MIST: Multiple Instance Self-Training Framework for Video Anomaly Detection (Supplementary Materials)

Jia-Chang Feng[1,3,4], Fa-Ting Hong[1,3], and Wei-Shi Zheng[1,2,3*]

[1] School of Computer Science and Engineering, Sun Yat-Sen University

[2] Peng Cheng Laboratory, Shenzhen, China

[3] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China

[4] Pazhou Lab, Guangzhou, China

fengjch8@mail2.sysu.edu.cn, hongft3@mail2.sysu.edu.cn, wszheng@ieee.org

a) Samples of Kinetices-400

b) Samples of UCF-Crime

Figure 1: Samples of action recognition dataset Kinetics-400 [4] and video anomaly dataset UCF-Crime [7]. The red boxes are the anomalous regions in frames and their corresponding enlarged images.

1. Comparisons of Action Recognition Datasets and Anomaly Detection Datasets

As Figure 1 shown, the samples from Kinetics-400 [4] are actor-centered while the samples from UCF-Crime [7] are not [1, 2]. Additionally, the anomalies in frames are usually small and low-resolution. These situations indicate the domain gap between the two kinds of datasets. In this work, we propose MIST to minimize the domain gap by training both feature encoder and classifier in a two stage self-training scheme.

*Corresponding author

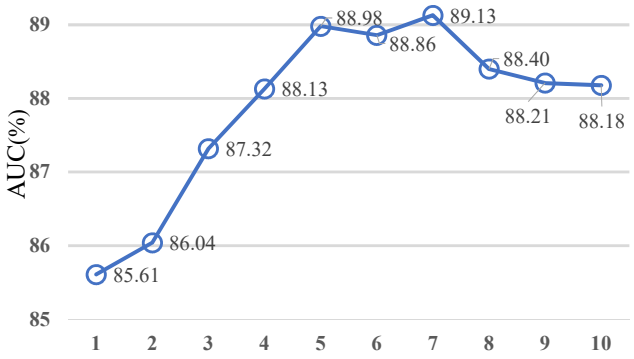

Figure 2: The effect of T for a fixed number of sub-bag 32 on ShanghaiTech dataset with $I3D^{RGB}$ features.

2. Details of Pseudo Label Generation

2.1. Feature Extraction and Sampling

We deploy a vanilla feature encoder, *i.e.* **C3D** [8] pretrained on Sport-1M [3] or **I3D** pretrained on Kinetics-400 [4] to extract features for generator training. We densely sample 16 frames per clip most of the times but 12 frames per clip for **I3D** on UCF-Crime. After extracting the features, sparse continuous sampling is applied to sample the $L \cdot T$ clips to form bags of features $\overline{B}$. Then, $\mathcal{L}_{MIL}$ is deployed to optimize the generator. Specifically, we follow [7] to select $L = 32$. As for T, we choose $T = 3$ for UCF-Crime and $T = 7$ for ShanghaiTech. We have shown the selection of $\mathcal{K}$ on ShanghaiTech with $I3D^{RGB}$ features in Figure 2. Additionally, λ is set as 0.01. 40 normal and 40 abnormal videos are randomly sampled as a batch when training.

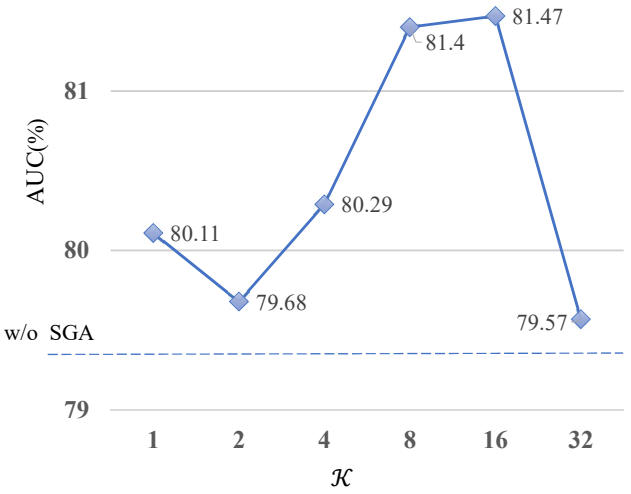

Figure 3: Variations of AUC for different values of multiple detector $\mathcal{K}$ with **C3D** on UCF-Crime dataset. The dotted line is the result of MIST training without self-guided attention module.

2.2. Pseudo Label Refinement

The trained generator predicts clip-level scores $S^a = \{s_i^a\}_{i=1}^N$ for all abnormal videos in the training set. Temporal moving average filter with kernel size $k = 5$ and min-max normalization are deployed to refine the anomaly scores into $\hat{Y} = \{\hat{y}_i^a\}_{i=1}^N$.

