A Lightweight Pedestrian Tracking Target Detection Algorithm

Miao Jin[1], Tianfu Huang[2], Chuning Peng[3], Xiwen Chen[1], Yanli Zhang[4]*

[1] *China Electric Power Research Institute (Wuhan), China*
[2] *State Grid Fujian Electric Power Company (Fuzhou), China*
[3] *State Grid Corporation of China (Beijing), China*
[4] *State Grid Jibei Electric Power Company (Beijing), China*
power_jinmiao@163.com, huangtianfugw@163.com, cp_sgc@163.com,
F-chenxiwen@epri.sgcc.com.cn, power_zhangyanli@163.com

Abstract

In recent years, pedestrian detection and tracking has been widely used in the fields of multimedia, computer vision and pattern recognition. Most of the traditional pedestrian tracking algorithms directly use the pixel level features in each frame of the image sequence in the video to calculate, ignoring the deep semantic feature information contained in each image, resulting in inaccurate or wrong positioning. Moreover, the traditional algorithm has a large number of parameters, high complexity and poor real-time performance. This paper proposes a new lightweight object detection algorithm Res-YOLO-tiny, which aims to accelerate the tracking efficiency of blurred image removal algorithms. It greatly improves the detection speed, and then combines with the PID (proportion integration differentiation) algorithm to increase the tracking effect and positioning accuracy of the whole tracking system through a new ROI (region of interest) combination method. Finally, the experimental and practical results show that the proposed algorithm has high positioning accuracy, sensitive tracking response, strong recognition robustness and flexibility.

Keywords: Pedestrian detection, Pedestrian tracking, Lightweight object detection

1 Introduction

With the development of object detection and tracking system [1-3] in the field of vision, intelligent video surveillance has been widely used in today's society [4]. Among them, multi-object tracking visualization technology [5] is an important technology in the field of computer vision and computer graphics, which is the basis of many follow-up studies, such as target recognition, point cloud, virtual try on, intelligent monitoring, abnormal behavior detection, etc. [6-12] has great application prospects in both civil and military fields. Object detection and tracking algorithms have made significant progress [13]. However, challenges persist in accurately and efficiently tracking objects in complex scenes. Some studies propose solutions. Furthermore, the algorithms used may not meet current requirements for accuracy and speed, limiting their effectiveness in real-time scenarios.

To improve accuracy, researchers have explored convolutional neural networks (CNNs) [14]. Luo et al. [15] proposed the Matching-Siamese network for pedestrian tracking, while Faster-RCNN [16] is used for pedestrian detection in surveillance videos. However, these approaches suffer from large network models and slow operation. Duanmu et al. [17] introduced a multi-view pedestrian tracking framework using graph matching techniques. However, this framework increases hardware costs. Traditional tracking algorithms perform well in specific scenes but face challenges in real-life scenarios due to variations in clothing, shape changes, backgrounds, lighting conditions, and occlusions among pedestrians. It is crucial to address these challenges without significantly increasing system costs and computational requirements.

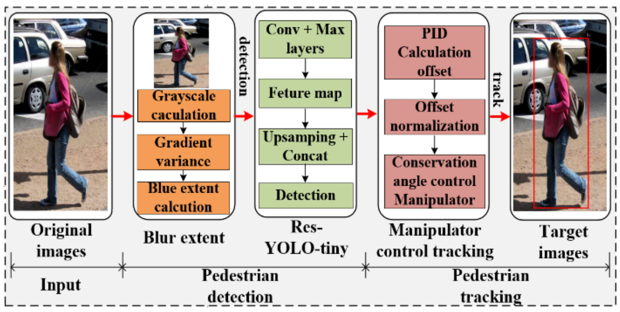

Figure 1. General process of pedestrian tracking system

Therefore, this paper designs and implements a pedestrian tracking system and method that schedules a tracking algorithm and a lightweight neural network for real-time object detection, as shown in Figure 1. The algorithm uses the improved lightweight YOLO-tiny [18] to detect the pedestrian in the image, realize the real time detection of the target and determine the task coordinate information. This achieves real-time detection of the target to determine the pedestrian coordinate information, and then to achieve the group target tracking information visualization. PID algorithm [19] is used to process the pedestrian coordinate information, so it can track the moving target in the scene at any time. In order to solve

