Received September 26, 2019, accepted October 14, 2019, date of publication October 23, 2019, date of current version November 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949106

Simulation of Evacuating Crowd Based on Deep Learning and Social Force Model

XIN LI, YANCHUN LIANG, MINGHAO ZHAO, CHONG WANG, HONGTAO BAI, AND YU JIANG
College of Computer Science and Technology, Jilin University, Changchun 130012, China
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

Corresponding author: Yu Jiang (jiangyu2011@jlu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 51679105, Grant 61872160, Grant 51809112, and Grant 51939003.

ABSTRACT It is always difficult to evacuate crowds in public places like subway stations. The traditional crowd behavior simulation models often ignore two important issues in crowd evacuation: pedestrian tracking and individual differences. To solve the problem, this paper combines social force model (SFM) with deep learning into a novel pedestrian detection method. Firstly, several deep learning algorithms for pedestrian detection were compared, and the best ones for sparse and dense crowds were determined. Next, the pedestrian positions in a real video were acquired by the selected algorithms, and converted into actual coordinates in the scene. Then, the evacuation process was simulated with our method and the SFM based on these coordinates. The results show that our model output closer-to-reality results than the SFM. The research findings shed important new light on evacuation in crowded areas.

INDEX TERMS Deep learning, social force model (SFM), crowd simulation.

I. INTRODUCTION

Many megacities have appeared across the globe for two intertwined reasons: the population boom and the accelerated urban growth. There are now 36 cities with over 10 million residents around the world. Tokyo, for instance, is home to 37 million citizens. Meanwhile, the Chinese mainland boasts 15 cities with a population in excess of 10 million. With city's development, the urban traffic environment is becoming more complex. Public places, like subway stations and bus stops can be very crowded. People usually enter and exit these places in order. However, accidents like crowding and trampling may happen in emergencies (e.g. fire or earthquake). It may lead to greater congestion, or more severe accidents, due to large gatherings of people at transport hubs. To design rational evacuation routes and strategies, many scholars have attempted to develop suitable models for real-world scenes and crowd behaviors [1]–[5].

To simulate crowd behaviors, Helbing et al. analyzed crowd panic in details, and put forward the social force model (SFM) [6]. Considering the discreteness of pedestrian flow, this model attributes the dynamic feature of pedestrian flow to the interaction force between individuals. Since its birth, the SFM has been continuously optimized. Nevertheless, neither the original nor the optimized models take account of individual factors like height and strength. Moreover, these models ignore the mentality of individuals in sparse crowds.

To make up for these gaps, this paper aims to predict the speed and mentality of individuals in sparse crowds through pedestrian detection [7]–[9]. To this end, the existing deep learning algorithms were adopted to detect and locate pedestrians quickly, and the SFM was introduced to simulate the real evacuation process. The contributions of our research are summed up as follows:

1. The pedestrians were detected by deep learning, and their speeds and accelerations were computed based on their trajectories. Several pedestrian detection algorithms were compared, and the one with the best effects on both sparse and dense crowds was selected for transfer learning on our dataset, with the aim to improve the detection accuracy. Next, the center of gravity of the crowd was pinpointed by classification algorithms in deep learning. Then, the specific position of an individual in the scene was derived from the center of gravity and boundaries of the crowd, thus forming the trajectory of each pedestrian.

2. Some individuals in the crowd are blocked and not captured by camera. For these individuals, their speeds and accelerations were also computed based on their trajectories, and

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

substituted into the SFM to predict the next positions. Pedestrian detection and trajectory calculation were introduced to solve the defects of the SFM in dealing with individual differences and random events. In return, the SFM enhances the computing accuracy of deep learning for dense crowd. The combination of deep learning and the SFM ensures the effectiveness and accuracy of the designed evaluation process.

The remainder of this paper is organized as follows: Section II introduces deep learning, the SFM and the coupling between them; Section III sets up our emergency evacuation strategy for crowded areas; Section IV verifies our strategy through simulation and comparative analysis; Section V puts forward the conclusions of this research.

II. RELATED WORK

A. EVACUATION MODEL

In 1971, Fruin [10] was the first to compare the movement of a crowd to the flow of fluid. In 1989, May noted the analogy between pedestrian flow and vehicle flow, and proposed the concepts of subjective and micro-pedestrian movements. In 1982, Hill suggested that pedestrians are unconscious when choosing their paths to the target, and that the routing strategy directly hinges on the length and complexity of each path. Later, it was found that the path selection could be affected by distance [11]–[15], the crowding degree [11], [13], [14], [16], cooperation, selfish behavior [17] and conformity behavior [18]. In 2009, Kretz [11] added an additional element in a model of pedestrian dynamics that makes the agents deviate from the rule of the shortest path. In 2010, Hartmann [12] proposed an approach correspond to the shortest distances to the pedestrian's targets with respect to arbitrary metrics by adopting a continuum perspective, navigation fields. It offers an easily adaptable framework for realistic navigation of single pedestrians as well as crowds in microscopic approaches to pedestrian dynamics. In 2011, Guo and Huang [13] proposed a method for formulating the route choice behavior of pedestrians in evacuation in closed areas with internal obstacles, which can simulate two classes of phenomena that cannot be reproduced accurately by those existing methods.. In 2012, Bovy and Stem [19] held that individuals in an evacuating crowd might not find the fastest evacuation route, and that the evacuation time of selfish individuals is shorter than that of cooperative ones. In 2017, Haghani and Savri [14] made a report on discrete-choice estimates derived from observations of SC and RC methods and found the potential applicability of the SC methods in virtual-reality decision experiments.

