UNSUPERVISED AND ADAPTIVE PERIMETER INTRUSION DETECTOR

Devashish Lohani, Carlos F Crispim-Junior, Quentin Barthélemy, Sarah Bertrand, Lionel Robinault, Laure Tougne

UNSUPERVISED AND ADAPTIVE PERIMETER INTRUSION DETECTOR

Devashish Lohani[⋆†] *Carlos Crispim-Junior*[⋆] *Quentin Barthélemy*[†]
Sarah Bertrand[†] Lionel Robinault[⋆†] Laure Tougne[⋆]

[⋆] Univ Lyon, Lyon 2, LIRIS, F-69676 Lyon, France, firstname.lastname@liris.cnrs.fr
[†] FOXSTREAM, Vaulx-en-Velin, France, f.lastname@foxstream.fr

ABSTRACT

Perimeter intrusion detection (PID) deals with the detection of intruders displacing in a protected perimeter. In the video surveillance domain, deep learning has shown tremendous progresses. Existing deep learning based PID systems (PIDS) are supervised and thus require a lot of annotated data. However, since intrusions are rare events, there are very few positives in datasets, thus making them highly imbalanced. Furthermore, a PIDS must adapt to varying real-life scene dynamics, like weather, light, environmental conditions, *etc*. To address these issues, we propose an autoencoder-based, end-to-end trainable, unsupervised PIDS with a module that can adapt to long-term variations in scene dynamics. Our results show competitive performance of the proposed system on the standard i-LIDS dataset.

***Index Terms*—** Perimeter intrusion detection, deep learning, unsupervised, adaptive, i-LIDS dataset.

1. INTRODUCTION

The task of perimeter intrusion detection (PID) consists in detecting unauthorized objects entering in a protected area [1, 2]. These objects are defined by users and can vary from one site to another, e.g., for one site cars can be intruders while not for others. Furthermore, they must displace on the site, e.g., the cars parked on a site cannot be treated as an intrusion while an incoming moving car is an intrusion. The movement of trees and animals, changing weather and lighting conditions further makes this task difficult.

In the past few years, deep learning has shown tremendous achievements on video surveillance tasks like object detection [3], tracking [4], anomaly detection [5], *etc*. Like other tasks, deep learning also positively influenced the PID task [6, 7, 8]. Most deep learning based PIDS rely on annotated intrusion classes [2, 6, 7]. However, since intrusions occur very rarely, recorded videos mostly contain non-intrusion frames. In other words, PID datasets are highly imbalanced [8] with very few true positives, *i.e.*, intrusions. To work around these machine learning issues, some works [6, 7] identify classes of intruders and use a pre-trained object detector to detect intrusions. These supervised approaches assume that all potential intrusion classes are known, and that they can be detected by pre-trained object detectors. Other works take non-intrusion frames from videos and learn normality from them [9, 8], acting like a one-class classifier, modelling normality and detecting abnormal frames as intrusion. These unsupervised approaches do not make any assumptions on the intrusion classes. Lohani *et al.* [8] introduced an autoencoder to learn normality from non-intrusion videos and detect intrusions by thresholding reconstruction error. However, they do not propose a strategy to select the threshold. This inhibits the real-life deployment of the PIDS because we must fix a threshold to raise alarms.

Monitoring a site continuously for days or weeks, accounts to changing weather, light, and environmental conditions. The PIDS must adapt itself to these conditions in order to protect the site efficiently. One solution, dominantly used for the task of video anomaly detection [5, 9], consists in rescaling values towards a known interval, where a fixed threshold can be chosen. This strategy works well on very short length videos (less than 1 minute) with still scene dynamics; however, it struggles on long videos as it is not sufficiently adaptive when scene dynamics vary considerably.

Our main contributions are summarized as follows: (i) we design an unsupervised, end-to-end trainable, 3D convolutional autoencoder architecture; (ii) we provide an adaptive thresholding strategy that can adapt to long-term variations in scene dynamics; (iii) we analyze our approach and compare it on a standard dataset. In a nutshell, this work validates a new deployable unsupervised perimeter intrusion detection pipeline, available here: `https://gitlab.liris.cnrs.fr/dlohani/pyPID`.

The article is organized as follows: Section 2 highlights related work, Section 3 describes details of our method, Sections 4 and 5 presents experiments and results for different PIDS, and Section 6 concludes this work.

