CH-004

Abnormal detection in nuclear security based on label-specific threshold-free mechanism and future frame prediction network

李 湛† 菅原 和将† 出町 和之†
Zhan Li Kazumasa Sugawara Kazuyuki Demachi

1. Introduction

Nuclear power plants (NPPs), which provide clean energy to our human society, are also exposed to serious risks caused by human malicious behaviors, such as personal and property safety, nuclear material theft and smuggling and also radiation leak. Especially after Fukushima Daiichi NPP accident happened in 2011, the vulnerable and weak parts inside NPP are exposed to public including people related to illegal and terrorism. Currently, to strengthen nuclear security and prevent NPPs from being destroyed by human malicious behaviors, a common flow chat of nuclear security measures is already proposed[1], as shown in Fig.1. It could be easily known that the nuclear security measures mainly contain the following four steps.

(1) *Deterrence*. Physical boundaries that are used for preventing illegal access to NPPs, such as wire nets.
(2) *Detection*. Specialized equipment as well as techniques to detect the existence of malicious behaviors.
(3) *Delay*. Measurements to prevent or delay illegal personnel from reaching sensitive areas of NPPs.
(4) *Response*. Remediation measures to alleviate the consequences caused by human malicious behaviors.

Among these four steps, the third one *Delay* and the fourth one *Response* are mainly related to post-processing measures while the malicious behaviors have been occurred at these two stages. At the same time, the first stage *Deterrence* is actually well developed in Japan, but the main defect is that the function of this stage is really limited and it could only be used for detection of relatively simple scenarios. Consequently, in this paper, the second step *Detection* should be the main focus, which is also the key step in Fig.1.

For the abnormal detection in nuclear security scenarios, our previous work[2] conducted preliminary research based on developed label-specific threshold-free framework and frame sequence reconstruction (FSR) network. By experimenting on relatively simple nuclear security scenarios such as external intrusion, the results seem to be practical and promising. In this research, to further advance the performance of abnormal detection process in nuclear security, a future frame prediction (FFP) network is designed while simultaneously combined with label-specific threshold-free framework. At the same time, an expanded self-collected dataset for nuclear security scenarios, mainly added with complex and hard-to-detect scenarios like nuclear material theft, is utilized to verify the performance. The main contributions of this paper are given as follows.

(1) A novel future frame prediction network is designed for abnormal detection of nuclear security scenarios.

(2) The self-collected nuclear security dataset is expanded with complex and hard-to-detect scenarios in nuclear security, which is nuclear material theft.

(3) The calculation results show that the proposed FFP network is obviously better than our previous work with promising evaluation metrics.

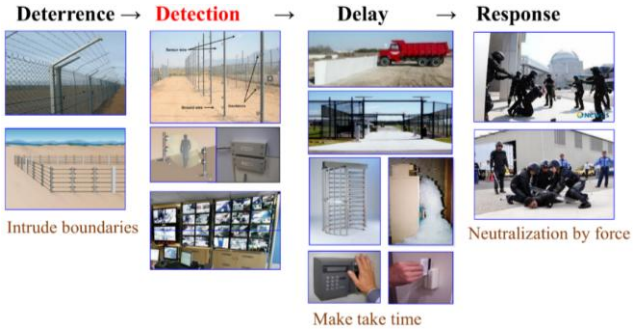

Fig.1 Flow chart of nuclear security measures

2. Methodology

In this section, firstly a brief introduction is given to the limitations of traditional framework and then the proposed label-specific threshold-free framework is presented to solve such limitations. Based on this proposed framework, a future frame prediction network is developed and implemented, combining with two training schemes.

2.1 Limitations of traditional framework

In related papers[3,4], the process of abnormal detection usually follows a common framework shown in Fig.2. In traditional framework, an autoencoder AI-model is supposed to be trained in training stage, which is only fed with normal samples. This process is activated by reconstruction errors shown in Eq. (1).

$$e_r = \|x - f(x)\| \quad (1)$$

Where x and $f(x)$ mean input and reconstructed frame sequences, respectively, e_r denotes the reconstruction error. Then in testing stage, the trained model could be used to calculate the corresponding e_r values for testing data, while the final reasoning results could be determined by comparing calculated e_r values and predefined threshold values. However, according to our previous work[2], the threshold value is very hard to determine which always leads to identical reasoning results.