Most models on pedestrian dynamics are grounded on individual trajectory. The most popular ones include gas dynamic model and pedestrian network flow model. In 2002, Hughes model [20] was developed to describe and predict the 2D macro-features of crowd movement. It is shown that pedestrians, such as pilgrims, aim at achieving each immediate goal in minimum time rather than achieving all goals in overall minimum time. In 2015, Jiang *et al.* [21] built a high-order macro model for bidirectional pedestrian flow, which can reproduce the self-organizing stratification of the flow. In 2016, Carrillo *et al.* [22] improved the Hughes model for the smooth turning and temporary waiting of pedestrians.

The SFM is the most representative pedestrian flow model [2], [3], [23]–[27]. In 2000, Helbing *et al.* [23] simulate the evaluation of a panic crowd with the SFM, drawing on the self-driven multiparticle system. Considering the impacts of conformity behavior and panic, these scholars concluded that individuals moving faster than usual tend to fall and get injured, forming obstacles that slow down the crowd evacuation and affect the use of emergency exit. In 2015, Johansson *et al.* [25] improved the SFM to describe the pedestrians in waiting situation, and studied their interaction with non-waiting pedestrians. They showed that the treatment of waiting pedestrians had a significant impact on simulations of pedestrian traffic. They introduced a series of extensions to the social force model to produce waiting behavior, presented a sensitivity analysis and provided sufficient criteria for stability. Li *et al.* [26] found that the evacuation speed of crowded escalators is slower than the mean walking speed of pedestrians. To reduce the congestion risk, the speed of the escalator should be set slower than the average speed of the pedestrians, and conductors can be employed to divert the traffic at the entrance, turns, and exit of the escalator. In 2018, Liu *et al.* [27] simulated and analyzed emergency evacuation with video data model. They quantified the relationship values among pedestrians, and group force was added to the primary social force model. The proposed approach was consistent with real-world situations and could assist in analyzing emergency evacuation scenarios.

B. PEDESTRIAN DETECTION BASED ON DEEP LEARNING

Many deep learning methods have been applied to pedestrian detection, which aims to identify all pedestrians in each frame of images or videos.

One of the most frequently applied deep learning methods is the region-based convolutional neural network (R-CNN) [28]–[30]. The R-CNN trains CNNs end-to-end to classify the region proposals into object categories or background. In 2013, Szegedy *et al.* [31] introduced deep neural networks (DNNs) to object detection problem, and successfully classified and pinpointed the objects, which achieves state-of-the-art performance on Pascal 2007 VOC. In 2014, Sermanet *et al.* [32] proposed an image positioning method that simultaneously categorizes, locates and detects objects through a single CNN. But it needs a huge amount of computation. The computing load of the R-CNN was greatly reduced in the Fast R-CNN [34], and further lowered in Faster R-CNN [29], [35]. The Fast R-CNN was inspired by spatial pyramid pooling (SPP) [33], while the Faster R-CNN draws on the region proposal network (RPN).

In addition, single-shot detector (SSD) [36], you look only once (YOLO) algorithm [37] and CornerNet [38] have also been employed to detect pedestrians. Without needing to

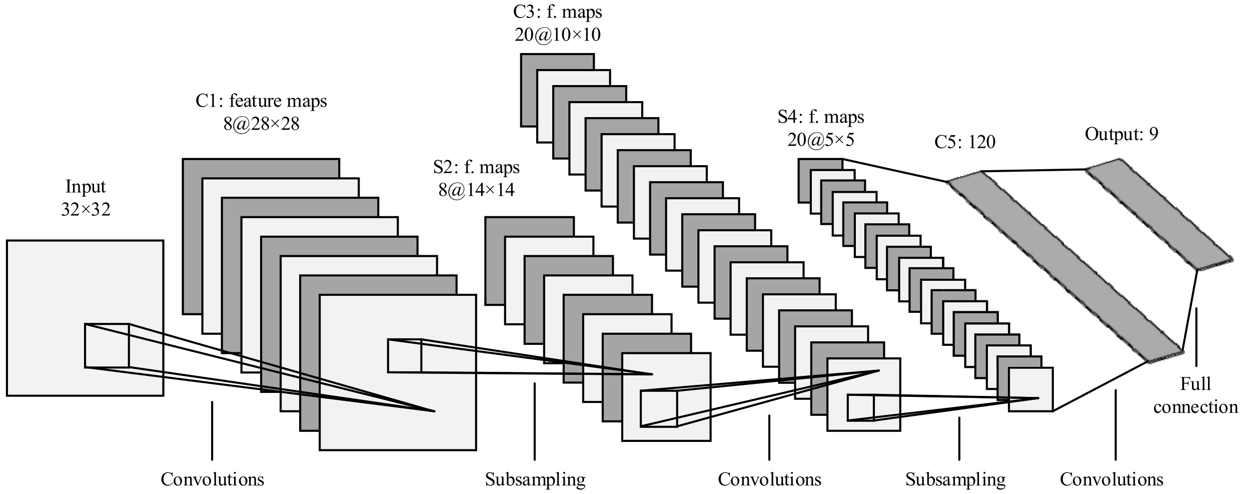


FIGURE 1. The typical structure of the CNN.

generate proposals, the SSD can achieve comparable accuracy more efficiently than two-shot detectors. The YOLO obtains bounding boxes and the class probability of each box with a single neural network, but faces high coordinate errors. Compared with the SSD, CornerNet removes the set of anchors. By detecting objects as paired keypoints, Law and Deng [38] eliminated the need for designing a set of anchor boxes commonly used in prior single-stage detectors. In addition, a new type of pooling layer helps the network better localize corners, which CornerNet achieved a 42.2% AP on MS COCO.