2. RELATED WORK

Detecting intrusion is a crucial task in intelligent surveillance systems [10]. Traditionally, this task was addressed by detecting a moving object using background modeling methods

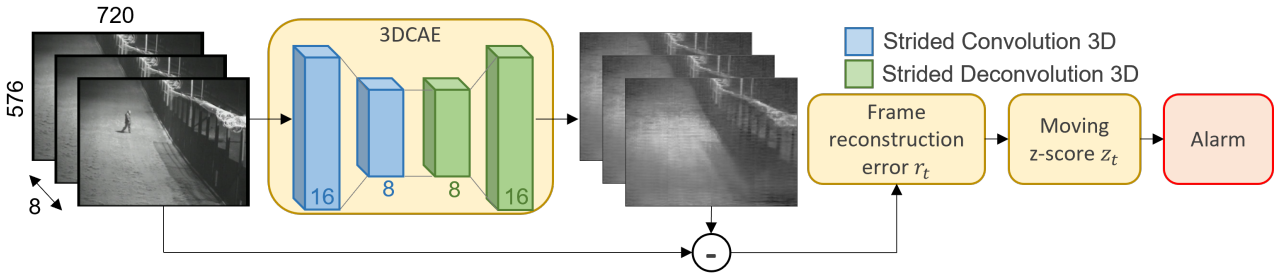

Fig. 1. Overall schema of proposed PIDS: input window is reconstructed after going through a 4 layered 3D convolutional autoencoder. The reconstruction error is fed to a moving z-score module where it is thresholded to raise an alarm.

[11] like frame differencing [12] or mixtures of Gaussians [2], tracking it using Kalman or particle filtering [13], and raising an alarm using some fixed rules [14].

Recently, supervised deep learning has been used for PID. Kim *et al.* [6] proposed a system where background modeling was used for movement detection and a convolutional neural network for classifying moving objects as intruders or not. Nayak *et al.* [7] used a pre-trained YOLOv2 object detector to detect potential intruders and track them using the simple online and real-time tracking [15]. These supervised approaches assume that the intrusion object classes are known *a priori*. Furthermore, they rely on object detectors trained on datasets like COCO [16] where objects are not representative for intrusion task, *e.g.,* nobody crawls in COCO images whereas a real intruder might.

Alternatively, unsupervised deep learning uses the abundant non-intrusion frames of the video for PID. Lohani *et al.* [8] proposed a deep spatio-temporal convolutional autoencoder which learns "normality" from non-intrusion frames. While testing, frames having high reconstruction errors are considered as intrusions. They however, did not propose a strategy to choose the threshold for raising alarms, which is essential for real-life PIDS deployment.

Since each video can record different weather, luminosity and climate conditions, thresholding the raw reconstruction error directly does not work well [17]. To resolve this issue, most works rescale the frame reconstruction error for each video [5, 17], usually using the min-max (MM) rescaling that forces the error values to be in the [0,1] interval. The underlying assumption is that the maximum value must be abnormal (intrusion) and minimum value must be normal [9]. This method is offline, since it requires the complete video to apply rescaling, and it can fail if max or min values are outliers. It can work well in short videos where there is no big change in scene dynamics. But in the case of PID, videos are long with continuously changing scene dynamics.

In this work, we address all these issues, proposing an unsupervised intrusion detector, where an adaptive thresholding strategy is updated online with the scene dynamics.

3. PROPOSED PIDS

In our PIDS (Fig. 1), we propose a new 3D convolutional autoencoder (3DCAE) to learn normality, coupled with an adaptive thresholding strategy to detect intrusions.

3.1. 3D convolutional autoencoder

The proposed autoencoder takes a video window I (a volume of time x width x height) as input and outputs a reconstructed window O. The idea is to minimize the error between these windows so that the autoencoder learns representative "normal" spatio-temporal features from non-intrusion videos.

Each encoder layer consists of a strided 3D convolution while each decoder layer uses a strided 3D deconvolution [18], with kernel size of 5x3x3 and stride 2 in both cases. Encoder (*resp.* decoder) is composed of two layers containing (16, 8) (*resp.* (8, 16)) filters, with a ReLU activation between each layer. The output layer consists of a 3D convolution with stride 1, followed by a *tanh* activation. A dropout layer with dropout probability of 0.25 is applied after the first layer. Overall, we have a light architecture with only 15, 889 parameters (1011 MB size).

The proposed 3DCAE is trained only on normal videos, *i.e.* without any intrusion, using Adadelta optimizer until the mean squared error loss converges.

3.2. Detecting intrusion

Once we have a trained model, we use it during testing phase with the hypothesis that the intrusion frames will be badly reconstructed, *i.e.*, with a high reconstruction error.