*Corresponding Author: Tianfu Huang; E-mail: huangtianfugw@163.com
DOI: https://doi.org/10.70003/160792642025052603012

the problem that the tracking object cannot be determined in the case of dense pedestrians, this paper uses an algorithm to integrate the target object group into a whole for tracking and detection. In addition, for the uncertainty of the range of activity of the target, this paper proposes a weekly detection method to actively find the target in the case of none. Experimental results show that the proposed method accelerates the accuracy and sensitivity of pedestrian tracking and improves the accuracy and speed of pedestrian tracking. It can calculate multi-angle and multi-bit simultaneously in static scenes, which is more conducive to pedestrian detection and tracking.

The remainder of this paper is arranged as follows. Section 2 describes the methods. Section 3 will carry on the experiment and the performance evaluation to our method. We have made future improvements to our work discussed in Section 4. Finally, Section 5 presents a brief conclusion to this paper.

2 Methods

Pedestrian tracking is more and more widely used in military and civil fields. The most effective method is to extract pedestrian features by CNN [20], but the model is too large to guarantee real-time. The pedestrian tracking algorithm proposed in this paper detects the input pedestrian image sequence, analyzes the parameters of the pedestrian and performs parameter transformation to track the pedestrian. The algorithm and setup of this paper are described in detail below.

2.1 Blurred Image Filtering

In order to improve the real-time performance and robustness of pedestrian detection, this paper proposes, for the first time, a blur image filtering preprocessing method based on image gradient analysis. This method automatically filters out irrelevant images caused by motion blur or other factors in the image sequence, reducing the computational load for subsequent detection.

Motion blur, occurring due to camera movement during short exposure times, leads to blurry images in image sequences. These blurred images lack clarity and can be mistakenly detected by pedestrian detection algorithms. To mitigate this issue, the paper utilizes image gradient analysis.

The paper uses image gradient to differentiate between clear and blurred images. Clear images exhibit strong gradients at boundaries, while blurred images have smaller gradients. By converting to grayscale and analyzing the gradient, the paper filters out blurry images to conserve detection resources and improve efficiency. The calculation process involves analyzing amplitude and angle gradients to determine if an image is blurred.

First, the initial image should be converted into gray image. Then, Laplacian operator [21] is used to extract edge information from image. Finally, the sum of edge information gradients after processing is calculated and the result is used as the blur degree of the image. The calculation formula is as follows:

$$I(x,y) = w_R R_{x,y} + w_G G_{x,y} + w_B B_{x,y} \tag{1}$$

$$Laplacian(f) = \nabla^2 f(x,y) = \frac{\partial^2 I(x,y)}{\partial x^2} + \frac{\partial^2 I(x,y)}{\partial y^2} \tag{2}$$

$$Lap = \frac{1}{N}\sum_x \sum_y \nabla^2 f\left(x,y\right) \tag{3}$$

where, $I(x, y)$ represents the gray value at the point (x, y), $\nabla^2 f(x, y)$ represents the gradient at (x, y), N is the total number of pixels in the image, and Lap is the ambiguity of the final calculation. After that, a threshold H is determined through test comparison, and the fuzziness greater than this threshold is considered clear. Otherwise, it is considered blurred and skip no detection.

2.2 Pedestrian Tracking

The object detection algorithm is crucial for pedestrian tracking. However, current mainstream two-stage detection algorithms such as Faster R-CNN are relatively slow, and single-stage algorithms like YOLO struggle to meet the requirements of real-time and high precision. Therefore, this paper innovatively incorporates residual structures and model compression strategies into the YOLO-tiny network, designing the Res-YOLO-tiny lightweight object detection network. First of all, we compare YOLO-tiny and YOLO-lite [22]. Finally, we choose to improve the YOLOv3-tiny to realize our own network architecture. This paper mainly improves the BackoneNet of YOLOv3-tiny. The basic structure is shown in Figure 2 to improve the recall rate and accuracy and speed up the training convergence speed.

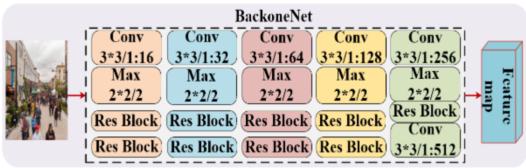

Figure 2. Res-YOLO-tiny network architecture

Res-YOLO-tiny network mainly improves the BackoneNet of YOLOv3-tiny, adds a convolution layer with a deep channel number at the end of the original network to extract more semantic features, and adds a ResBlock form ResNet [23] at the back of each layer of the first 4 convolution pooling layers to prevent information loss and gradient dispersion caused by the deepening of network depth.