III. METHODOLOGY

This paper extracts the actual position of pedestrians by the deep learning algorithm Faster R-CNN. The SFM-based simulation focuses on the positions of visible and invisible pedestrians. The pedestrian behaviors were simulated by connecting the visible and invisible positions of the same individual in all frames into the trajectory of that person. This section begins with basic convolutional neural network for deep learning. Then, the improvement of the CNN by the objection recognition algorithm Faster R-CNN, and the original social force model are introduced. Finally, the simulation algorithm combining social force model with deep learning is proposed.

A. CONVOLUTIONAL NEURAL NETWORK

The CNN is a feedforward network that performs well in largescale images. The main operations of the CNN include convolution, pooling and full connection. Specifically, convolution is to extract features from the input in two steps: dividing the original image into several small regions, and acquiring the relevant features from each region; pooling compresses the image to save the main features and reduce the number of parameters; full connection connects every neuron in layer i to a neuron in layer i+1. The typical structure of the CNN is shown in Figure 1 below.

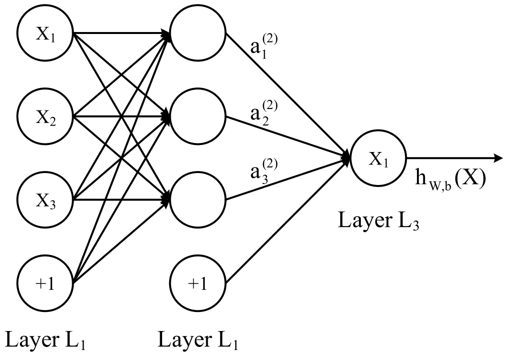


FIGURE 2. The structure of the DNN.

As shown in Figure 1, the CNN usually has several convolutional layers. The first convolutional layer extracts low-level features, the second extracts mid-level features, and the third extracts high-level features. Each convolutional layer is followed by a pooling layer, such that feature extraction and image compression are repeated until most feature are high-level ones. The features on the last layer are used for tasks like classification and regression. The fully-connected layer often adopts the DNN structure, and outputs the classification result using the Softmax activation function. The DNN structure is presented in Figure 2 below.

In Figure 2, every neuron is connected to all neurons in the previous layer. These neurons satisfy the linear relationship $z = \sum w_i x_i + b$, where z is the output of the current neuron, w is the weight, x is the output of neurons in the previous layer and b is the bias value.

B. FASTER R-CNN - OBJECT RECOGNITION

The CNN is suitable for analyzing an image containing a single object [39], [40]. If the image contains multiple objects, it should be split into multiple parts. This process may produce a heavy computing load. To solve the problem, the RPN needs to be introduced to create the region of interest (ROI) for target detection.

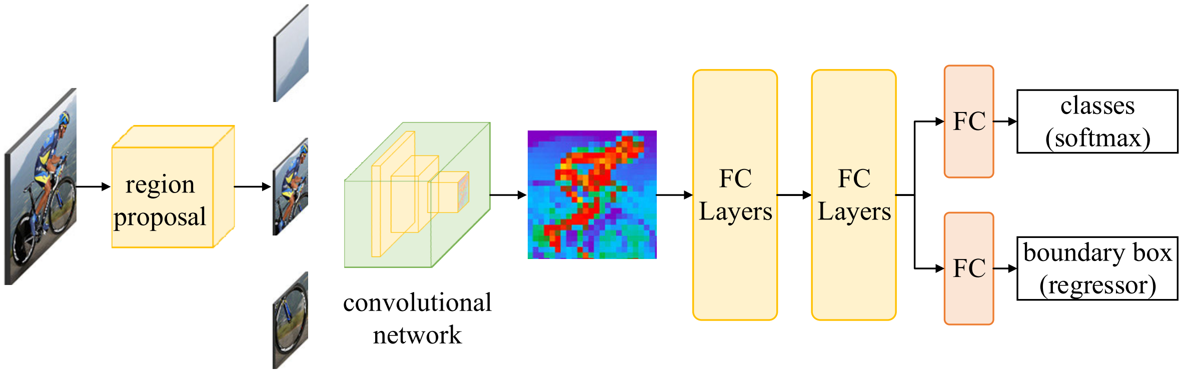


FIGURE 3. The structure of the R-CNN.

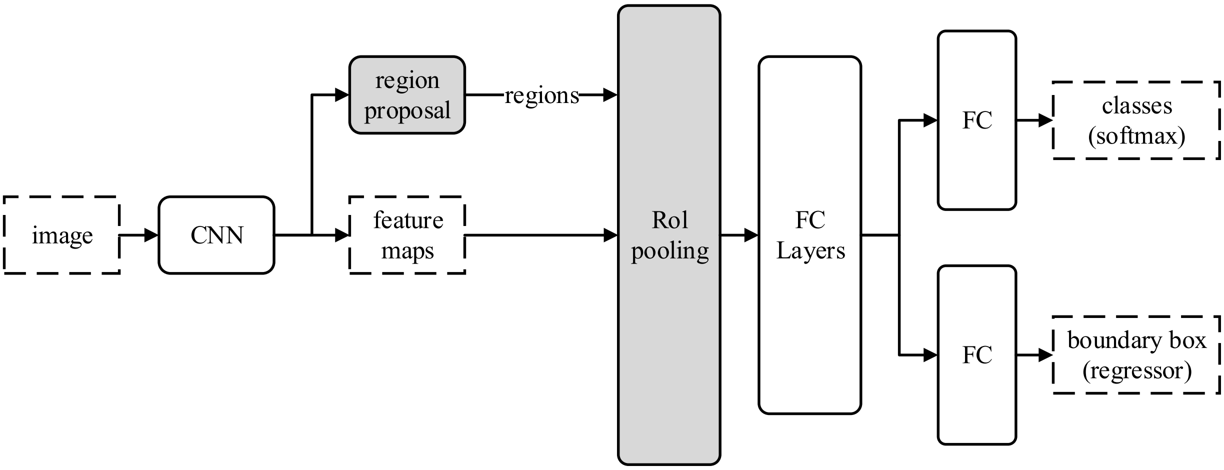


FIGURE 4. The structure of Fast R-CNN.

