Learning Normal Dynamics in Videos with Meta Prototype Network Supplemental Materials

Hui Lv[1], Chen Chen[2], Zhen Cui[1,*], Chunyan Xu[1], Yong Li[1], Jian Yang[1]

[2]PCALab, Nanjing University of Science and Technology, [2]University of North Carolina at Charlotte

{hubrthui, zhen.cui, cyx, yong.li, csjyang}@njust.edu.cn, chen.chen@uncc.edu

1. Visualization of anomaly score curves in test videos

In Fig. 1, we visualize some examples of anomaly score curves on test videos. The K-shot models are meta-trained on Shanghai Tech dataset and applied on test videos of Ped2 [1], Ped1 [1] and Avenue [2] in (c) and (d) columns, compared with models trained and tested on the same datasets mentioned above. As we can see, the performance of the scene-adapted models with meta-training (K $=10$) is superior than the baseline (K $=0$) without the adaption process. For example, the anomaly scores (K $=0$) among the normal temporal region in the 1st row (c) column (before red box of ground-truth anomaly) retain a high value before adaption, and the scores drop after adaption as in (d) column. Similar circumstances can be found in other rows.

In addition, the few-shot models almost catch up with models trained in the unsupervised setting in the second, fourth and sixth rows. However, there are still room for improving our algorithm, such as enhancing the consistence of anomaly scores from continuous anomalous frames and amplifying the score margins between normal and anomalous frames.

2. Meta-training Pseudo Code

For further details of the meta-training, we summarize the entire learning algorithm in Algorithm 1.

Algorithm 1: Meta-training for few-shot scene-adaptive anomaly detection algorithm

Input: Pre-trained AE model $E_\eta(x)$, Randomly initialized θ_0 and α, training dataset $\mathcal{D}$

Output: θ_0^* and α^*

while *not converged* **do**
- Initialize grad$_{\theta_0}$, grad$_\alpha$ to zero vector;
- **for** *each eposide in a mini-batch* **do**
  - Sample a training example $j, k \sim p(\mathcal{D})$;
  - $y_j{}' = f_\theta(E_\eta(x_j))$;
  - $\hat{\theta}_0 = \theta_0 - \alpha \odot \nabla_{\theta_0^t} L(y_j, y_j{}'; \theta_0)$;
  - $y_k{}' = f_{\hat{\theta}}(E_\eta(x_j))$;
  - grad$_{\theta_0}$ $=$ grad$_{\theta_0}$ $+ \nabla_{\theta_0} L(y_k, y_k{}')$;
  - grad$_\alpha$ $=$ grad$_\alpha$ $+ \nabla_\alpha L(y_k, y_k{}')$;
- **end**
- Update θ_0: $\theta_0 =$ Optimizer(θ_0, grad$_{\theta_0}$);
- Update α: $\alpha =$ Optimizer(α, grad$_\alpha$);

end

References

[1] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *TPAMI*, 2013. 1

[2] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013. 1

*Corresponding authors

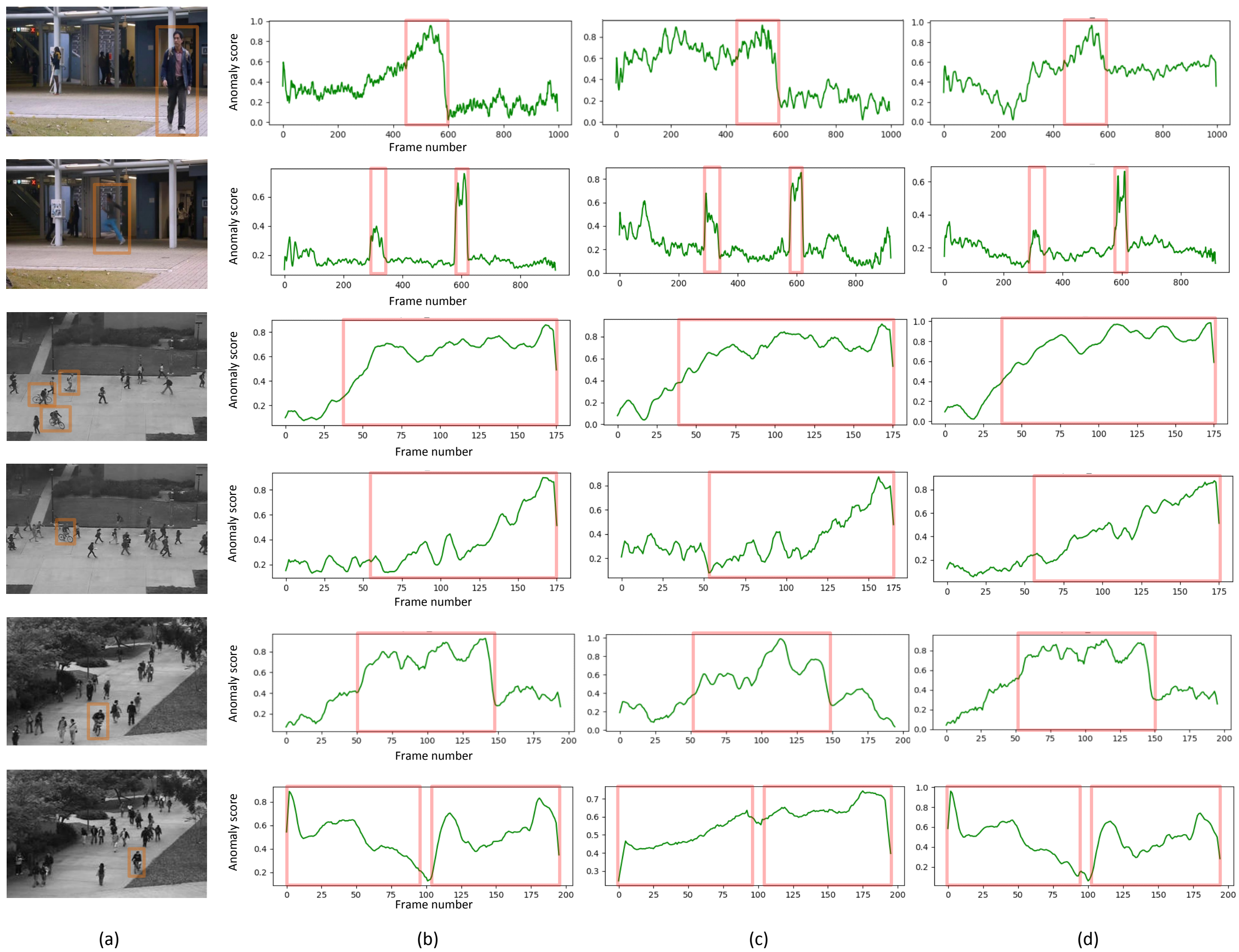

Figure 1: Visualization of some examples of test videos. The groups of pictures in different columns denote (a) anomalous frame, (b) anomaly scores under the unsupervised setting, (c) anomaly scores under the few-shot setting (K $=$ 0), (d) anomaly scores under the few-shot setting (K $=$ 10). The green curves denote the evolution of anomaly scores. The orange and red boxes represent the ground-truth anomalous regions in the frames and temporal ground-truth anomaly locations of videos, respectively. In each figure of the anomaly score curve, the x-axis denotes the frame number in a video sequence and the y-axis denotes the scalar of anomaly score.