SPECIAL SECTION ON AI-DRIVEN BIG DATA PROCESSING: THEORY, METHODOLOGY, AND APPLICATIONS

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 18, 2019, accepted February 28, 2019, date of publication March 6, 2019, date of current version March 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2902658

Deep Learning for Risk Detection and Trajectory Tracking at Construction Sites

YU ZHAO, (Student Member, IEEE), QUAN CHEN, (Student Member, IEEE), WENGANG CAO, (Student Member, IEEE), JIE YANG, (Member, IEEE), JIAN XIONG, (Member, IEEE), AND GUAN GUI, (Senior Member, IEEE)

College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

Corresponding authors: Jie Yang (jyang@njupt.edu.cn) and Guan Gui (guiugan@njupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701258, in part by the Jiangsu Specially Appointed Professor under Grant RK002STP16001, in part by the Innovation and Entrepreneurship of Jiangsu High-Level Talent Grant under Grant CZ0010617002, in part by the ''Summit of the Six Top Talents'' Program of Jiangsu under Grant XYDXX-010, and in part by the 1311 Talent Plan of Nanjing University of Posts and Telecommunications.

ABSTRACT This paper investigates deep learning for risk detection and trajectory tracking at construction sites. Typically, safety officers are responsible for inspecting and verifying site safety due to many potential risks. Traditional target detection algorithms depend heavily on hand-crafted features. However, these features are difficult to design, and detection accuracy is poor. To solve these problems, this paper proposes a deep-learning-based detection algorithm that uses pedestrian wearable devices (e.g., helmets and colored vests) to identify pedestrians. We train a special dataset by labeling helmets and colored vests to detect the two features among construction workers. Specifically, Kalman filter and Hungarian matching algorithms are employed to track pedestrian trajectories. The testing experiment is run on an NVIDIA GeForce GTX 1080Ti with a detection speed of 18 frames/s. The mean average precision can reach 0.89 when the intersection over union is set at 0.5.

INDEX TERMS Safety officer detection, pedestrian tracking, deep learning, Kalman filter, Hungarian matching algorithm.

I. INTRODUCTION

Intelligent detection is an emerging technology in computer vision and internet of things (IoT), especially at construction sites. The core idea is to replace manpower in traditional technology with artificial intelligence (AI) technology. With the development of artificial intelligence, intelligent detection can achieve better identification performance than humans. Intelligent detection technology has been applied to many practical scenarios; however, actual scenarios pose many risks, and existing technologies cannot fully monitor all of them. Many companies thus require safety officers to supervise local safety throughout projects. However, it is not always possible to tell whether safety officers are on site and their trajectories are being supervised effectively.

At construction sites, many potential risks often occur such as falling objects from tall building. Wearing a helmet can avoid these risks or at least reduce the degree of injury. In addition, there are many dangerous places at the construction site and hence non-professionals are not allowed. So we have to verify the identity of the workers and track their walking trajectories. Workers wearing a vest is not only be used to identify the identity, but also the eye-catching color is more conducive to confirm their position.

In traditional target detection, the sliding window method is used to determine the candidate region. Then, hand-crafted features (e.g., Hog [1], Haar [2], LBP [3]) are used for feature extraction, followed by the use of classifiers for identification. To improve detection accuracy, system complexity must increase continuously, which requires growing detection efficiency. However, the sliding window method contains many repeated calculations, which cannot meet real-time requirements of the system. In addition, hand-crafted features require extensive expert knowledge and are not robust. Hinton *et al.* [4] proposed the concept of deep learning in 2006. With ongoing improvements in computer performance, deep learning has evolved substantially. The approach has come to be widely applied in various fields, such as intelligent wireless communications [5]–[14], natural language

The associate editor coordinating the review of this manuscript and approving it for publication was Zhanyu Ma.

processing [15]–[21], computer vision [22]–[32], and robot design [33]–[38].

Computer vision presents an important application area for deep learning. Image classification, detection, and segmentation are three major tasks in this field. Current deep-learning-based target detection algorithms can be classified into two-stage and one-stage algorithms. Conventional two-stage detection algorithms include faster region-based convolutional neural networks [39] and region-based fully convolutional networks [40]. They divide the detection mechanism into two phases. First, the network generates candidate regions and then detects and classifies these regions. Common examples of one-stage detection algorithms are the single shot multi-box detector [41] and You Only Look Once (YOLO) [42]. They directly generate the class probability and position coordinate values of the object. The advantage of one-stage detection algorithms is their rapid detection speed, whereas two-stage detection algorithms possess high detection accuracy. In addition, Zhang *et al.* [43], Cheng *et al.* [44], and Han *et al.* [45]–[47] and took the lead in combining deep learning with target detection. They have made outstanding contributions in the field of target detection.