The BackoneNet model of Res-YOLO-tiny composed of 8 convolution layers with convolution kernels of 3 × 3 size and 4 ResBlocks. In addition, except the last convolution layer, the other convolution layers will follow a maxpooling layer and a rectified linear unit (ReLU). After the model input 224 × 224 × 3 image, it will generate 13 × 13 × 2048 feature map.

In the ResBlock layer, there are two convolution modules. Each convolution module is composed of a convolution kernel of 3 × 3 and a convolution step with a value of 1. Then, the tensor of the original input layer and the convoluted output layer are added to synthesize the final output layer.

Compared with YOLOv3-tiny algorithm, Res-YOLO-tiny can acquire deeper information features, which greatly improves the positioning accuracy and speed of the target, and only increases twice the amount of YOLOv3-tiny data. Because of the characteristics of high speed and low cost, it can meet the requirements of target detection and tracking. It can also achieve a faster speed on common equipment and achieve better results.

When the pedestrian goes out of this range, the target cannot be tracked and located, and the pedestrian does not often move in the center of the camera perspective, which greatly reduces the amount of target information acquisition and the flexibility of target tracking and marginalizes the detection effect. In order to solve these problems, this paper proposes a real-time tracking algorithm. Res-YOLO-tiny will send the pedestrian coordinate parameters detected and located to the tracking algorithm, which will calculate the X and Y-direction normalized offsets Δx, Δy of the camera and the ROI (region of interest) area of pedestrians, and then the tracking algorithm will use the distance offsets to calculate the angle offsets θx, θy of the mechanical arm relative to the pedestrian, as shown in Figure 3. The coordinate starting point of the whole image starts from (0,0), the width of the image is width (W), the height of the image is height (H), the center point coordinate of the image is calculated from width and height ($W/2$, $H/2$), and the center point coordinate value of ROI is calculated.

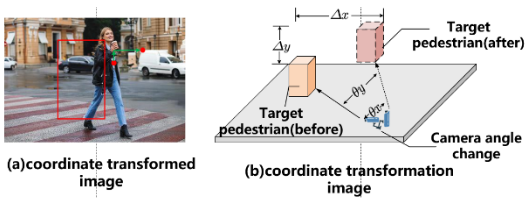

Figure 3. Coordinate transformation

When the human body is detected by Res-YOLO-tiny, the ROI of the human body object in the image will be returned, including the normalized x, y, w, h values and confidence. x and y represent the upper left corner of the ROI region relative to the X direction of the whole image and the starting coordinates in Y direction. w and h represent the width and height of ROI. Confidence indicates whether ROI is equal to the correct prediction of the object the confidence degree of ROI, where the central coordinate value (C_x, C_y) of ROI is calculated by the following formula:

$$C_x = x + \frac{w}{2}$$
$$C_y = y + \frac{h}{2} \tag{4}$$

The coordinate value of the center pixel of the whole image (T_x, T_y) is:

$$T_x = \frac{W}{2}$$
$$T_y = \frac{H}{2} \tag{5}$$

Thus, we can get the distance offsets between the ROI center and the center of the whole picture.

Because different pedestrians are far from each other in the whole pedestrian, the pixel value representing the size is very different. Therefore, to scale the calculated distance offsets to the range of [-1,1] as a whole for normalization processing [24], the formula of the calculated normalized distance offsets (Δx, Δy) is as follows:

$$\Delta x = 2 \times \left(\frac{C_x}{T_x} - 0.5 \right)$$
$$\Delta y = 2 \times \left(\frac{C_y}{T_y} - 0.5 \right) \tag{6}$$

When the distance offsets changes, the angle increment of the servo will also change, so the distance offsets and angle offsets show a linear relationship. Here, a scale factor (Kp) is defined to describe the relationship. The formula is as follows:

$$D_{nx} = D_{ox} + K_p \cdot \Delta x$$
$$D_{ny} = D_{oy} + K_p \cdot \Delta y \tag{7}$$

Among them, the angle of the servo is represented by Do, and the angle after the change of the servo is represented by Dn.