3. Details of Feature Encoder Finetuning

3.1. Implementation of Self-Guided Attention Module

As shown in Figure 4 of the submission, our proposed self-guided attention module includes 3 encoding units, namely $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$. All of these encoding units are constructed by convolutional layers. Let C represents the number of channel of $\mathcal{M}_{b-4}$. $\mathcal{F}_1$ consists of a $3 \times 3 \times 3 \times C$ 3DConv layer with the stride of 2 and a $1 \times 1 \times 1 \times 2\mathcal{K}$ 3DConv layer, which are both activated by ReLU function; $\mathcal{F}_2$ is a $1 \times 1 \times 1 \times 1$ 3DConv layer activated by Sigmoid function; $\mathcal{F}_3$ is a $1 \times 1 \times 1 \times 2\mathcal{K}$ 3DConv layer. Then, the attention map $\mathcal{A}$ is calculated as follows:

$$\mathcal{A} = \mathcal{F}_2(\mathcal{F}_1(\mathcal{M}_{b-4})), \tag{1}$$

while the guided classification prediction $\hat{p}$ is an aggregation results from $\mathcal{M}$, which is calculated below:

$$\mathcal{M} = \mathcal{F}_3(\mathcal{F}_1(\mathcal{M}_{b-4})). \tag{2}$$

Specifically, $\hat{p}$ is transformed from $\mathcal{M}$ via spatiaotemporal average pooling Π and class-specific channel-wise average pooling Φ:

$$\hat{p} = \Phi(\Pi(\mathcal{M})), \tag{3}$$

which is further optimized by $\mathcal{L}_2$ to guide the optimization of class-wise discriminative feature map $\mathcal{M}_{b-4}^*$ and then strengthen the attention map generation indirectly.

3.2. Implementation of E_{SGA} Finetuning

For UCF-Crime, we sample 16 abnormal videos and 16 normal videos per batch, and uniformly sample 3 clips from each video. For ShanghaiTech, we sample 10 abnormal videos and 10 normal videos per batch. The training process finishes in 300 epochs. Specifically, at the begining of finetuning, we conduct *warm-up* for 5 epochs. Since only a few clips of the abnormal video are anomalous, there exists a class-imbalance problem, especially for **I3D**. We introduce class-reweighting to cross-entropy loss as class-weighted cross-entropy loss $\mathcal{L}_w$:

$$\mathcal{L}_w = -w_0 y \log p - w_1(1-y)\log(1-p), \tag{4}$$

where w_0 and w_1 are class weights for abnormal and normal class, respectively. Specifically, $\mathcal{L}_1$ and $\mathcal{L}_2$ are adopted the same kind of loss function $\mathcal{L}_w$ We adopt $w_0 = 1.2$ and $w_1 = 0.8$ for UCF-Crime, while $w_0 = 0.8$ and $w_1 = 0.65$ for ShanghaiTech.

In the left of Figure 3, we report the AUC of STSA with different $\mathcal{K}$. The performance goes up as the $\mathcal{K}$ get larger and reaches the top with $\mathcal{K}$ of 8 or 16. When the value getting even larger, it seems to be overfitting and get worse. Considering a trade-off between the efficiency with effectiveness, we set $\mathcal{K} = 8$ in our framework for all other experiments.

After finetuning, we acquire a task-specific feature encoder E_{SGA}. E_{SGA} outperforms state-of-the-art encoder-based method Zhong *et al.* [11], which is shown in Figure 4 in detail. Moreover, E_{SGA} can focus on the anomalous regions in frames, which is shown in Figure 5. As the left 5 columns of the figure shown, self-guided attention module help the feature encoder in focusing the anomalous regions. We have also listed the failure on the right 2 columns of the figure, the results from too small size of anomaly regions.

4. More Experimental Results

4.1. Speed and Computational Complexity

Model	#Params	Speed (FPS)	FLOPs (MAC)
MIST-I3D	31M	324.46	45.68G
MIST-C3D	85M	197.10	39.26G
Zhong-C3D[11]	78M	130.04	386.2G

Table 1: Speed and computational complexity comparisons with Zhong *et al.* [11].