Given the limitations of current technology, it is impossible to efficiently and effectively detect all dangerous situations at construction sites; therefore, safety officers are indispensable. This paper proposes deep-learning-based risk detection and trajectory tracking for safety officers at construction sites. Safety officers often wear red vests and helmets, which are obvious features used to identify safety officers in our proposed method. To improve detection accuracy, we have established a proprietary dataset. We use YOLOv3 [48] to implement safety officer detection, which guarantees real-time detection. Based on the detection results, the Kalman filter [49] and Hungarian matching algorithm [50] are used to establish a correlation between the previous frame and current frame, ultimately achieving safety officer tracking.

The remainder of this paper is arranged as follows. Section II proposes the deep-learning-based risk detection and tracking algorithm. Section III presents the experimental results when testing our proposed method. Section IV concludes the paper.

II. PROPOSED METHOD

In this paper, we propose a method for deep-learning-based risk detection and trajectory tracking at a construction site. The method is divided into two parts, namely safety officer detection and pedestrian tracking. In this section, we discuss the method principles in detail.

A. PRINCIPLE OF YOLOV3

YOLOv3 is an excellent performance network structure, which transforms the problem of target detection into a regression problem. For a given image, the bounding box of the target and its classification category are directly returned at multiple image locations. Thanks to this design, the detection speed of YOLOv3 is quite fast, essentially meeting real-time requirements. We review each aspect of the algorithm below.

YOLOv3 is a typical supervised learning algorithm. For a given picture, we first divide it into S × S grids. If the center of an object falls within this grid, then the grid is responsible for predicting the object. Each grid predicts B bounding boxes and the category confidence to which these bounding boxes belong. This is good for the detection of small objects or overlapping objects. Each bounding box contains five pieces of information (x, y, w, h, C), denoting the center position, width, height of the bounding box, and confidence, respectively. Confidence reflects the accuracy of the bounding box containing the object. The calculation method is given as

$$C = \Pr(object) \times IoU_{pred}^{truth} \tag{1}$$

where $\Pr(object)$ represents whether the object is contained in the grid. If the bounding box contains the object, then $\Pr(object) = 1$; otherwise, $\Pr(object) = 0$. The intersection over union (IoU) indicates that the bounding box contains the accuracy of objects.

$$IoU = \frac{area(ground\ truth) \cap area(prediction\ box)}{area(ground\ truth) \cup area(prediction\ box)} \tag{2}$$

The final confidence we use c to represent:

$$\begin{aligned} c &= \Pr(class_i|object) \times \Pr(object) \times IoU_{pred}^{truth} \\ &= \Pr(clsass_i) \times IoU_{pred}^{truth} \end{aligned} \tag{3}$$

Figure 1 shows that YOLOv3 uses the darknet53 network for feature extraction. This network is superimposed by the residual unit, which is more conducive to model convergence.

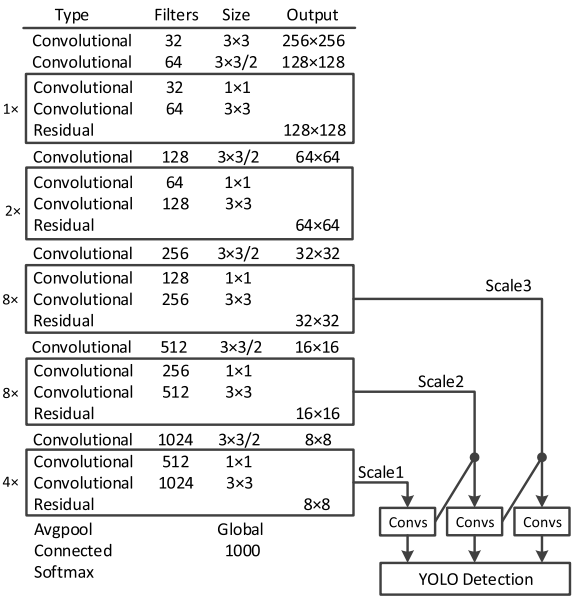

FIGURE 1. YOLOv3 network architecture. (Column 1 lists the network type, Column 2 lists the number of channels in the convolutional layer, Column 3 lists the size of the convolution kernel, and Column 4 lists the output size.)

In addition, due to the addition of the residual unit, the number of network layers can be expanded, and network feature extraction can be improved.

The introduction of the 1×1 convolution kernel in the residual module reduces the number of channels in the convolution operation. This step reduces the number of parameters in the network, thus making the entire network model weigh less, and reduces the calculation amount.