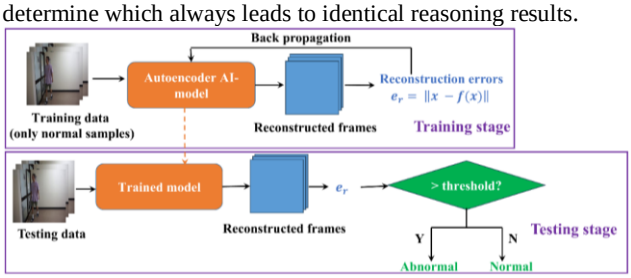

Fig.2 Traditional framework

†東京大学 The University of Tokyo

2.2 Label-specific threshold-free framework

To overcome the limitations mentioned in Section 2.1, a label-specific threshold-free framework is proposed shown in Fig.3. Here, different from the traditional framework shown in Fig.2, two deep learning models are required to be trained in label-specific threshold-free framework. Among them, one is fed only with normal samples while the other one is fed only with abnormal samples. Then in testing stage, these two trained label-specific models would be utilized to obtain the corresponding two label-specific reconstruction errors, denoted as e_r^N and e_r^A in Fig.3. Finally, the reasoning results could be easily obtained by comparing e_r^N and e_r^A without the need to determine threshold values in advance.

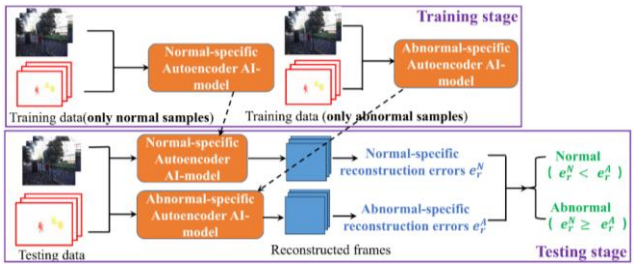

Fig.3 Label-specific threshold-free framework

2.3 Future frame prediction network

Based on the construction of label-specific threshold-free framework, a FSR network is designed and implemented in our previous work[2]. In this research, to further advance the performance of abnormal detection in nuclear security, a novel FFP network is proposed as shown in Fig.4.

As shown in Fig.4, assuming the input frame sequences corresponds to frame 1-*t*, the output data only contains one predicted future frame corresponding to frame *t*+1. In detail, the input frame sequences would be firstly resized to a fixed dimension (227, 227, 10)[2], and then 2 Conv3d operators, 3 ConvLSTM2D operators and 2 Conv3dTranspose operators would be sequentially conducted to, obtaining feature maps with identical dimensions to the resized input frame sequences. The first two Conv3d operators are utilized for capturing spatial and appearance features of input data, and the middle three ConvLSTM2D operators focus on extracting temporal features, while the last two Conv3dTranspose operators are inverse process of Conv3d operators, responsible for reconstructing frame sequences. Finally, the reconstructed frame sequences, which are supposed to emphasize discriminative features corresponding to relatively dynamic parts of human behaviors, would be processed by a MaxPooling3D operator. In this way, all the important and discriminative parts within input frame sequences could be fused and reflected in the predicted future frame. In addition, the reconstruction errors in FFP network should be the difference values between predicted future frame x'_{t+1} and real future frame x_{t+1}. Comparing to FSR network, the proposed FFP network is supposed to emphasize abnormal-related features easily, thus advancing the detection performance.

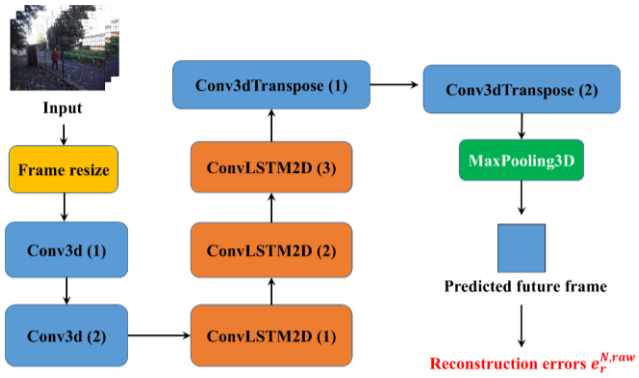

Fig.4 Proposed future frame prediction network

2.4 Training schemes

2.4.1 Extraction of optical flow frames

Before the introduction of training schemes for the proposed FFP network shown in Fig.4, it is necessary to present the common preprocessing method in the field of abnormal detection. Although the raw frame sequence data retains the most primitive and richest appearance characteristics, the diversity of static background part might influence the performance of abnormal detection. To alleviate such risks, a common preprocessing method before abnormal detection work is to extract the corresponding optical flow frames, as shown in Fig.5.