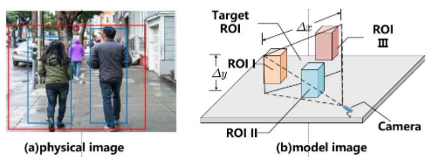

Figure 4. ROI merging algorithm

If the camera's central pixel coordinate is always near the human body's central pixel coordinate (C_x, C_y), the servo will vibrate due to frequent fine-tuning back and forth. We set a range deadband. When the calculated normalized offsets exist in the deadband, the servo adjustment and control operation will be cancelled.

When there are multiple pedestrians (multiple ROI) in the current image, we adopt an algorithm that can combine these ROI into a total ROI to track. As shown in Figure 4, due to the different distribution size and direction of pedestrians in the image, we combine the upper left corner coordinates of n ROI under the same frame of image (x_i, y_i) into the coordinates closest to the origin (x_{min}, y_{min}), and combine the width of all ROI w_i into the value wmax farthest from the origin of X axis. Similarly, the height hi of all ROI is summed up as the value h_{max} farthest from the origin of Y axis.

In this way, we will get a final ROI for tracking. When the number of pedestrians in the field of view image decreases or increases at a certain moment, the algorithm will adjust the parameters in time, so as to solve the problem of uncertainty of tracking objects under multiple targets and jitter caused by tracking pedestrians in turn.

3 Experimental Evaluation

3.1 Expansion of Experimental Dataset

The computer configuration used in this experiment is Intel (R) core i5-4200M CPU, 8GB RAM, 2GB GTX840 GPU. The software environment is pycharm 2.2 combined with Python 3.7 under the operating system window10. The following will show and analyze the experiment.

The official provides 20 weights of target categories, but due to the change of BackoneNet, the official weights cannot be used, and the system only needs to detect pedestrians, so our model only trains for the type of person data. We selected and sorted out the VOC dataset [25] and COCO dataset [26] and merged them into a dataset containing 4000 images. 3000 images of the human body were randomly selected as the training model, 500 pictures were selected as the verification set, and the remaining 500 pictures were used as the test set, and 50 epochs were trained iteratively. The experimental environment is placed indoors, and the equipment is installed in a wide area close to the active area.

3.2 Image Gradient Experiment

In order to filter the blurred image before image detection, we choose nine image gradient calculation methods to test the time-consuming of single image filtering. We randomly select 500 images in the dataset for gradient calculation one by one and record the sum of them. At last, we calculate the average time-consuming of each method to filter the single clear and blurred image respectively.

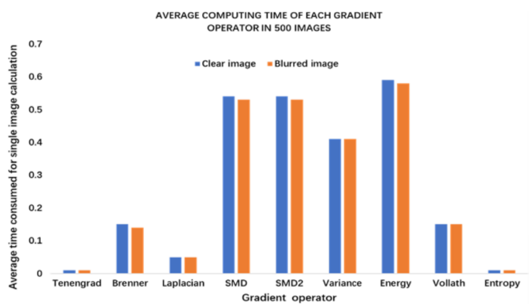

Figure 5. Average computing time of each gradient operator

We selected 9 methods, namely Tenengrad [27], Brenner, Laplacian [28], SMD, SMD2, Variance, Energy, Vollath and Entropy. After calculation, the average detection time consumption of each function with clear and fuzzy filtering [29] is shown in Table 1, and Figure 5.

Table 1. Average calculation times of single image for each image gradient calculation methods

Image	Tenengrad	Brenner	Laplacian
Clear image	0.002579	0.146626	0.001635
Blurred image	0.002589	0.145416	0.001755
-	**SMD**	**SMD2**	**Variance**
-	0.551884	0.551490	0.411390
-	0.546165	0.544429	0.412433
-	**Vollath**	**Entropy**	**Energy**
-	0.141084	0.001447	0.594332
-	0.140711	0.001570	0.587399

We stipulate that the threshold value H of the image's fuzziness is 30. When the fuzziness is greater than H, the image is determined to be clear and can be detected in the system. If the fuzziness is less than the threshold h, the image is determined to be a fuzzy image that cannot be detected by the system.