Unlike the previous version, YOLOv3 is predicted from three scale feature maps, which greatly improves detection rate of small targets.

In the detection phase, YOLOv3 adopts a full convolution method. There are two advantages to doing so: first, the network can accept input images of any size without requiring all training images and test images to be the same size; and second, the convolutional layer replaces the fully connected layer in the traditional network, greatly reducing the amount of computation.

The output of YOLOv3 is a tensor with $S \times S \times ((4 + 1) \times B \times C')$ dimensions, where $S \times S$ is the number of input images divided into grid cells; B is the number of bounding boxes predicted by each grid cell; and C' is the number of categories of detected objects.

We take the visual object classes (VOC) dataset as an example, which contains 20 object types to be detected. The network output will have 96 prediction results, which is not in line with the actual situation. We use non-maximum suppression to find the correct bounding box as described below.

In the first step, we set a certain threshold; if the confidence score of the bounding box is lower than this threshold, then the bounding box is deleted. In the second step, we sort the remaining bounding boxes by the confidence score and select the bounding box with the highest score. In the third step, the remaining bounding boxes are traversed, and the IoU between them and the highest bounding box is calculated. When the obtained IoU exceeds a certain threshold, the bounding box is deleted. In the fourth step, we continue to select the highest-margined bounding box from the unprocessed bounding boxes and repeat the above steps. The results of non-maximum suppression are shown in Figure 2.

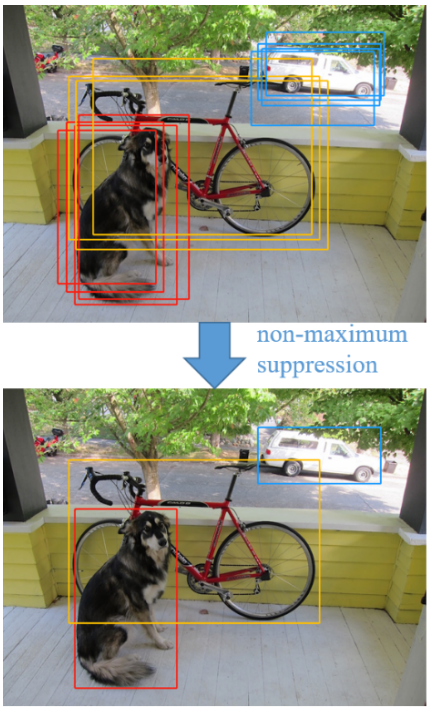

FIGURE 2. Comparison of bounding box before and after non-maximum suppression.

B. PRINCIPLES OF PEDESTRIAN TRACKING

In this section, we discuss the principles of pedestrian tracking. We use the Kalman filter and Hungarian algorithm to achieve pedestrian tracking. The role of the Kalman filter is to predict the position of the current frame pedestrian based on the position of the pedestrian in the previous frame. We use a discrete control process system to represent the position prediction process. The system can be described by a linear stochastic difference equation:

$$X(k) = A \times X(k-1) + B \times U(k) + W(k) \tag{4}$$

The measured value of the system can be expressed as

$$Z(k) = H \times X(k) + V(k) \tag{5}$$

where $X(k)$ is the system state at time k, and $U(k)$ is the amount of system control at time k. A and B are system parameters, and for multi-model systems, they are matrices. $Z(k)$ is the measured value at time k, and H is the parameter of the measurement system; for multi-measurement systems, H is a matrix. $W(k)$ and $V(k)$ represent noise in the process and measurement, respectively. They are assumed to be white Gaussian noise, and the covariance is R (assuming no variance with the state of the system).

The Kalman filter workflow is divided into two parts, namely prediction and updating. During prediction, assuming that the system state at time k is $X(k)$, and according to the system model, a certain state can first be predicted based on the previous system state:

$$X(k|k-1) = A \times X(k-1|k-1) + B \times U(k) + W(k) \tag{6}$$

where $X(k|k-1)$ is the result of the prior state prediction, and $X(k-1|k-1)$ is the result of the previous state. Second, we predict the covariance of $X(k|k-1)$ according to the covariance of $X(k|k-1)$.

$$P(k|k-1) = A \times P(k-1|k-1) \times A^{\mathrm{T}} + Q \tag{7}$$

In the update part, we first calculate the weighting matrix (e.g., Kalman gain) via

$$Kg(k) = P(k|k-1) \times H^{\mathrm{T}} / (H \times P(k|k-1) \times H^{\mathrm{T}} + R) \tag{8}$$

Then, we calculate $X(k|k)$ (the optimal estimate of k time) based on the Kalman gain obtained in the previous step.

$$X(k|k) = X(k|k-1) + Kg(k) \times (Z(k) - H \times X(k|k-1)) \tag{9}$$

Finally, we update the covariance of X$(k|k)$.

$$P(k|k) = (I - Kg(k) \times H) \times P(k|k-1) \tag{10}$$

where I is expressed as an identity matrix.