Based on the open-sourced toolbox *MMFlow*[5], the default extraction network called PWC-Net is utilized in this research. In specific, between each pair of consecutive raw frames, the corresponding optical flow frames could be extracted, as shown in the bottom row of Fig.5. The red part in such optical flow frames are highly related to human behaviors, while the yellow part with relatively low pixel values is occurred due to some environment noise such as lighting changes. In fact, such noise have limited influence for abnormal detection and the extraction of optical flow frames based on *MMFlow* toolbox seems to be advisable and acceptable.

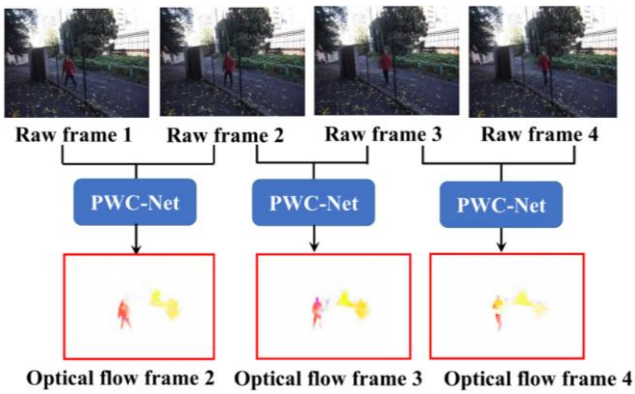

Fig.5 Extraction of optical flow frames

2.4.2 Single-stream training scheme

According to the extraction of optical flow frames introduced in Section 2.4.1, there are two types of input data in our research work, which are raw frames and optical flow frames.

Consequently, how to comprehensively utilize the loss values of these two types of input data during the training process remains an important problem. In our research, two training schemes which are single-stream and double-stream training schemes are further developed.

In this section, the proposed single-stream training scheme is introduced as shown in Fig.6. In this scheme, the corresponding deep learning models used for raw frames and optical flow frames are updated simultaneously while the total loss function values $e^{N}_{r,single}$ are calculated in Eq. (2).

$$e^{N}_{r,single} = w^{N}_{1,single} \times e^{N,raw}_{r,single} + w^{N}_{2,single} \times e^{N,flow}_{r,single} \quad (2)$$

Where $e^{N,raw}_{r,single}$ and $e^{N,flow}_{r,single}$ mean the loss function values corresponding to raw and optical flow frames input. The superscript N denotes normal-specific models shown in Fig.3, and it is only used as an example for Fig.6 while the abnormal-specific models could also be calculated in similar way. In this way, there are only two deep learning models that are required to be trained in single-stream training scheme (i.e., stacked normal-specific and stacked abnormal-specific models).

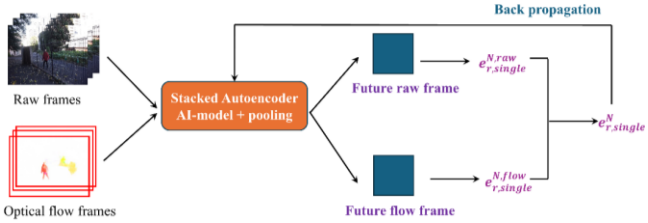

Fig.6 Single-stream training scheme

2.4.3 Double-stream training scheme

Apart from single-stream training scheme mentioned in Section 2.4.2, the other proposed training scheme called double-stream training scheme is also implemented in our research, as shown in Fig.7. In this scheme, the corresponding abnormal detection networks for raw and optical flow frames input are separately updated. Then, in testing stage, the weighted summation of loss function values would be calculated for reconstruction error comparison, as shown in Eq. (3) (similarly, chosen normal-specific as an example).

$$e^{N}_{r,double} = w^{N}_{1,double} \times e^{N,raw}_{r,double} + w^{N}_{2,double} \times e^{N,flow}_{r,double} \quad (3)$$

Although the format of Eq. (3) is almost identical to Eq. (2), the operation stages between these two equations are totally different. The update process of Eq. (2) would be conducted in the training process so that the hyperparameters about deep learning models of raw frames and optical flow frames are consistent all the time. However, the update process of Eq. (3) would only occur in testing stage, and the training process of deep learning models are already completed while their parameters are also frozen and fixed. Consequently, Eq. (3) fails to be an iterative procedure, while we just calculate and compare the weighted summation of normal-specific and abnormal-specific models, denoted as $e^{N}_{r,double}$ and $e^{A}_{r,double}$ (A means abnormal-specific), to determine the final reasoning results of abnormal detection. In this way, four deep learning models are trained in this scheme.