There are four 1080Ti GPUs for stage 2 but one 1080Ti GPU for stage 1 and validation. In C3D (I3D) based model,

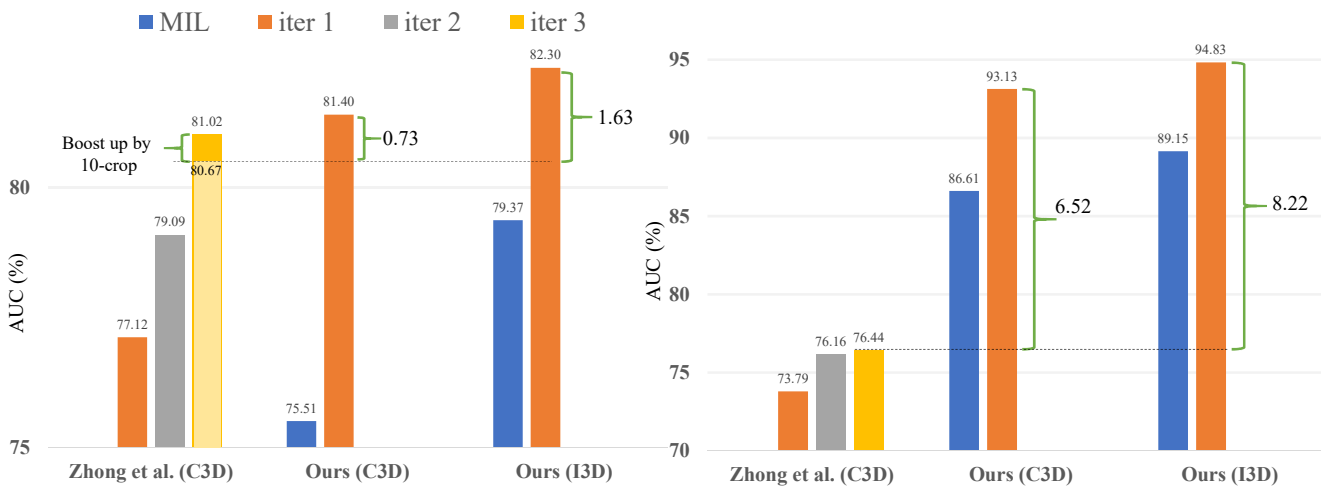

Figure 4: Quantitative Comparisons with state-of-the-art encoder-based method Zhong *et al.* [11] on UCF-Crime and ShanghaiTech.