Next, we calculate the IoU_{track} between the predicted position and real position after calculating the predicted position of the current frame by the Kalman filter. The calculation method is as follows:

$$IoU_{track} = \frac{area(predicted\ position) \cap area(real\ position)}{area(predicted\ position) \cup area(real\ \ position)} \tag{11}$$

We combine the calculated IoU_{track} into a matrix and use the Hungarian matching algorithm to find the location where the front and back frames match. The specific steps are discussed in detail as below.

We take four people as an example, assuming no detection loss in the front and back frames. The matrix is shown in Figure 3(a). We use $\{J_i, i = 1, 2, 3, 4\}$ to represent the predicted position of the Kalman filter, and $\{W_i, i = 1, 2, 3, 4\}$ represents the actual position of the current frame. The calculated IoU_{track} values are in the matrix. For the convenience of calculation, all values are multiplied by 100.

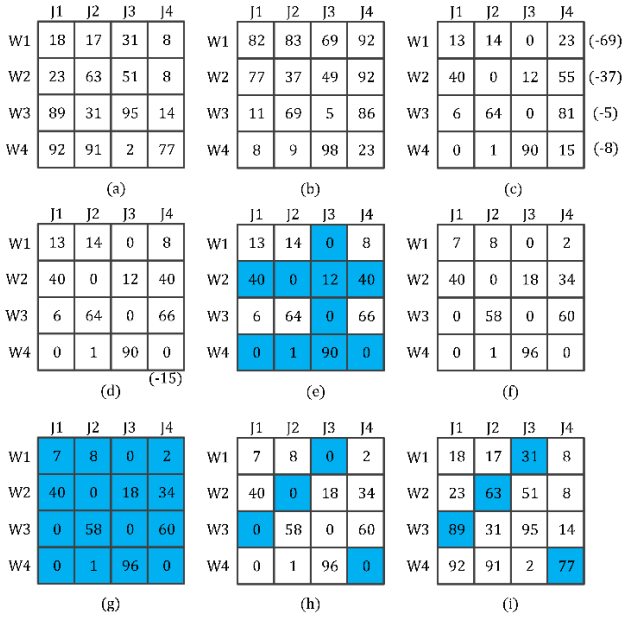

FIGURE 3. Flow of Hungarian matching algorithm.

The Hungarian matching algorithm solves the optimal solution of the assignment problem, and we want to maximize the IoU_{track} of the predicted position and actual position; as such, we first subtract 100 from the value in the matrix as shown in Figure 3(b).

Second, each row of the matrix is subtracted from the minimum value of the row, as depicted in Figure 3(c). In the third step, the minimum value of the column is subtracted from each column of the matrix [see Figure 3(d)] to ensure each row and column contains at least one zero.

In the fourth step, we find the location of all zero elements and include them with as few rows and columns as possible. As shown in Figure 3(e), we find the row containing W2 and W4 and the column containing J3 to have all zeros.

In the fifth step, we find the minimum number in the row and column not included in the fourth step. Then, we subtract the minimum number from all uncontained numbers and add the minimum number in the fourth step where the row and column overlap; see Figure 3(f). We repeat steps 4 and 5 until the number of rows required to contain all zeros is equal to the size of the matrix, which represents the optimal distribution of zeros in the matrix, as shown in Figure 3(g).

In the sixth step, we find the location of all zeros whose rows and columns do not coincide, as shown in Figure 3(h). These locations correspond to the largest IoU_{track} allocation in the original matrix as indicated in Figure 3 (i). W1 is paired with J3, W2 is paired with J2, W3 is paired with J1, and W4 is paired with J4.

Through the above method, we can match the pedestrian between the previous frame and the current frame, thus achieving pedestrian tracking.

III. RESULTS OF EXPERIMENT

In this section, we introduce the experimental process. The corresponding flow chart is shown in Figure 4.

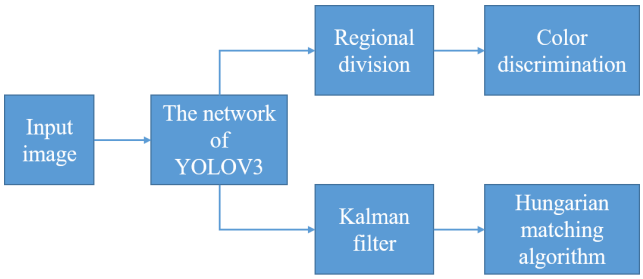

FIGURE 4. Flow chart of experiment.