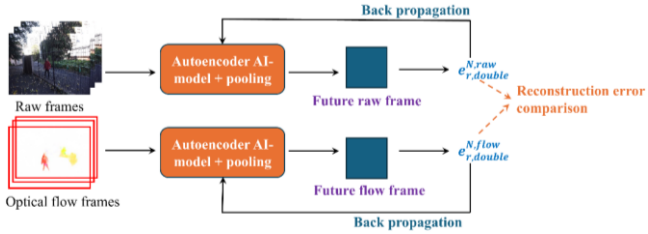

Fig.7 Double-stream training scheme

3. Experiments & results

3.1 Datasets & implementations

To evaluate the proposed FFP network as well as the two training schemes on the task of abnormal detection in nuclear security, a self-collected nuclear security dataset is utilized, as shown in Fig.8. In specific, it is a self-shot dataset simulating five typical nuclear security scenarios, including fence climbing, wire net cutting, weapon holding, sabotage and nuclear material theft. Among them, the final scenario nuclear material theft is extra appended in this research, which is not included in our previous work[2]. The corresponding statistical information in our dataset is given in Table 1. Here, for each video, a sliding window technique is utilized to transfer them into samples with fixed length (10 consecutive frames).

All the experiments are conducted by *Keras* library in Python with a calculation hardware of NVIDIA GeForce RTX 3060.

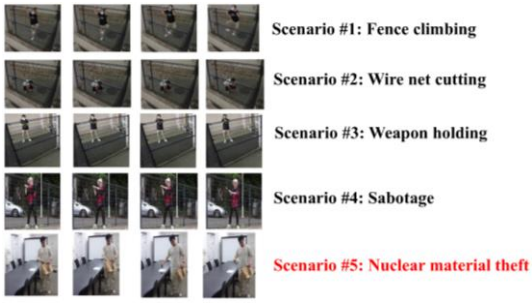

Fig.8 Samples of self-collected dataset

Table 1 Statistical information of dataset

	Training dataset	Testing dataset
Number of samples	6903	2333
Number of videos	141	76
Total time	44 min 40 s	20 min 26 s

3.2 Calculation results of future frame prediction

According to Eq. (2) and Eq. (3), it could be known there are two undetermined parameters for each training scheme. In this section, the value of w_1 is fixed as 1.0, and by adjusting the values of w_2 the calculation results for single-stream and double-stream training schemes are given in Table 2 and Table 3.

Firstly, in Table 2, it could be easily known that the optimal case should be w_2=0.1, with the highest precision, accuracy and F1-score values. Although the corresponding recall value of

0.8824 is not the optimal one, the rank of it is second, only slightly lower than the optimal one of 0.9118 (when w_2=1.0). However, other metrics of w_2 =1.0 are obviously lower, especially for accuracy and F1-score values.

On the other hand, as shown in Table 3, with the increase of w_2 values, only the recall values tend to increase (except when w_2=0.2), while the other three metrics are decreased. In addition, the optimal case in Table 3 is also the first row (when w_2=0.1), achieving the highest precision, accuracy and F1-score values. Its corresponding recall value is also rank the second, while only slghtly lower than the best one (taken as 0.9118).

Table 2 Results of single-stream training scheme

$w_{2,single}$	Precision	Recall	Accuracy	F1-score
0.1	**0.5882**	0.8824	**0.6528**	**0.7059**
0.2	0.5682	0.7353	0.6111	0.6410
0.3	0.5800	0.8529	0.6389	0.6905
0.4	0.5472	0.8529	0.5972	0.6667
0.5	0.5000	0.8235	0.5278	0.6222
0.6	0.5385	0.8235	0.5833	0.6512
0.7	0.5370	0.8529	0.5833	0.6591
0.8	0.5472	0.8529	0.5972	0.6667
0.9	0.5172	0.8824	0.5556	0.6522
1.0	0.5254	**0.9118**	0.5694	0.6667