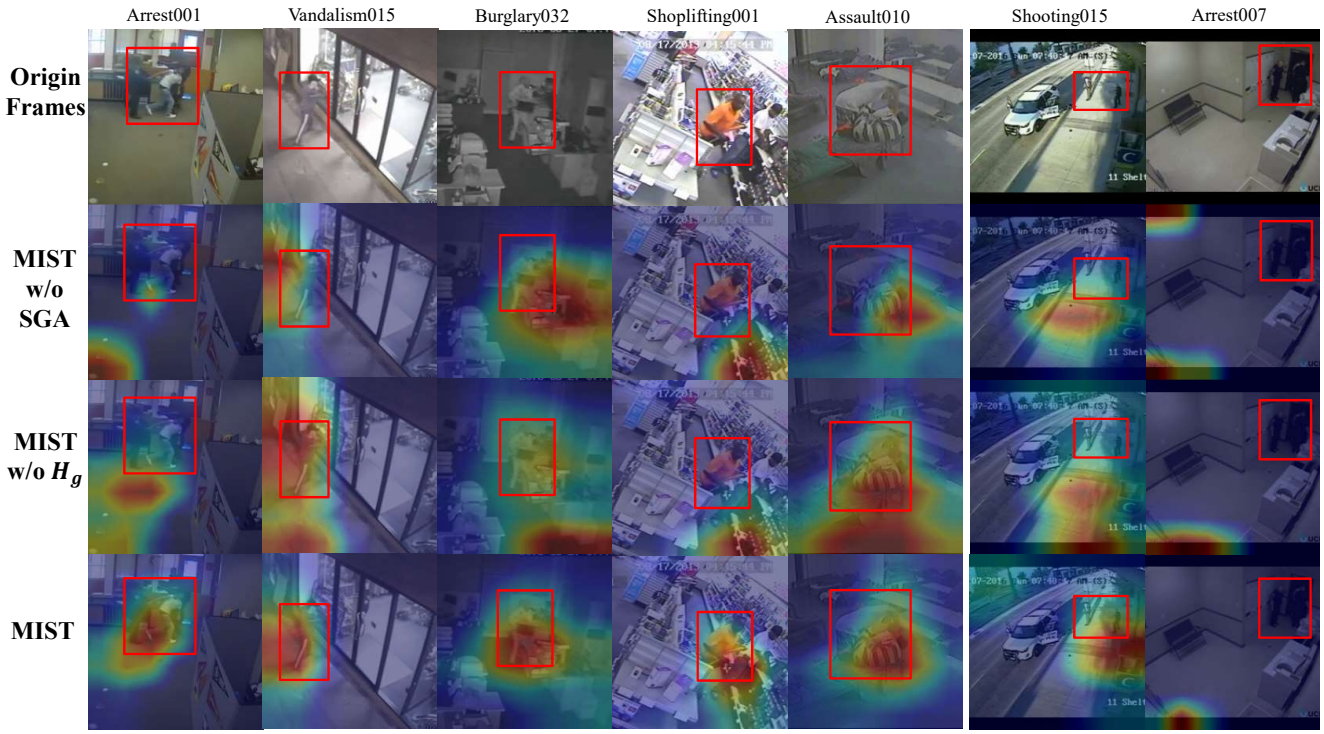

Figure 5: More spatial anomaly activation maps visualization on UCF-Crime. The left 5 columns of the graphs are the successful results while the right 2 columns are the failures. The red boxes are the ground-truth spatial annotations [5].

#Params are 85 M (31 M), the FLOPs are 39.26 G (45.68 G) and the speed is 197.10 FPS (324.46 FPS). Compared to Zhong *et al.* that adopt *10-crop* testing time augmentation, our method is much faster but costs much lower computational complexity as shown in Table 1.

4.2. More Quantitative Comparisons

We show more quantitative comparisons with Zhong *et al.* [11] on UCF-Crime and ShanghaiTech on Figure 4. We observe a huge improvement in ShanghaiTech. As for UCF-Crime our method still do much better when compared fairly without using *10-Crop*. Moreover, our method does

much better on iter 1 as MIST does not need iterative optimization.

4.3. More Spatial Visualization

We also present more spatial visualization in Figure 5. We observe that MIST performs better than those without SGA or H_g. The left two columns are the failure case where the front ground is extremely small and vague to be detected.

5. Discussions of the Formulation

5.1. Label Noise Learning vs MIST

Zhong *et al.* [11] treats weakly supervised video anomaly detection as a label noise learning task. However, the extreme label noise results from assigning video-level labels to each clip. In contrast, MIST offers pseudo labels with lower noise via multiple instance generator, which is more efficient. Additionally, MIST can further co-operate with label noise learning methods to refine pseudo labels iteratively and train a more powerful feature encoder.

Model	Before (%)	After (%)	Gain(%)
MIST-C3D	58.66	67.14	+8.48
MIST-I3D	63.63	73.37	+9.74

Table 2: Performance comparisons of before and after refinement on ShanghaiTech in term of AUC scores of anomaly videos.

In contrast to Zhong *et al.* that reduces the noise via a specific module, *i.e.* GCN-based label noise cleaner, we resist label noise via post procession likes min-max norm and temporal smoothing. As shown in Table 2, we conduct these two types of refinement are do a great help in removing label noise. Moreover, we also use large a batch size with the help of gradient accumulation to reduce the label noise [6].

5.2. 2D Feature Encoder vs 3D Feature Encoder

We also conduct experiment on 2D feature encoder the RGB branch of TSN [10] but fail. Similar result is also reported in [5]. Since the RGB branch of TSN operates only on a single frame, it fails in catching the motion to represent temporal information. Instead, we deploy two popular 3D spatiotemporal feature encoders, *i.e.* **C3D** and **I3D**, whose results well-verified the capacity of MIST.

5.3. Fine-Grained vs Coarse-Grained and Online vs Offline

Our method focuses on online fine-grained anomaly detection. Previous works follow Sultani *et al.* [7] to perform anomaly detection in a coarse-grained manner. However, in the real world, we expect anomaly detection can be applied for streaming surveillance videos to detect anomalies precisely and quickly, while the methods in coarse-grained do not meet the requirement. Some work like Ullah *et al.* [9] performs anomaly detection in an offline manner based on an external assumption as complete observation of the testing videos. As discussed above, it also violates the expectation for detection on streaming video.

References

[1] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *Adv. Neural Inform. Process. Syst.*, 2019.

[2] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. 2020.

[3] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[5] Kun Liu and Huadong Ma. Exploring background-bias for anomaly detection in surveillance videos. In *ACM Int. Conf. Multimedia*, 2019.

[6] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[7] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[8] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Int. Conf. Comput. Vis.*, 2015.

[9] Waseem Ullah, Amin Ullah, Ijaz Ul Haq, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Cnn features with bi-directional lstm for real-time anomaly detection in surveillance networks. *Multimedia Tools and Applications*, pages 1–17, 2020.

[10] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(11):2740–2755, 2018.

[11] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.