A. PREPARING THE DATASET

As the project needs are unique and no public datasets are available, we must develop the dataset ourselves and integrate it into the existing dataset. According to an analysis of project needs, we need three types of data: people, helmets and vests.

First, by using the annotation tool, we create image annotations and box out the areas of the image that contain pedestrians, helmets, and vests.

In the second step, we determine that the coco dataset contains only humans without helmets and vests. Hence, we use a script to filter out images of pedestrians in the coco dataset and find the corresponding annotations.

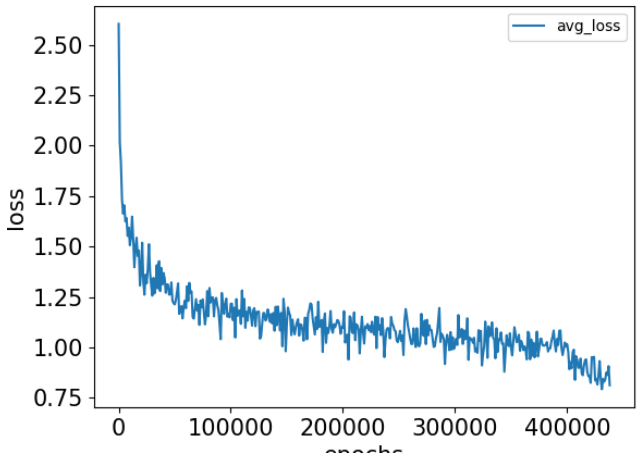

FIGURE 5. Loss curves of neural network model.

The third step involves modifying the annotation file to meet the following format.

$$\langle \text{object-class} \rangle \langle x \rangle \langle y \rangle \langle w \rangle \langle h \rangle$$

where object-class is the index of the class; x, y are the horizontal and vertical ordinates of the center of the region of interest (ROI), respectively; and w and h are the width and height of the ROI, respectively. x, y, w, h are relative to the size of the entire image.

The last step is to check whether the annotation file matches image and then delete the unmatched file.

B. TRAINING THE NEURAL NETWORK MODEL

We pre-train the first 74 layers of weight in the darknet53 network on the ImageNet dataset, which allows the network to extract image features more effectively. Since the dataset we produced is much smaller than ImageNet, we fixed the first 74 layers of the network after pre-training. We only train the next 32 layers parameters. After the training results converge, we fine-tune all the parameters.

To prevent the training results from diverging in the training process, the system automatically adjusts the learning rate. We find that simultaneous training with multiple GPUs can greatly improve the training speed but may lead to a lower loss rate, higher recall rate, and a final weight that cannot predict the bounding box of the object. After many trials, the following solution emerges. Before using multiple GPUs for training, we use a single GPU and switch to multiple GPUs when the 'Obj' (the parameter indicating whether the system detects an object) starts to rise steadily in the training result.

The loss curve of the neural network model is illustrated in Figure 5. Our training samples combine multiple datasets, and the number of training samples reaches 100,000. However, from the final training results, our network converges, and the actual test results were quite good.

C. PROCESSING OF TEST RESULTS

Using the neural network model from the above training, we could detect the positions of pedestrians, vests, and helmets. To determine the vest color, we convert the color space of the vest area; specifically, we convert the vest area into the HSV color space, set a certain threshold to select the red pixel area, and binarize the obtained area. After entering corrosion expansion, we calculate the ratio of white pixels to all pixels to determine vest color. In addition, if any pedestrians are not wear a helmet, we warn them.

Using the pedestrian position information, we employ the Kalman filter to predict the pedestrian position in the current frame based on the pedestrian position in the previous frame. We calculate the IoU of the predicted pedestrian position and the position of the current frame. The above-mentioned