Firstly, the texture of each pixel is computed, and the two pixels with the most similar texture are combined into the same group. Then, the groups with similar texture are combined into a region proposal. The resulting ROIs are converted into fix-sized images, and imported to the CNN. Several fully-connected layers are added to classify targets and extract the bounding box. The resulting R-CNN is illustrated in Figure 3 below.

The accuracy of the R-CNN is positively correlated with the number of region proposals, many of which are overlapping with each other [28]. Suppose there are 2,000 region proposals to be inputted into the R-CNN independently. Then, the feature extraction needs to be performed 2,000 times for different ROIs. As a result, the R-CNN is very slow in training and inference.

The above defect can be overcome by the Fast R-CNN (Figure 4). The Fast R-CNN extracts the features of the entire image, rather than acquire the features part by part [34]. Then, the region proposals are created directly on the feature map.

As shown in Figure 4, Fast R-CNN places the feature extraction procedure with heavy computing load after convolution operations, and thus achieves a faster speed than the R-CNN. However, Fast R-CNN mainly relies on exterior region proposals like selective search.

Faster R-CNN retains the design of Fast R-CNN, but replaces the region proposals with a deep network. The new RPN is more efficient in creating the ROIs. The structure of Faster R-CNN is shown in Figure 5.

As shown in Figure 5, Faster R-CNN enjoys high detection accuracy, thanks to the removal of the feature map [29]. Therefore, Faster R-CNN is selected for object detection in this research.

C. SOCIAL FORCE MODEL

In 1995, the SFM was established by Helbing et al. according to Newton's second law of motion [6]. The model assumes that pedestrians are influenced by three factors: mentality, other pedestrians and environment. The influences of the three factors can be respectively measured by three forces: the self-driving force, the acting force between two pedestrians and the acting force between a pedestrian and obstacles. The resultant force of the three forces is the acceleration of each pedestrian. The force pattern and speed of each pedestrian in the SFM are presented in Figure 6, where $\vec{f_0}$ is the self-driving force of pedestrian i, $\vec{f_{ij}}$ is the acting force between pedestrians i and j, $\vec{f_{iw}}$ is the acting force between pedestrian i and the wall, $\vec{v}_i$ is the actual speed of pedestrian i and $v_i^0\vec{e}_i^0$ is the speed in the direction towards the target.

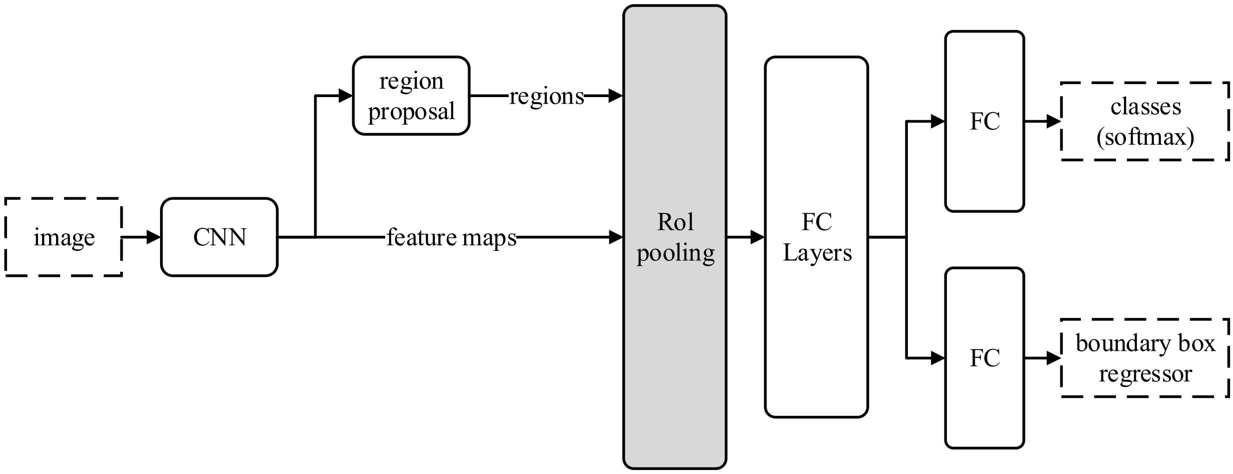


FIGURE 5. The structure of Faster R-CNN.

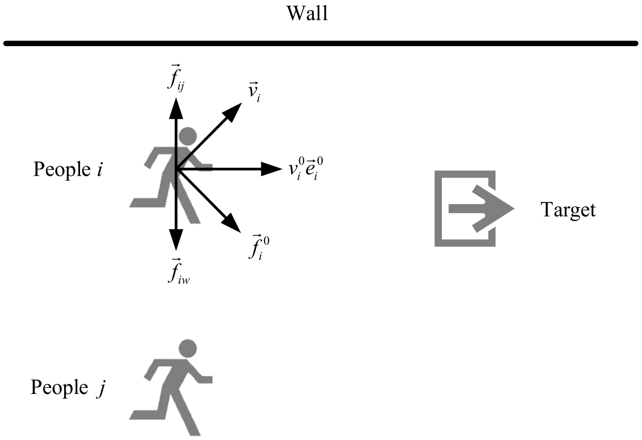


FIGURE 6. The force and speed of each pedestrian based on the SFM.