3.3 Pedestrian Object Detection

In this paper, the Res-YOLO-tiny network model is used for the experiment. The initial learning rate is set to 5e-3, and the learning rate termination value is set to 1e-6, a weight decay of 5e-4 is used, the training data has 50 epochs, and the batch size is set to 1.

The experimental results show that the enhancement of the image can greatly increase the generalization ability of the model, reduce the over fitting phenomenon in the process of image training, remove the external interference as much as possible, and then enhance the detection ability of the model.

We compared the performance of the other four models: YOLOv2-tiny, YOLOv3-tiny, YOLO-lite, SSD mobilenet V1 [30-31] and our model on our low performance devices. It can be seen from the Table 2 that when comparing the performance of the five models, the introduction of residual network can deepen the layer number and weight parameters of the network, and effectively improve the accuracy and recall rate of the network model. It can be seen that the accuracy of this model is better than other lightweight networks, and the detection speed has no significant impact.

Table 2. Performance comparison of various models on COCO dataset

Model	mAP (COCO)	FPS
YOLOv3-tiny	33.10%	23
YOLOv2-tiny	23.70%	5.6
YOLO-lite	12.26%	34
SSD Mobilenet-V1	21%	8.2
Res-YOLO-tiny	28.42%	28

Before the operation of the detection network in this paper, the fuzzy degree of the image collected by the camera will be calculated first, which is used to remove the fuzzy images collected in the process of motion. The existence of fuzzy images will greatly waste the cost of calculation and reduce the efficiency of operation. We randomly cut 1000 image sequences to judge the fuzzy degree. The result shows that 464 images are fuzzy and the remaining 536 images are clear. As shown in Figure 6, we calculate the total execution time of the algorithm with and without fuzzy degree judgment.

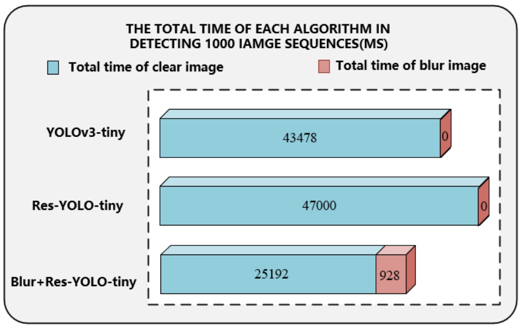

Figure 6. The total time of each algorithm in detecting 1000 image sequences

It can be seen that the target tracking system using the fuzzy image removal algorithm can save nearly 40% of the time to detect new images, greatly improving the execution efficiency of the whole system, and the final tracking effect is shown in Figure 7.

Figure 7. Camera device tracking effect

We test the feature extraction layer of the target detection network, in which we use the CIFAR dataset as our detection database. The CIFAR dataset is divided into CIFAR-10 and CIFAR-100. The CIFAR-10 dataset consists of 60000 32 × 32 color images of 10 classes, and each class has 6000 images with a total of 50000 training images and 10000 test images. There are 100 classes in CIFAR-100 dataset. Each class contains 600 images of 32 × 32 size, and each class has 500 training images and 100 test images.

We use the CIFAR-100 dataset to calculate the error rate, parameter quantity and the floating point of operations (FLOPs) of these five kinds of algorithms. As shown in Table 3, the Res-YOLO-tiny network designed by us does not have much difference in accuracy compared with the original network (YOLOv3-tiny), but the number of parameters is reduced by 44.5%, and the number of floating point operations is also reduced by 43.79% compared with the original network architecture. The reduction of these parameters can greatly improve the detection speed without greatly reducing the accuracy, which just meets the requirements of our equipment for the detection algorithm speed.

Table 3. Performance of top-1 err., top-5 err., number of parameters and MFLOPs of various models on CIFAR-100

Model	top-1 err.	top-5 err.	Params (M)	MFLOPs
YOLOv3-tiny	0.7805	0.514729	24.4023	12.60933
VGG19	0.6741	0.382415	97.14	50.9331
YOLO-lite	-	-	2.1879	1.145644
MobileNetV2	-	-	8.6624	4.468249
Res-YOLO-tiny	0.7841	0.527321	13.5431	7.087719

In our work, we use the CIFAR-10 dataset to train and test the Res-YOLO-tiny model. As shown in Figure 8, we train a total of 25 epochs on the dataset, stopped training when the accuracy rate tended to be stable, and recorded the data of each node. After training to the 7th epoch on the test set, we get the accuracy of 0.7174, and the loss value is close to the balance of the curve at the 18th epoch. It can be seen that the overall convergence speed of the model is faster. As shown in Figure 9, we obtain the confusion matrix of Res-YOLO-tiny model on CIFAR-10 dataset, in which the detection error between cat and dog is the largest.