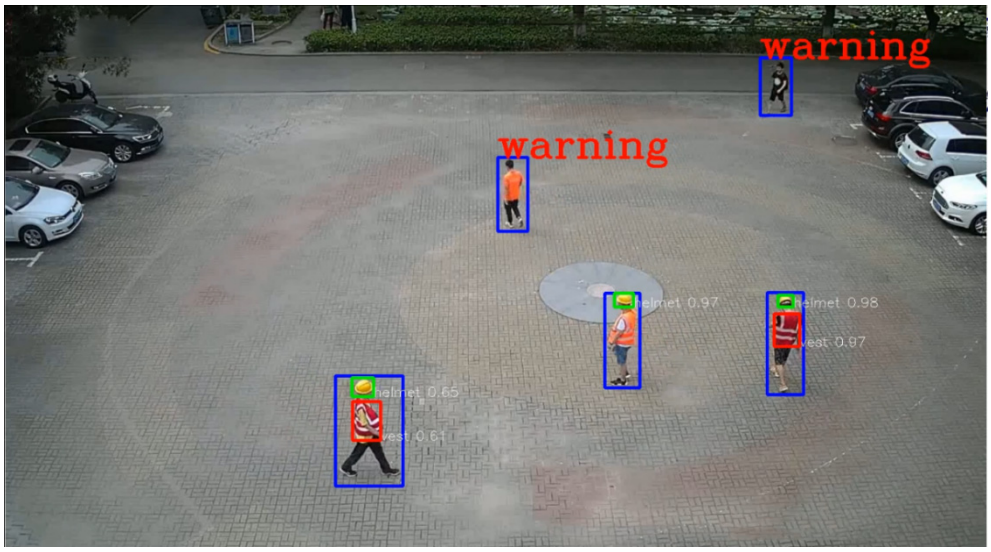

FIGURE 6. Results of safety officer detection.

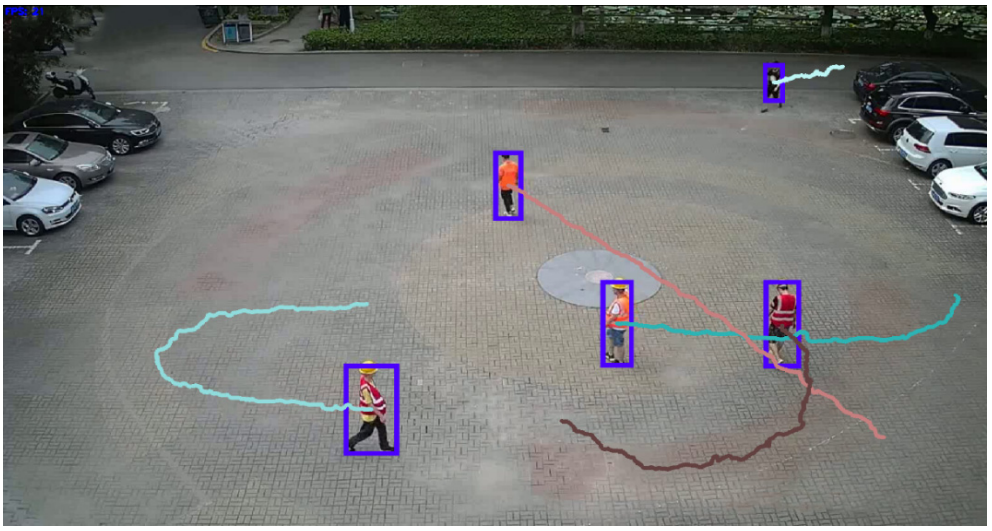

FIGURE 7. Results of pedestrian tracking.

Hungarian matching algorithm can be applied in match pedestrians before and after the frame, thereby accomplishing pedestrian tracking.

D. ACCURACY OF THE RESULT

This experiment is accelerated by GPU, and the calculating experiment is a NVIDIA GeForce GTX 1080Ti. Specific experimental accuracy is shown in TABLE 1. The actual effects of security officer detection and pedestrian tracking are pictured in Figure 6 and Figure 7.

TABLE 1. Average precision of different IoU.

IoU \ Category	Pedestrian	Helmet	Vest
0.5	0.89	0.84	0.94
0.7	0.7	0.23	0.4

IV. CONCLUDING REMARKS

In this paper, we have proposed a method for safety officer detection and tracking based on deep learning. The detection speed is 18 frames per second, which is close to the real-time requirement. The proposed method can save substantial labor costs. At many construction sites, detection objects may be sparsely distributed; thus, sparse signal processing techniques may provide some feasible solutions [51]–[58].