The acceleration of pedestrian i can be computed by:

$$m_i\frac{d\vec{v}_i}{dt}=\vec{f}_i^0+\sum_{j(\neq i)}\vec{f}_{ij}+\sum_{w}\vec{f}_{iw} \tag{1}$$

where m_i is the weight of the pedestrian.

The self-driving force of pedestrian i can be defined as:

$$\vec{f}_i^0=m_i\frac{v_i^0(t)\vec{e}_i^0-\vec{v}_i(t)}{\tau_i} \tag{2}$$

where $v_i^0(t)$ and $\vec{v}_i(t)$ are the expected speed and actual speed of pedestrian i at time t, respectively; τ_i is the time for pedestrian i to avoid colliding into other pedestrians and obstacles, i.e. the time for him/her to adjust speed.

In 2005, Helbing et al. pointed out that anxiety directly bears on the expected speed of each pedestrian, and defined the anxiety as [23]:

$$n_i(t)=1-v_{id}(t)/v_i^0(0) \tag{3}$$

where $v_{id}(t)$ is the expected speed of pedestrian i in the direction towards the target; $v_i^0(0)$ is the initial value of the expected speed. The initial value can be computed by:

$$v_i^0(t)=(1-n_i(t))v_i^0(t)+n_i(t)v_i^{max} \tag{4}$$

where v_i^{max} is the maximum speed of the pedestrian under the circumstance.

The acting force between two pedestrians $\vec{f}_{ij}$ is the resultant of the repulsion $\vec{f}_{ij}^s$ and the contact force $\vec{f}_{ij}^p$ between them. The acting force and the repulsion can be respectively computed by:

$$\vec{f}_{ij}=\vec{f}_{ij}^s+\vec{f}_{ij}^p \tag{5}$$

$$\vec{f}_{ij}^s=A_i\exp\ [(r_{ij}-d_{ij})/B_i]\vec{n}_{ij}\left(\lambda_i+(1-\lambda_i)\frac{1+\cos\varphi_{ij}}{2}\right) \tag{6}$$

where A_i and B_i are two constants; r_{ij} is the sum of the radius of pedestrian i and that of pedestrian j; $\vec{n}_{ij}=(\vec{r}_i-\vec{r}_j)\big/d_{ij}$; $\lambda_i\in(0,1)$ is the effect of visual anisotropy of moving pedestrians; $\cos(\varphi_{ij})=-\vec{n}_{ij}\cdot\vec{e}_i$. The contact force $\vec{f}_{ij}^p$ can be further divided into extrusion and friction:

$$\vec{f}_{ij}^p=\vec{f}_{ij}^{p_1}+\vec{f}_{ij}^{p_2} \tag{7}$$

$$\vec{f}_{ij}^p=kg(r_{ij}-d_{ij})\vec{n}_{ij}+\kappa g(r_{ij}-d_{ij})(\vec{v}_j-\vec{v}_i)\vec{t}_{ij} \tag{8}$$

where κ is the friction coefficient; g is the gravitational acceleration; $\vec{t}_{ij}=(-n_{ij}^2,n_{ij}^1)$.

The acting force between a pedestrian and the wall $\vec{f}_{iw}$ can be calculated similarly to the acting force $\vec{f}_{ij}$ between pedestrians i and j:

$$\vec{f}_{iw}=\vec{f}_{iw}^s+\vec{f}_{iw}^p \tag{9}$$

$$\vec{f}_{iw}^s=A_w\exp\ [(r_{iw}-d_{iw})/B_w]\vec{n}_{iw} \tag{10}$$

$$\vec{f}_{iw}^p=\vec{f}_{iw}^{p_1}+\vec{f}_{iw}^{p_2} \tag{11}$$

$$\vec{f}_{iw}^p=kg(r_{ij}-d_{iw})\vec{n}_{iw}+kg(r_{iw}-d_{iw})(-\vec{v}_i\cdot\vec{t}_{iw})\vec{t}_{iw} \tag{12}$$

where $\vec{f}_{iw}$ is the force applied by the wall to pedestrian i; $\vec{f}_{iw}^s$ is the subjective repulsion of pedestrian i; $\vec{f}_{iw}^p$ is the contact force between pedestrian i and the wall; $\vec{f}_{iw}^{p_1}$ is the extrusion force applied by the wall to pedestrian i; $\vec{f}_{iw}^{p_2}$ is the friction between the wall and pedestrian i.

D. EVACUATION BEHAVIOR SIMULATION

In the principle of Faster R-CNN, the coordinates of pedestrian pixels in the original image were obtained, and converted to scene coordinates. Taking them as inputs, deep learning and SFM were combined into Algorithm 1 below.

Algorithm 1

```
Input: current frame, $P_L$ and $v_L$
Output: $P_N$ and $v_N$
Initialize the scene, including exits, entrances, walls and obstacles.
$P_N \leftarrow []$. $v_N \leftarrow []$
$i \leftarrow 0$
$P_{LC} \leftarrow P_L$
Obtain the positions $P_S$ of all pedestrians from deep learning.
Convert $P_S$ into $P_C$.
for each pedestrian $k$ in $P_C$ do
    $P_D \leftarrow \left\{ P_l \in P_L \,\middle|\, \left|\overrightarrow{P_lP_k}\right| < d \right\}$
    $P_N[i] \leftarrow P_k$
    if $P_D$ is non-empty
    then
        if there is a point $P_j$ in the set $P_D$ and the angle of $\overrightarrow{P_lP_k}$ and $\vec{v}_k$ are less than $\theta$
        then
            $\vec{v}_k \leftarrow \overrightarrow{P_jP_k} / \Delta t$.
            $v_N[i] \leftarrow \vec{v}_k$.
            remove $P_j$ from $P_{LC}$.
        else
            $\vec{v}_k \leftarrow \frac{1}{n}\vec{e}_k \cdot \sum_{j \in D} \left|\vec{v}_j\right|$
            $v_N[i] \leftarrow \vec{v}_k$
            find the nearest $P_j$ to $P_k$ in set $P_D$, and remove $P_j$ from $P_{LC}$.
        end if
    else
        $\vec{v}_k \leftarrow v_0 \vec{e}_k$
        $v_N[i] \leftarrow \vec{v}_k$
    end if
    $i \leftarrow i + 1$
end for
for each pedestrian $k$ in $P_{LC}$ do
    $P_kP_k + \vec{v}_k \cdot \Delta t$
    if $p_k$ is not in the exit
    then
        $P_N[i] \leftarrow P_k$
        $v_N[k] \leftarrow \vec{v}_k + \vec{a}_{SFM} \cdot \Delta t$
        $i \leftarrow i + 1$
    end if
end for
```