3.2.1. Frame level reconstruction error

Given i^{th} input window I_i and its reconstruction O_i, the window level reconstruction error (RE) is calculated as:

$$w_i = \frac{1}{N}\sum_{j=1}^{N} \|I_{i,j} - O_{i,j}\|_F^2 \ , \qquad (1)$$

where j in $I_{i,j}$ and $O_{i,j}$ corresponds to j^{th} frame of the window i, N is the temporal window size and $\|\cdot\|_F$ denotes the Frobenius norm.

For real-time application, we need a reconstruction error for each frame that the system encounters. Therefore, we need to extract per-frame error from the window level reconstruction error. The reconstruction error r_t for frame t is defined as: $r_t = w_{t-N+1}$ with $t \geq N$. This signifies that for each video, we have per-frame reconstruction error from N^{th} frame onwards.

3.2.2. Adaptive thresholding using moving z-score

We propose an adaptive mechanism which follows the reconstruction error along time and trigger an alarm as soon as it is deviated from the normal behavior and then continue adapting to the new values. We compute the moving z-score (MZ) of reconstruction errors in order to provide a temporal standardization of values [19]. Once standardized, values can be easily compared to a fixed threshold z_{th}.

Being initialized on first few frames, μ_t and σ_t are respectively the mean and standard deviation of reconstruction errors at frame t. For the frame $t+1$, z-score of reconstruction error r_{t+1} is computed as:

$$z_{t+1} = \frac{r_{t+1} - \mu_t}{\sigma_t} \ , \tag{2}$$

and the system raises an alarm when $z_{t+1} \geq z_{th}$. The moving mean and standard deviation are then updated as:

$$\begin{aligned} \mu_{t+1} &= \alpha \, r_{t+1} + (1-\alpha) \, \mu_t \\ \sigma_{t+1} &= \sqrt{\alpha \ (r_{t+1} - \mu_{t+1})^2 + (1-\alpha) \, \sigma_t^2} \ , \end{aligned} \tag{3}$$

where $\alpha \in [0, 1]$ defines the speed of the exponential update. This process with Eq. (2) and Eq. (3) is used for each new frame of the video.

4. EXPERIMENTS

4.1. Dataset and evaluation protocol

Methods are evaluated on the i-LIDS sterile zone dataset [20] as it is the only publicly available dataset for the PID task. It includes videos captured by two cameras (named as view 1 and view 2), with a frame resolution of 720x576. It consists of people approaching a fence in various ways like walking, running, crawling, *etc*. Furthermore, it captures different time of the day like dawn / day / night, weather conditions like cloudy / rainy / snowy and distractions like bats / birds / wild animals. The training set consists of intrusion and non-intrusion videos for both views, with 10 non-intrusion videos (29 min. average length) per view. The testing set contains 17 and 16 videos of view 1 and view 2, with 7 and 6 videos containing intrusions respectively (from 36 to 92 minutes in length).

Methods	View 1			View 2		
	Pre	Rec	F_1	Pre	Rec	F_1
Nayak [7]	0.26	0.92	0.41	0.28	0.94	0.43
Lohani [8]	0.57	0.49	0.53	0.61	0.48	0.54
Lohani [8] + MZ	0.87	0.82	0.84	0.82	0.77	0.79
Ours	0.87	0.83	**0.85**	0.85	0.74	**0.79**

Table 1. Results on the two views of the i-LIDS dataset. MZ stands for moving z-score thresholding.

We use the evaluation protocol proposed by i-LIDS dataset [20], defining a correct detection when the system raises at least one alarm within 10 seconds from the start of the intrusion. All alarms raised after this delay are defined as incorrect detections. Finally, to provide quantitative results, we use precision, recall and F_1 score.

4.2. Compared methods and implementation details

We compare with work of Nayak [7] and the upsampling variant of the autoencoder proposed by Lohani [8]. Both shared the source code publicly and we applied it on the i-LIDS dataset. Unfortunately, no other method provides source code or reports performance on the i-LIDS dataset.

We train and test on each view of the dataset separately. For both views of the dataset, we draw a protection perimeter following the fences. Our method and that of Lohani [8] are implemented similarly. Non-intrusion videos from training set of each view are used for training, using one Nvidia RTX 3090 GPU, with a batch size of 32. Each frame is converted to grayscale, pre-processed using histogram equalization and pixels were rescaled to [-1, 1]. For each video, input window is constructed using 8 frames with temporal stride of one frame, leading to an input of shape 8x720x576x1. For Lohani [8], threshold is applied on frame reconstruction error (RE). For our work, adaptive thresholding is done on moving z-score of RE. Initialized with reconstruction errors of first 10 frames, moving z-score is used with $z_{th} = 4.5$ and $\alpha = 0.01$. Thresholds in all cases were chosen from validation set while training.