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.
[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, p. 1.
[3] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.
[4] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.
[5] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.
[6] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.
[7] X. Sun, G. Gui, Y. Li, R. P. Liu, and Y. An, "ResInNet: A novel deep neural network with feature reuse for Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 679–691, Feb. 2019.
[8] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389, 2017.
[9] Y. Xin *et al.*, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.
[10] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink MIMO," *IEEE Access*, vol. 7, pp. 7599–7605, 2018.
[11] M. Liu, T. Song, and G. Gui, "Deep cognitive perspective: Resource allocation for NOMA based heterogeneous IoT with imperfect SIC," *IEEE Internet Things J.*, to be published. doi: 10.1109/JIOT.2018.2876152.
[12] N. Zhang, S. Zhang, J. Zheng, X. Fang, J. W. Mark, and X. Shen, "QoE driven decentralized spectrum sharing in 5G networks: Potential game approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7797–7808, Sep. 2017.
[13] N. Zhang, N. Lu, N. Cheng, J. W. Mark, and X. S. Shen, "Cooperative spectrum access towards secure information transfer for CRNs," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2453–2464, Nov. 2013.
[14] N. Zhang, H. Liang, N. Cheng, Y. Tang, J. W. Mark, and X. S. Shen, "Dynamic spectrum access in multi-channel cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 11, pp. 2053–2064, Nov. 2014.
[15] B. Dhingra *et al.*(2017). "Towards end-to-end reinforcement learning of dialogue agents for information access." [Online]. Available: https://arxiv.org/abs/1609.00777
[16] X. Li, T. Qin, J. Yang, and T.-Y. Liu. (2016). "LightRNN: Memory and computation-efficient recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1610.09893
[17] A. Vaswani *et al.* (2017). "Attention is all you need." [Online]. Available: https://arxiv.org/abs/1706.03762
[18] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Jun. 2016, pp. 260–270.
[19] M. White, C. Vendome, M. Linares-Vásquez, and D. Poshyvanyk, "Toward deep learning software repositories," in *Proc. 12th Work. Conf. Mining Softw. Repositories*, May 2015, pp. 334–345.

[20] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Empirical Methods Natural Lang. Process.*, Oct. 2014, pp. 1746–1751.
[21] F. Zhu *et al.*, "Image-text dual neural network with decision strategy for small-sample image classification," *Neurocomputing*, vol. 328, pp. 182–188, Feb. 2019.
[22] K. Zhao, H. Zhang, Z. Ma, Y.-Z. Song, and J. Guo, "Multi-label learning with prior knowledge for facial expression analysis," *Neurocomputing*, vol. 157, pp. 280–289, Jun. 2015.
[23] X. Li *et al.*, "Supervised latent Dirichlet allocation with a mixture of sparse softmax," *Neurocomputing*, vol. 312, pp. 324–335, Oct. 2018.
[24] J. Xiong, X. Long, R. Shi, M. Wang, J. Yang, and G. Gui, "Background error propagation model based RDO in HEVC for surveillance and conference video coding," *IEEE Access*, vol. 6, pp. 67206–67216, 2018.
[25] R. Zhu, Z. Wang, Z. Ma, G. Wang, and J.-H. Xue, "LRID: A new metric of multi-class imbalance degree based on likelihood-ratio test," *Pattern Recognit. Lett.*, vol. 116, pp. 36–42, Dec. 2018.
[26] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3FD: Single shot scale-invariant face detector," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 192–201.
[27] K. Lenc and A. Vedaldi, "Understanding Image Representations by Measuring Their Equivariance and Equivalence," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 991–999.
[28] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?—Weakly-supervised learning with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 685–694.
[29] R. Theagarajan, F. Pala, X. Zhang, and B. Bhanu, "Soccer: Who has the ball? Generating visual analytics and player statistics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 1830–18308.
[30] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[31] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[32] J. Pan, Y. Yin, J. Xiong, W. Luo, G. Gui, and H. Sari, "Deep learning-based unmanned surveillance systems for observing water levels," *IEEE Access*, vol. 6, pp. 73561–73571, 2018.
[33] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, May/Jun. 2017, pp. 3357–3364.
[34] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Proc. Conf. Robot Learn.*, Nov. 2017, pp. 262–270.
[35] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, 2017.
[36] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2015.
[37] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Automat.*, May/Jun. 2017, pp. 3389–3396.
[38] L. Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wang, "Deep learning based improved classification system for designing tomato harvesting robot," *IEEE Access*, vol. 6, pp. 67940–67950, 2018.
[39] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[40] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 379–387.
[41] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2016, pp. 21–37.
[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.
[43] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int. J. Comput. Vis.*, vol. 120, no. 2, pp. 215–232, 2016.
[44] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
[45] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.
[46] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1309–1321, Aug. 2015.
[47] J. Han *et al.*, "Representing and retrieving video shots in human-centric brain imaging space," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2723–2736, Jul. 2013.
[48] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1–6.
[49] R. Faragher, "Understanding the basis of the Kalman filter via a simple and intuitive derivation," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 128–132, Sep. 2012.
[50] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics*, vol. 52, no. 1, pp. 7–12, 2005.
[51] Y. Li *et al.*, "Sparse adaptive iteratively-weighted thresholding algorithm (SAITA) for Lp-regularization using the multiple sub-dictionary representation," *Sensors*, vol. 17, no. 12, pp. 2920–2936, 2017.
[52] Y. Li, X. Cheng, and G. Gui, "Co-robust-ADMM-net: Joint ADMM framework and DNN for robust sparse composite regularization," *IEEE Access*, vol. 6, pp. 47943–47952, 2018.
[53] Y. Li, Y. Lin, X. Cheng, Z. Xiao, F. Shu, and G. Gui, "Nonconvex penalized regularization for robust sparse recovery in the presence of $S\alpha S$ noise," *IEEE Access*, vol. 6, pp. 25474–25485, 2018.
[54] Y. Li *et al.*, "MUSAI-$L_{1/2}$: MUltiple sub-wavelet-dictionaries-based adaptively-weighted iterative half thresholding algorithm for compressive imaging," *IEEE Access*, vol. 6, pp. 16795–16805, 2018.
[55] J. Wen, J. Wang, and Q. Zhang, "Nearly optimal bounds for orthogonal least squares," *IEEE Trans. Signal Process.*, vol. 65, no. 20, pp. 5347–5356, Oct. 2017.
[56] J. Wen, Z. Zhou, Z. Liu, M.-J. Lai, and X. Tang, "Sharp sufficient conditions for stable recovery of block sparse signals by block orthogonal matching pursuit," *Appl. Comput. Harmon. Anal.*, to be published. doi: 10.1016/j.acha.2018.02.002.
[57] J. Wen, Z. Zhou, D. Li, and X. Tang, "A novel sufficient condition for generalized orthogonal matching pursuit," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 805–808, Apr. 2017.
[58] F. Wen, P. Liu, Y. Liu, R. C. Qiu, and W. Yu, "Robust sparse recovery in impulsive noise via ℓ_p-ℓ_1 optimization," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 105–118, Jan. 2017.