Table 3 Results of double-stream training scheme

$w_{2,double}$	Precision	Recall	Accuracy	F1-score
0.1	**0.7317**	0.8824	**0.7917**	**0.8000**
0.2	0.6591	0.8529	0.7222	0.7436
0.3	0.6000	0.8824	0.6667	0.7143
0.4	0.5556	0.8824	0.6111	0.6819
0.5	0.5439	**0.9118**	0.5972	0.6814
0.6	0.5345	0.9118	0.5833	0.6739
0.7	0.5000	0.9118	0.5278	0.6458
0.8	0.4921	0.9118	0.5139	0.6392
0.9	0.4844	0.9118	0.5000	0.6327
1.0	0.4844	0.9118	0.5000	0.6327

3.3 Comparison & analysis

Based on the insider comparison process of FFP network, as mentioned in Section 3.2, the corresponding evaluation results of Table 2 and Table 3 are combined and further compared with our previous FSR network[2]. The calculation results are shown in Table 4. As shown in Table 4, the optimal case should be the proposed FFP network with double-stream training scheme, achieving the highest recall value of 0.8824, the highest accuracy value of 0.7917 and the highest F1-score value of 0.8000. In addition, the corresponding precision value of 0.7317 also at a higher level, only slightly lower than the best one of 0.7647.

To clarify the superiority of the FFP network over previous FSR network, a qualitative analysis is given in Fig.9. Here, assuming the input data contains 10 consecutive frames, while only Frame 7 is strongly related to abnormal result. When implementing FSR network, the mean reconstruction error among these 10 frames would be utilized for reasoning, while other abnormal-unrelated frames might dilute the importance of Frame 7. However, for our proposed FFP network, it is only required to calculate the error of Frame 7, while its abnormality could be emphasized, assisting to reasoning process.

Table 4 Results of double-stream training scheme

Work	Training scheme	Precision	Recall	Accuracy	F1
FSR	Single	**0.7647**	0.3824	0.6528	0.5098
	Double	0.7241	0.6176	0.7083	0.6666
FFP	Single	0.5882	0.8824	0.6528	0.7059
	Double	0.7317	**0.8824**	**0.7917**	**0.8000**

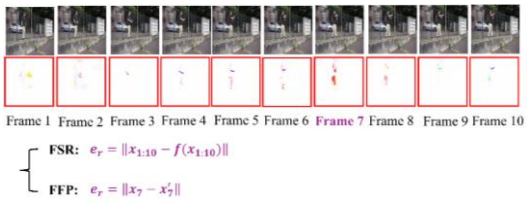

Fig.9 Qualitative analysis between FSR and FFP

4. Conclusions

In this research, a novel FFP network is proposed for abnormal detection in nuclear security video datasets, while an important but trick scenario of nuclear material theft is appended in our experiment. As a result, the proposed FFP network outperforms our previous FSR network, with the optimal recall, accuracy and F1-score values of 0.8824, 0.7917 and 0.8000, while its precision value of 0.7317 is also in high level. According to a qualitative analysis, the proposed FFP could effectively emphasize the abnormal-related frames and features.

Acknowledgment

This work is supported by UTokyo Fellowship. In addition, thanks for the following members for assisting us to collect the dataset used in this research.

·Yang, K.-E., from Sakai's Lab in University of Tokyo
·Yang, F., from Murakami's Lab in University of Tokyo
·Li, R., from Sakai's Lab in University of Tokyo
·Yi, Y., from Fujii's Lab in University of Tokyo
·Laraison, M., from Kasahara's Lab in University of Tokyo
·Cai, X.-Q., from Sugiyama's Lab in University of Tokyo

Reference

[1] U.S. Congress, Office of Technology Assessment, "Technology against terrorism: Structuring security", OTA-ISC-511. https://www.princeton.edu/ota/ disk1/1992/9235/9235.PDF (1992).
[2] Li Z., Song X., Chen S., Demachi K., "Abnormal detection in nuclear security videos based on label-specific autoencoders and reconstruction errors comparison", Nuclear Engineering and Technology, Vol.57, No.3, 103239 (2025).
[3] Li T., Chen X., Zhu F., Zhang Z., Yan H., "Two-stream deep spatial-temporal auto-encoder for surveillance video abnormal event detection", Neurocomputing, Vol.439, 256-270 (2021).
[4] Shao W., Xiao R., Rajapaksha P., Wang M., Crespi N., Luo Z., Minerva R., "Video anomaly detection with NTCN-ML: A novel TCN for multi-instance learning", Pattern Recognition, Vol.143, 109765 (2023).
[5] MMFlow Contributors, "MMFlow: OpenMMLab optical flow toolbox and benchmark", https://github.com/open-mmlab/mmflow/tree/master (Accessed in 2025)