In Algorithm 1, the input values are the images of the current frame, as well as the coordinates (P_L) and speeds (v_L) of pedestrians in the previous frame; the output values are the coordinates (P_L) and speeds (v_L) of pedestrians in the current frame; each uppercase subscript represents a set of coordinates, while each lowercase subscript stands for a single coordinate.

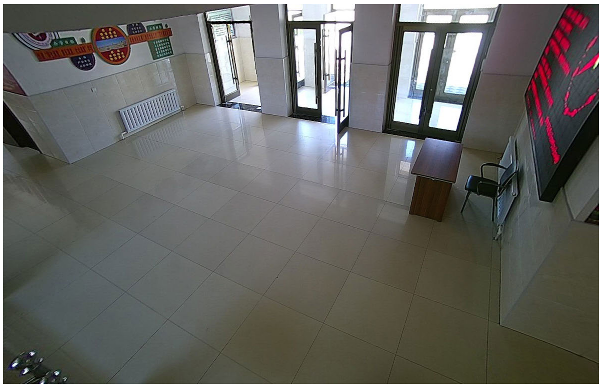


FIGURE 7. The scene of our dataset.

Firstly, the scene is initialized, and the speed and acceleration of each pedestrian are computed by the SFM. Then, P_{LC} is obtained as a copy of P_L. The coordinates P_S of pedestrian positions in the current frame are converted into the scene coordinates P_C.

For each pedestrian k in P_L, all the pedestrians within the distance of d(0.4m) are collected into a set P_D. Let v_k be the speed of pedestrian k, and e_k be the vector of the direction to the exit. If it is in the speed direction of a point P_j in the P_D, then the pedestrian k must be moving along the direction of the ray P_jP_k and his/her speed must be the ratio of P_jP_k to the interval Δt between two frames. Otherwise, the SFM simulation does not accurately reflect the real situation.

Next, the closest point to the pedestrian was selected as the position that the pedestrian appeared in the previous frame. The pedestrian speed was corrected against the mean speed of the crowd, and the speed direction was set towards the exit. If P_D is an empty set, then the pedestrian must be new to the scene. In this case, his/her speed was set to the normal walking speed, and the direction was set towards the exit.

For the pedestrians not fully captured by camera, their speeds and positions were computed by the SFM. The computed accelerations are denoted as a_{SFM}. Once a pedestrian reaches the exits, he/she was removed from the scene.

IV. EXPERIMENTAL VERIFICATION

Our method was compared with the original SFM in terms of trajectory and time through a five-stage experiment, which involves data preprocessing, pedestrian tracking, coordinate conversion, feet detection and SFM simulation.

A. DATA PREPROCESSING

The scene of our crowd evacuation dataset is the entrance hall (Figure 7) of the teaching building in a middle school of Jilin province, China. The camera was placed at the southwest corner of the hall.

The 6m-long, 4.8m-wide hall has three doors. The floor is covered by 0.6×0.6m ceramic tiles. The desk on the right

TABLE 1. The accuracy and time of the three models.

Model	D1 Accuracy	D1 Time (ms)	D2 Accuracy	D2 Time (ms)
Model A	0.934	74	0.750	107
Model B	0.912	91	0.818	121
Model C	0.917	108	0.831	148

side was regarded as the obstacle in evacuation. The hall is connected to the other parts of the building by a corridor on the left, a corridor on the right, and stairs at the bottom of the image. A total of 50 people volunteered to act as the evacuating crowd. They simulated the scenes in subway station by entering into the hall through the left corridor, the right corridor and the stairs in order. The simulation lasts 140min, including 120min of free access and 20min of evacuation. The current study focuses on the middle door, where is located at 1280×720 25fps in the video with intervals of 0.04s.

B. PEDESTRIAN TRACKING

Three popular object detection models were compared, including SSD ResNet50 FPN (Model A), Faster R-CNN-ResNet 50 (Model B) and Faster R-CNN ResNet 101 (Model C). All models were pretrained on COCO dataset. The experiment was conducted on Ubuntu 16.04, using Scikit-learn 19.1 and Redis 4.08.

A total of 500 images were randomly selected and grouped into dataset D1 (with fewer than 15 pedestrians in the scene), and another 500 random images were collected into dataset D2 (with more than 15 pedestrians in the scene). The accuracy and time of the three models are compared in Table 1 below. Note that the accuracy refers to the ratio of the number of pedestrians identified by each model to the number of pedestrians marked manually, and the time stands for the processing time of each model. Fewer than half of the manually marked pedestrians are blocked by other pedestrians.

As shown in Table 1, Model A was faster than the other two models over D1 but less accurate than the latter over D2; Model C was more accurate yet slower than Model B. Therefore, Model A was selected for sparse crowds (with fewer than 15 pedestrians) and model C was selected for dense crowds (with more than 15 pedestrians).