YU ZHAO (S'18) is currently pursuing the master's degree in communication engineering with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include dictionary learning, deep learning, and convex optimization.

QUAN CHEN (S'18) is currently pursuing the master's degree in communication engineering with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include dictionary learning, deep learning, and convex optimization.

WENGANG CAO (S'18) is currently pursuing the master's degree in communication engineering with the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include dictionary learning, deep learning, and convex optimization.

JIE YANG received the B.Sc., M.Sc., and Ph.D. degrees in communication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, 2006, and 2018, respectively, where she is currently an Assistant Professor.

JIAN XIONG (M'17) received the B.Sc. degree in computer science and technology from the Anhui University of Finance and Economics, Bengbu, China, in 2007, the M.Sc degree in computer application technology from Xihua University, Chengdu, China, in 2010, and the Ph.D. degree in single and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2015. In 2014, he was a Research Assistant with the Image and Video Processing Laboratory, The Chinese University of Hong Kong, Hong Kong. Since 2015, he has been a Lecturer with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include image and video coding, pattern recognition, and machine learning.

GUAN GUI (M'11–SM'17) received the Dr. Eng degree in information and communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2012.

From 2009 to 2014, he was with the Wireless Signal Processing and Network Laboratory (Prof. Fumiyuki Adachi Laboratory), Department of Communications Engineering, Graduate School of Engineering, Tohoku University, as a Research Assistant and a Postdoctoral Research Fellow. From 2014 to 2015, he was an Assistant Professor with the Department of Electronics and Information System, Akita Prefectural University. Since 2015, he has been a Professor with the Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China. He is currently engaged in research of deep learning, compressive sensing, and advanced wireless techniques. He has published more than 200 international peer-reviewed journal/conference papers. He received the Member and Global Activities Contributions Award from the IEEE ComSoc and seven best paper awards, i.e., ICEICT 2019, ADHIP 2018, CSPS 2018, ICNC 2018, ICC 2017, ICC 2014, and VTC 2014-Spring. He was also selected as for Jiangsu Specially Appointed Professor, in 2016, the Jiangsu High-level Innovation and Entrepreneurial Talent, in 2016, the Jiangsu Six Top Talent, in 2018, the Nanjing Youth Award, in 2018, and the 1311 Talent Plan of NJUPT, 2017. He was an Editor of *Security and Communication Networks*, from 2012 to 2016, and the *Journal of Communications*, in 2019. He has been an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, since 2017, the IEEE ACCESS, since 2018, and *KSII Transactions on Internet and Information Systems*, since 2017, and the Editor-in-Chief of *EAI Transactions on Artificial Intelligence*, since 2018.