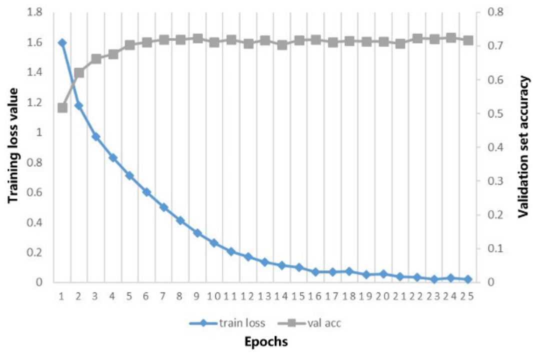

Figure 8. The training loss and validation accuracy of our designed network architecture on CIFAR-10

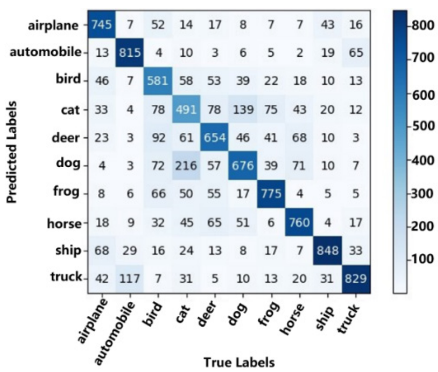

Figure 9. The confusion matrix of the network architecture we designed on CIFAR-10

3.4 Pedestrian Object Tracking

Our proposed target tracking device utilizes a manipulator controlled by a Raspberry Pi [32]. The device calculates and controls the manipulator's steering gear based on values transmitted from the model. A PID algorithm on the Raspberry Pi regulates the direction and angle of the steering gear. To handle multiple pedestrian targets in a single image, we employ an algorithm that groups pedestrians in the field of view into a single target. This integration of correct ROI areas allows the camera device to track a unified target, solving the challenge of selecting which target to track. To ensure real-time performance, the device has a specific time allocated for each frame of image detection. This time includes the duration of image detection and transmitting the results to the manipulator device for scheduling. We have modified the network structure to speed up detection and optimized the scheduling algorithm on the manipulator to guarantee responsive and timely device operation.

4 Conclusion

This paper presents an algorithm for real-time tracking and recording of pedestrian's behavior and motion in complex situations. The algorithm proposed in this paper aims at the camera to expand the detection range, reduce other interference factors, and increase the tracking efficiency and accuracy. During the tracking process, the motion blurred image caused by the fast speed of human motion or camera rotation is removed by calculating the image blur. We improve the target detection network, make it in a certain speed range, and improve the detection ability of positive samples of the algorithm. We also avoid the acquisition of irrelevant images and prevent the loss of tracking targets. The experimental results show that the scheme is feasible. It greatly improves the accuracy of pedestrian recognition, increases the efficiency of pedestrian tracking, and ensures the real-time performance of the device. Compared with the traditional tracking method, the effect is significantly improved. In addition, future research directions and prospects encompass incorporating additional environmental factors, expanding into multi-object tracking, enhancing deep learning models, exploring cross-domain applications, and improving datasets and evaluation metrics. By delving into these areas, further advancements can be achieved to enhance the algorithm's performance and applicability, thereby fostering advancements in relevant fields.

Acknowledgements

This work was supported by STATE GRID Corporation of China, the science and technology projects of research on efficient operation and safety risk intelligent analysis technology in measurement field (5700-202314264A-1-1-ZN).

MJ performed the data analyses and wrote the manuscript; TH performed the experiment; CP, BL contributed significantly to analysis and manuscript preparation; XC, YZ contributed to the conception of the study; ZG, JZ contributed to the dataset preparation; XW helped perform the analysis with constructive discussions. All authors read and approved the final manuscript.

References

[1] P. Singh, R. Krishnamurthi, IoT-Powered Real-Time Object Detection and Alert System for Intelligent Surveillance, *International Journal of Performability Engineering*, Vol. 20, No. 4, pp. 232–241, April, 2024.

