Testing information about our method. "ObjectNum" denotes number of detected salient objects. "FrameNum" denotes number of testing frames.

and an Intel(R) Core(TM) i7-7800X CPU @ 3.50GHz.

3. Failure Cases of Object Detector

Ped2[6] dataset contains lots of crowded scenes, which could compromise the detector's performance. We show some failure cases in Fig. 1. These failure cases can be roughly divided into three categories, half objects in Fig. 1a, blurred objects in Fig. 1b and multiple objects in Fig. 1c. These wrongly detected regions could adversely affect our model's performance. Avenue[4] and ShanghaiTech[5] datasets are less affected by this. We show some detection results in Fig. 2.

We increase the confidence threshold of the detector from 0.5 to 0.75 to reduce the amount of these bad cases and retrain our model in Ped2 dataset. We report the results in Tab. 7. Our method is able to achieve significant performance gains, 5.18%, in phase1. This shows that contaminated detection results could negatively affect model's performance. We will adopt other detector options and explore new tasks for handling contaminated data during training in the future.

Conf_thr	0.5	0.75	Δ
Phase1	91.34	95.42	+4.08

Table 7. Comparisons of AUC (%) performance in Ped2 dataset after increasing the confidence threshold of the detector. "Conf_thr" denotes the value of confidence threshold.

4. Experiments

Leveraging optical-flow images is helpful for models to detect anomalies, since most anomalies are time-relevant. In order to show how optical-flow images boost model's performance, we conduct a primitive experiment to use optical-flow images in our method in Ped2 dataset. We make minor revisions to our original sequential learning curriculum:

1. We change the frame prediction task to optical-flow prediction task, where we predict the next frame's optical-flow images from RGB frames.

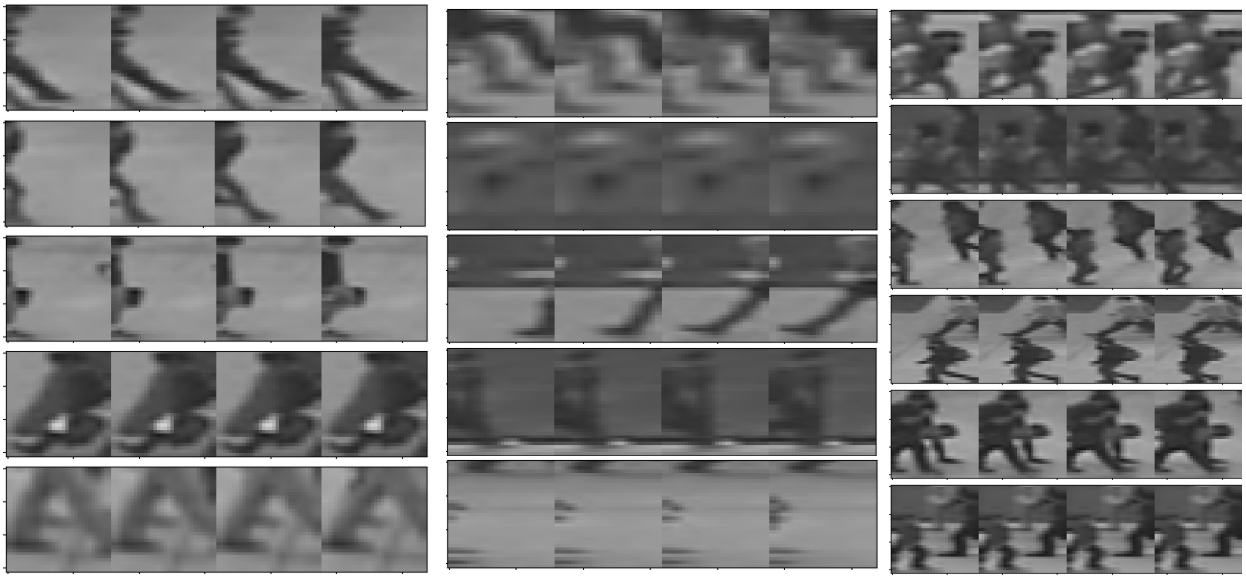
2. We don't use the frame reconstruction task anymore, since the optical-flow prediction task is able to capture temporal and spatial normality.

3. We shorten our learning curriculum and only include the original Phase1 and Phase3 of the curriculum in absence of the frame reconstruction task.