Then, the pedestrian images and environment images were converted into the size 240×400, and the empty parts were filled with white color. Then, Model A and Model C were applied for transfer learning. Through the learning, Model C reached the accuracy of 0.961 on D1 and 0.880 on D2. In this way, the position of each pedestrian was obtained, facilitating the drawing of his/her frame.

C. COORDINATE CONVERSION

As shown in Figure 8, a rectangular plane coordinate system P (Figure 9) was set up with the blue point as the origin, the intersecting line between the doors and the floor as the y-axis, the intersecting line between the side wall and floor as the x-axis.

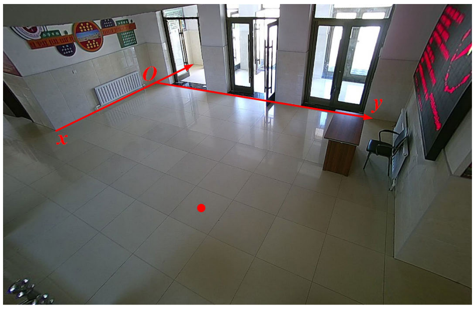


FIGURE 8. The coordinate system in the image.

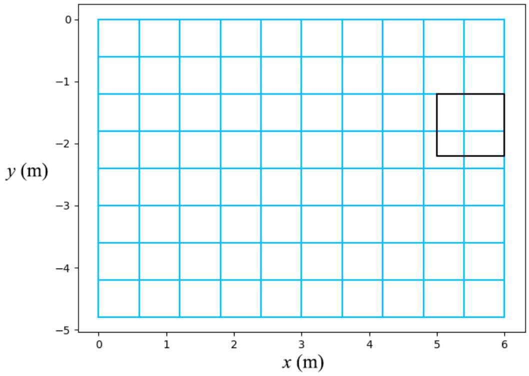


FIGURE 9. The rectangular plane coordinate system.

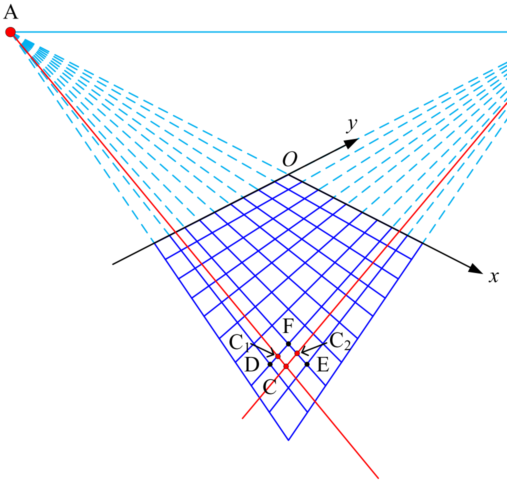


FIGURE 10. The intersecting lines.

In Figure 9, each point corresponds to a pixel in Figure 8, and each blue square is of the same size of the ceramic tile (0.6×0.6m). Next is to convert the coordinates of each pixel in Figure 8 to the coordinates in the coordinate system P. The conversion process is explained as follows.

Firstly, the parallel lines in Figure 9 were mapped into intersecting lines, similar to how straight rails intersect each other in the distance. Two intersection points, A and B, were obtained. As shown in Figure 10, the coordinates of the two points are (x_A, y_A) and $x_B, y_B)$, respectively, and can be determined by computing the intersection between any two lines. Then, the origin O in P and the x- and y-directions in the original image can be derived from these coordinates.

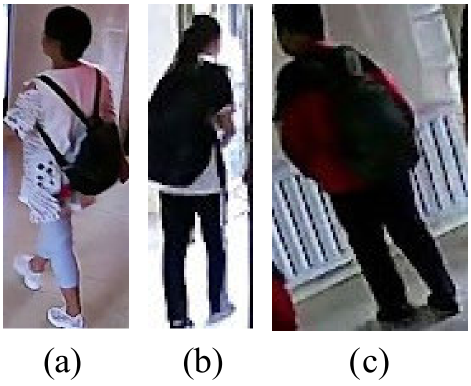


FIGURE 11. Several pedestrians in the scene.

In Figure 10, C is the position of the object; AC and BC are two rays passing through point C; C_1 is the intersection point between AC and BD; C_2 is the intersection point between BC and AF. The coordinates of C, D, E and F can be measured directly from the image. Then, $y = a_{BD}x + b_{BD}$ can be derived from the coordinates of D and B, $y = a_{AF}x + b_{AF}$ from those of A and F, $y = a_{AC}x + b_{AC}$ from those of A and C, and $y = a_{BC}x + b_{BC}$ from those of B and C. Thus, the coordinates of C_1 and C_2 can be respectively computed by:

$$\begin{cases} y = a_{BD}x + b_{BD} \\ y = a_{AC}x + b_{AC} \end{cases} \tag{13}$$

$$\begin{cases} y = a_{AF}x + b_{AF} \\ y = a_{BC}x + b_{BC} \end{cases} \tag{14}$$

The coordinates of C in P can be determined by:

$$\begin{aligned} x_C &= x_E + 0.6 \cdot d_{EC2}/d_{EF} \\ y_C &= y_E - 0.6 \cdot d_{EC1}/d_{ED} \end{aligned} \tag{15}$$

where d_{EC2}, d_{EC1}, d_{ED} and d_{EF} are the distances between E and C_2, C_1, D and F, respectively, in the original image; X_E and Y_E are the coordinates of point E, the corner of ceramic tile. Here, the values of X_E and Y_E are directly measured as 4.2 and −3.6. Thus, the coordinates of C can be determined as 4.5 and −3.9.