[2] A. Kumarand, G.-S. Lehal, Layout Detection of Punjabi Newspapers using the YOLOv8 Model, *International Journal of Performability Engineering*, Vol. 20, No. 3, pp. 186–193, March, 2024.

[3] S.-K. Pal, A. Pramanik, J. Maiti, P. Mitra, Deep learning in multi-object detection and tracking: state of the art, *Applied Intelligence*, Vol. 51, No. 9, pp. 6400–6429, September, 2021.

[4] T. Sultana, K. Wahid, Iot-guard: Event-driven fog-based video surveillance system for real-time security management, *IEEE Access*, Vol. 7, pp. 134881–134894, September, 2019.

[5] X. Wan, J. Cao, S. Zhou, J. Wang, N. Zheng, Tracking beyond detection: Learning a global response map for end-to-end multi-object tracking, *IEEE Transactions on Image Processing*, Vol. 30, pp. 8222–8235, September, 2021.

[6] J. Li, Q. Huang, Y. Du, X. Zhen, S. Chen, L. Shao, Variational abnormal behavior detection with motion consistency, *IEEE Transactions on Image Processing*, Vol. 31, pp. 275–286, December, 2021.

[7] F. Yu, Y. Zhang, H. Li, C. Du, L. Liu, M. Jiang, Phase Contour Enhancement Network for Clothing Parsing, *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 1, pp. 2784–2793, February, 2024

[8] S. Wan, S. Ding, C. Chen, Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles, *Pattern Recognition*, Vol. 121, Article No. 108146, January, 2022.

[9] J. Yan, H. Jiao, W. Pu, C. Shi, J. Dai, H. Liu, Radar sensor network resource allocation for fused target tracking: A brief review, *Information Fusion*, Vol. 86-87, pp. 104–115, October, 2022.

[10] F. Yu, A. Hua, C. Du, M. Jiang, X. Wei, T. Peng, L. Xu, X. Hu, Vton-mp: Multi-pose virtual try-on via appearance flow and feature filtering, *IEEE Transactions on Consumer Electronics*, Vol. 69, No. 4, pp. 1101–1113, November, 2023.

[11] Y. Sun, J. Yang, W. An, Infrared dim and small target detection via multiple subspace learning and spatial-temporal patch-tensor model, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 59, No. 5 pp. 3737–3752, May, 2021.

[12] C. Du, F. Yu, M. Jiang, A. Hua, X. Wei, T. Peng, X. Hu, Vton-scfa: A virtual try-on network based on the semantic constraints and flow alignment, *IEEE Transactions on Multimedia*, Vol. 25, pp. 777–791, February, 2022.

[13] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, S. Lyu, Ua-detrac: A new benchmark and protocol for multi-object detection and tracking, *Computer Vision and Image Understanding*, Vol. 193, Article No. 102907, April, 2020.

[14] F. Yu, C. Yu, Z. Tian, X. Liu, J. Cao, L. Liu, C. Du, M. Jiang, Intelligent Wearable System With Motion and Emotion Recognition Based On Digital Twin Technology,

IEEE Internet of Things Journal, Vol. 1, pp. 1–14, April, 2024. DOI: 10.1109/JIOT.2024.3394244

[15] Y. Luo, D. Yin, A. Wang, W. Wu, Pedestrian tracking in surveillance video based on modified cnn, *Multimedia tools and applications*, Vol. 77, No. 18, pp. 24041–24058, September, 2018.

[16] V. Sharma, R.-N. Mir, Saliency guided faster-rcnn (SGFr-RCNN) model for object detection and recognition, *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 5, pp. 1687–1699, May, 2022.

[17] F. Duanmu, X. Feng, X. Zhu, W. Tan, Y. Wang, A multi-view pedestrian tracking framework based on graph matching, *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, Miami, Florida, USA, 2018, pp. 315–320.

[18] L. Guan, L. Jia, Z. Xie, C. Yin, A Lightweight Framework for Obstacle Detection in the Railway Image Based on Fast Region Proposal and Improved YOLO-Tiny Network, *IEEE Transactions on Instrumentation and Measurement*, Vol. 71, pp. 1–16, February, 2022.