We report the results in Tab. 8. The performance of our method after using optical-flow images increased by 8.12% and 1.64% in Phase1 and Phase3 respectively. This demonstrates that optical-flow images are helpful to detect anomalies. The performance of our method after using optical-flow images outperformed BDPN[1] and achieved a comparable results with HF2VAD[3], 99.02% versus 99.3%. The minor revision made to the curriculum is but a simple replacement of the frame prediction task and is nowhere near an ideal way of using optical-flow images. Nonetheless, the comparable results demonstrate the effectiveness of our method. We will improve our method by introducing more suitable tasks for optical-flow images and designing better curriculums in the future.

5. Ablation Study

We report the results of the ablation studies in ShanghaiTech dataset in Tab. 9.



(a) Half objects. (b) Blurred objects. (c) Multiple objects.

Figure 1. Failure cases of object detector.

(a) Detection results in ShanghaiTech. (b) Detection results in Avenue.

Figure 2. Object detection results in ShanghaiTech and Avenue datasets.

	our	our+of	Δ
Phase1	91.34	98.36	7.02
Phase2	95.17	-	-
Phase3	97.38	99.02	1.64

Table 8. Comparisons of AUC (%) performance after each training phase between our method "*our*" and our method with optical-flow images "*our+of*" in Ped2 dataset.

6. Visualization

We show more visualizations of our method in Fig. 3.

References

[1] Chengwei Chen, Yuan Xie, Shaohui Lin, Angela Yao, Guannan Jiang, Wei Zhang, Yanyun Qu, Ruizhi Qiao, Bo Ren, and Lizhuang Ma. Comprehensive regularization in a bi-directional predictive network for video anomaly detection. In *Proceedings of the American Association for Artificial Intelligence*, pages 1–9, 2022. 2

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 1

[3] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13588–13597, 2021. 2

[4] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE Inter-*

Dataset	Sequential						Simultaneous
	The Learning Order			AUC			AUC
	Phase1	Phase2	Phase3	Phase1	Phase2	Phase3	
ShTech	Pre	+Rec	+Cls	74.39	78.69	**78.77**	70.06
		+Cls	+Rec		72.87	71.71	71.01
	Rec	+Pre	+Cls	65.52	71.43	71.91	69.72
		+Cls	+Pre		67.15	71.82	67.85
	Cls	+Pre	+Rec	-	65.98	67.51	69.21
		+Rec	+Pre		64.95	68.33	68.00

Table 9. AUC (%) performances of models trained sequentially with different learning orders, and models trained simultaneously with different weight assignments in ShanghaiTech dataset.

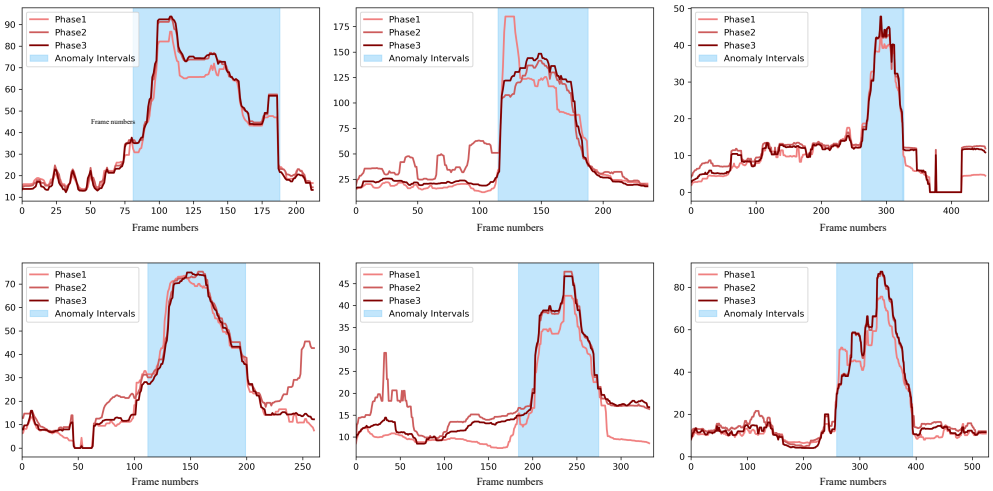


Figure 3. Curves of predicted anomaly scores from each phase from ShanghaiTech dataset.

national Conference on Computer Vision, pages 2720–2727, 2013. 2

[5] 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. 2

[6] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *IEEE computer society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981, 2010. 2