For Nayak [7], we use person as intrusion class with the default detection threshold of 0.25. Their method provides a binary response of intruder or otherwise, and does not require training on the i-LIDS dataset.

5. RESULTS AND DISCUSSION

5.1. Results for perimeter intrusion detection

Overall results are presented on Table. 1. We can observe that method of Nayak [7] has the highest recall regardless of the dataset view. This means that it detects most of the intrusions. But we also observe a very poor precision for both views, signifying a large number of false alarms. Since their system de-

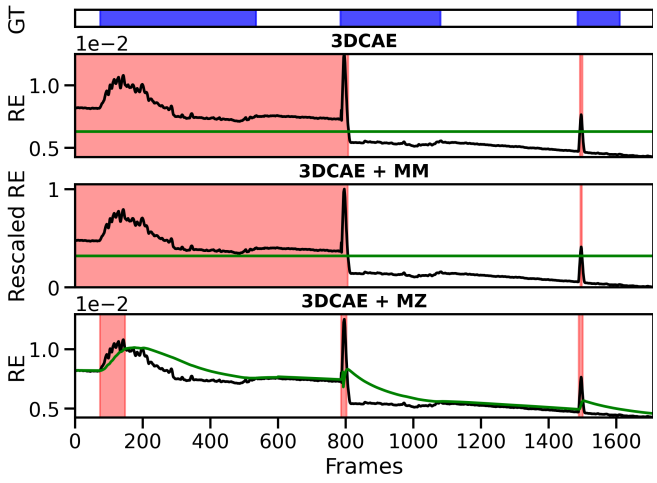

Fig. 2. Qualitative comparison of thresholding strategies on a video, with ground-truth (GT) in blue, thresholds in green and alarms in red. Fixed threshold on reconstruction error (RE) of 3DCAE (top); fixed threshold on min-max rescaled RE (middle); adaptive threshold by moving z-score of RE (bottom).

pends on object detection, each time an object track is lost, the object is re-detected and it raises an extra unnecessary alarm which is counted as a false positive. The method of Lohani [8] performs slightly better than that of Nayak [7] but still has an overall poor performance. Adding moving z-score boosts their results, signifying importance of adaptive strategy. Our proposed PIDS has a lower recall than that of Nayak [7], indicating it misses some intrusions. These missed intrusions are usually far from camera (accounting for few pixels in video frame), camouflaged with scene background and in low luminosity (*e.g.*, during night). But we have a high precision value regardless of the dataset view, signifying fewer false alarms (caused by birds and insects on camera). Each time we have a detection, our adaptive strategy adapts the new values and thus our system is ready for the next intrusion without re-detections. Our method and method of Lohani [8] with z-score have close performances but our method is 1.5x faster due to strided convolutions. Overall, our proposed unsupervised system has highest performance on both camera views.

5.2. Comparison of thresholding strategies

Fig. 2 shows impact of various thresholding strategies. We can observe that the RE varies a lot from one intrusion to another. Therefore, thresholding on it can lead to ambiguous results, e.g., false alarms before and after 1^{st} intrusion. After min-max rescaling, we obtain a similar conclusion as only the range of values have changed to [0,1], without any significant difference in threshold choosing strategy. We can see that the moving z-score adapts itself with the reconstruction error. It detects beginning of each intrusion and then adapts itself.

In Table 2, we provide quantitative comparison of thresholding strategies on proposed 3DCAE. We can observe that the direct outcome from the autoencoder leads to a poor performance. As expected, it is difficult to choose a threshold when the error varies due to scene dynamics. Rescaled reconstruction error using min-max (MM) scheme augments the overall result by only 4%. Since i-LIDS test-set videos are very long, the rescaling scheme did not work well. We can clearly observe that adding an adaptive thresholding with moving z-score (MZ) almost doubles the overall performance from F_1 of 0.49 to 0.85. These results strongly support our proposition that an adaptive component is necessary for the deployment of a PIDS in real-life scenes.

Methods	Prec	Rec	F_1
3DCAE	0.54	0.44	0.49
3DCAE + MM	0.48	0.59	0.53
3DCAE + MZ	**0.87**	**0.83**	**0.85**

Table 2. Quantitative comparison of thresholding strategies, on View 1 of i-LIDS dataset. 3DCAE stands for proposed autoencoder with frame reconstruction error, MM for min-max scaling, and MZ for moving z-score.