[19] I. Carlucho, M.-D. Paula, G. Acosta, Double q-pid algorithm for mobile robot control, *Expert Systems with Applications*, Vol. 137, pp. 292–307, December, 2019.

[20] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 12, pp. 6999–7019, December, 2022.

[21] S. Chen, R. Zhan, W. Wang, J. Zhang, Learning Slimming SAR Ship Object Detector Through Network Pruning and Knowledge Distillation, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 14, pp. 1267–1282, 2021.

[22] H. Zhao, Y. Zhou, L. Zhang, Y. Peng, X. Hu, H. Peng, X. Cai, Mixed yolov3-lite: a lightweight real-time object detection method, *Sensors*, Vol. 20, No. 7, Article No. 1861, April, 2020.

[23] Z. Wu, C. Shen, A. Van Den Hengel, Wider or deeper: Revisiting the resnet model for visual recognition, *Pattern Recognition*, Vol. 90, pp. 119–133, June, 2019.

[24] D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, *Applied Soft Computing*, Vol. 97, Article No. 105524, December, 2020.

[25] R. Zhang, H. Wang, Y. Tan, M. Zhang, X. Zhang, K. Wang, W. Ji, L. Sun, X. Yu, J. Zhao, B. Xu, J. Xiong, Using a machine learning approach to predict the emission characteristics of vocs from furniture, *Building and Environment*, Vol. 196, Article No. 107786, June, 2021.

[26] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, A. Torralba, Semantic understanding of scenes through the ade20k dataset, *International Journal of Computer Vision*, Vol. 127, No. 3, pp. 302–321, March, 2019.

[27] H.-I. Jeong, Y.-J. Jeong, Focus measure of tunnel images, *ICIC Express Letters*, Vol. 11, No. 9, pp. 875–880, September, 2020.

[28] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. Meerschaert, M. Ainsworth, What is the fractional laplacian? a comparative review with new results, *Journal of Computational Physics*, Vol. 404, Article No. 109009, March, 2020.

[29] X. Song, M. Wang, B. Zhang, S. Song, Event triggered reliable h fuzzy filtering for nonlinear parabolic pde systems with markovian jumping sensor faults, *Information Sciences*, Vol. 510, pp. 50–69, February, 2020.

[30] Z. Ouyang, J. Niu, T. Ren, Y. Li, J. Cui, J. Wu, Mbbnet: An edge iot computing-based traffic light detection solution for autonomous bus, *Journal of Systems Architecture*, Vol. 109, Article No. 101835, October, 2020.

[31] Z. Qin, W. Wang, K.-H. Dammer, L. Guo, Z. Cao, Ag-yolo: A real-time low-cost detector for precise spraying with case study of palms, *Frontiers in Plant Science*, Vol. 12, Article No. 753603, December, 2021.

[32] F. Yu, Z. Chen, M. Jiang, Z. Tian, T. Peng, X. Hu, Smart Clothing System With Multiple Sensors Based on Digital Twin Technology, *IEEE Internet of Things Journal*, Vol. 10, No. 7, pp. 6377–6387, April, 2023.

Biographies

Miao Jin was born in Huanggang, Hubei Province, China, in 1989. He received the M.Sc. and Ph.D. degrees from Wuhan University, Wuhan, China. He is a senior engineer at the State Grid of China. Currently, he mainly focuses on the research of intelligent perception technology of power.

Tianfu Huang was born in Fuzhou, Fujian Province, China, in 1988. He received the M.Sc. degree from North China Electric Power University, Beijing, China. Currently, He is a senior engineer at the State Grid of China. At present, he is involved in the research of artificial intelligence technology for electricity.

Chuning Peng was born in Beijing, China, in 1973. He received the B.Sc. degree from Huazhong University of Science and Technology, Wuhan, China. He is a senior engineer at the State Grid of China. His current research interests include electric power measurement and perceptive technology.

Xiwen Chen was born in Jingzhou, Hubei Province, China, in 1983. He received the B.Sc. degree from Huazhong University of Science and Technology, Wuhan, China. He is a senior engineer at the State Grid of China. Currently, He mainly conducts research on high-voltage power metering technology.

Yanli Zhang was born in Tangshan, Hebei Province, China, in 1975. She is a senior engineer at the State Grid of China. At present, her research interests include energy metering technology and data analysis technology.