5.3. Discussion

Method of Nayak [7] is good for detections but it suffers from massive false alarms. Further their system relies on a limited number of pre-trained classes, implying if a new category of object appears, their system will fail to detect it. Our proposed autoencoder and that of Lohani [8] performed similarly, having exactly the same number of parameters, but our method was much faster both in training and testing. These autoencoder based methods struggle when the intrusion account for few frame pixels and is in low luminosity. Furthermore, they still lack semantics to differentiate between a human intruder with animals or birds.

Thresholding directly on reconstruction error of autoencoder leads to poor performance. The rescaling schemes also suffer with long length videos and so we found that an adaptive thresholding is essential for a PIDS.

6. CONCLUSION

In this work, we proposed an unsupervised perimeter intrusion detection system where a 3D convolutional autoencoder learned normality from non-intrusion videos while training, and detected intrusions with an adaptive thresholding using moving z-score. Our method is robust with the changing scene dynamics, allowing a real-life deployment of the PIDS. Experiments on the i-LIDS dataset showed that our approach outperformed recent approaches: supervised detector relying on limited classes and fixed threshold based unsupervised detector. In future work, we would like to further strengthen our model with attention modules.

7. REFERENCES

[1] G Aravamuthan, P Rajasekhar, RK Verma, SV Shrikhande, S Kar, and S Babu, "Physical intrusion detection system using stereo video analytics," in *CVIP*, 2020, pp. 173–182.

[2] JA Vijverberg, RTM Janssen, R de Zwart, and PHN de With, "Perimeter-intrusion event classification for on-line detection using multiple instance learning solving temporal ambiguities," in *ICIP*, 2014, pp. 2408–2412.

[3] Z Zou, Z Shi, Y Guo, and J Ye, "Object detection in 20 years: A survey," *arXiv:1905.05055*, 2019.

[4] G Ciaparrone, F L Sánchez, S Tabik, L Troiano, R Tagliaferri, and F Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[5] R Chalapathy and S Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[6] SH Kim, SC Lim, and DY Kim, "Intelligent intrusion detection system featuring a virtual fence, active intruder detection, classification, tracking, and action recognition," *Ann Nucl Energy*, vol. 112, pp. 845–855, 2018.

[7] R Nayak, MM Behera, UC Pati, and SK Das, "Video-based real-time intrusion detection system using deep-learning for smart city applications," in *ANTS*, 2019, pp. 1–6.

[8] D Lohani, C Crispim-Junior, Q Barthélemy, S Bertrand, L Robinault, and L Tougne, "Spatio-temporal convolutional autoencoders for perimeter intrusion detection," in *RRPR*, 2021, pp. 47–65.

[9] B R Kiran, D M Thomas, and R Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *J Imaging*, vol. 4, pp. 36, 2018.

[10] M Valera and S A Velastin, "Intelligent distributed surveillance systems: a review," *IEE Proc Vis Image Sign Process*, vol. 152, pp. 192–204, 2005.

[11] T Bouwmans, F El Baf, and B Vachon, "Background modeling using mixture of Gaussians for foreground detection - A survey," *Recent Patents on Computer Science*, vol. 1, pp. 219–237, 2008.

[12] N Buch and SA Velastin, "Human intrusion detection using texture classification in real-time," *Tracking Humans for the Evaluation of their Motion in Image Sequences*, p. 1, 2008.

[13] E Cermeño, A Pérez, and JA Sigüenza, "Intelligent video surveillance beyond robust background modeling," *Expert Syst Appl*, vol. 91, pp. 138–149, 2018.

[14] N Buch and SA Velastin, "Local feature saliency classifier for real-time intrusion monitoring," *Opt Eng*, vol. 53, pp. 073108, 2014.

[15] A Bewley, Z Ge, L Ott, F Ramos, and B Upcroft, "Simple online and realtime tracking," in *ICIP*, 2016, pp. 3464–3468.

[16] T-Y Lin, M Maire, S Belongie, J Hays, P Perona, D Ramanan, P Dollár, and C L Zitnick, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014, pp. 740–755.

[17] M Ribeiro, AE Lazzaretti, and HS Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognit Lett*, vol. 105, pp. 13–22, 2018.

[18] M D Zeiler, D Krishnan, G W Taylor, and R Fergus, "Deconvolutional networks," in *CVPR*, 2010, pp. 2528–2535.

[19] Q Barthélemy, L Mayaud, D Ojeda, and M Congedo, "The Riemannian potato field: a tool for online signal quality index of EEG," *IEEE Trans Neural Syst Rehabil Eng*, vol. 27, pp. 244–255, 2019.

[20] i-LIDS Team, "Imagery library for intelligent detection systems (i-LIDS); a standard for testing video based detection systems," in *IET Conference on Crime and Security*, 2006, pp. 445–448.