D. FEET DETECTION

With model B and Faster R-CNN, the boundaries of the crowd were obtained very rapidly. It is impossible to pinpoint a pixel with a single image alone. Therefore, the feet position of each pedestrian was used to determine his/her position in the scene.

As shown in Figure 11, the center of gravity of the pedestrians was not always the midpoint of the bottom boundary. Errors brought by the projection is expected to be controlled within 0.2m. Assuming that the coordinates of that midpoint are (60, 500) in P, the actual coordinates of the center of gravity and their errors from the midpoint can be summed up in Table 2 below.

To reduce the errors, the relative position (left, middle or right in the image) of the feet was evaluated by deep learning models, including ResNet-50, ResNet-101 and VGG-16. The pedestrian images were taken as training sets, and the feet coordinates as labels. Then, the 5,000 labelled images were divided into a training set (80%) and a test set (20%). Our dataset was not expanded, to prevent the shift in the center of gravity. The accuracies of the three models are compared in Table 3.

TABLE 2. The actual coordinates of the center of gravity and their errors from the midpoint.

Coordinates of	Error (m)
(10, 500)	0.41
(20, 500)	0.42
(30, 500)	0.32
(40, 500)	0.21
(50, 500)	0.105
(70, 500)	0.105
(80, 500)	0.21
(90, 500)	0.31
(100, 500)	0.42

TABLE 3. The accuracies of the three models.

	ResNet-50	ResNet-101	VGG-16
Accuracy	0.44	0.61	0.91

As shown in Table 3, VGG-16 achieved the highest accuracy in the small dataset. The poor performance of ResNet-50 and ResNet-101 is attributable to their emphasis on texture classification. The feet position is measured as r through VGG, and the horizontal coordinate of the pedestrian's position can be calculated by Equation 16 in the picture.

$$x = a \cdot l_x + x_{\min} \tag{16}$$

In Equation 16 lx is the length in x-axis of the pedestrian identification's border in chapter 4.1, $x_{\min}$ is the x-coordinate of left border. When the feet position is on the left, a is 0.165, when the position is in the middle, a is 0.5, and a takes 0.825 when the pedestrian foot position is on the right. The y-coordinate of the pedestrian's position is the border y-coordinate under the pedestrian identification's border in the picture. The error between calculated coordinates and the real coordinates of the maximum is less than 0.15m.

E. SFM SIMULATION

The SFM simulation parameters were configured as follows: the weight of each pedestrian, 50kg (the mean weight of all volunteers); the radius of each pedestrian, 0.4m; extrusion between pedestrians, 2,000N; extrusion between pedestrian and wall, 2,000N; B_i 0.04m; B_w, 0.08m. (B_i and B_w are the constants of the social force model, which has been mentioned in Section 3.C)

Firstly, a single pedestrian in the scene was simulated by the SFM. The results on a pedestrian coming out of the left aisle are presented in Figure 12 for example, where the blue line is the trajectory captured by the camera (the actual trajectory) and the red one is the simulated trajectory.

As shown in Figure 12, the wall had a limited impact on the single pedestrian. During the walking process, the pedestrian

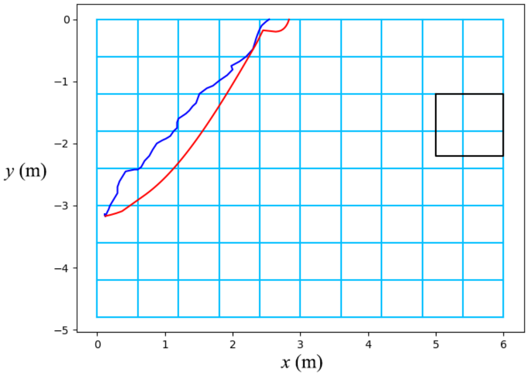


FIGURE 12. The simulated and actual trajectories of a single pedestrian.

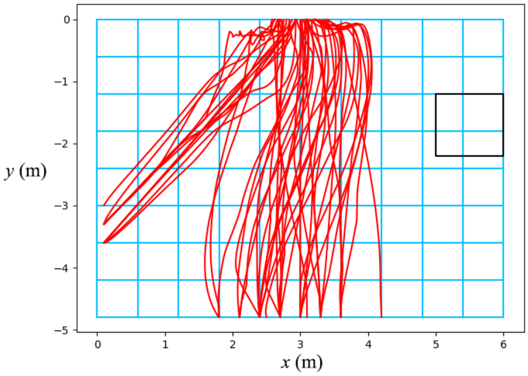


FIGURE 13. The trajectories simulated by our method.

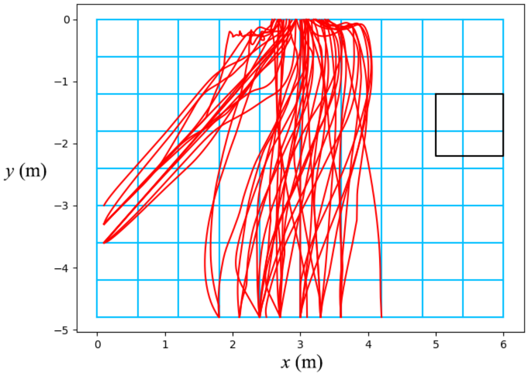


FIGURE 14. The trajectories simulated by the SFM.

trajectory was almost straight. The blue trajectory took 3.44s and the red one took 3.64s, showing a very small difference.

Next, the trajectories of multiple pedestrians were simulated separately by the SFM and our method, which couples the SFM with deep learning. The results of the SFM are displayed in Figure 14, and those of our method in Figure 13.

Comparing Figure 13 and Figure 14, it is easy to learn that the walls exerted little impact on pedestrians. This is because the volunteers are familiar with the environment. The results of our method were closer to straight lines than those of the SFM. This phenomenon can be explained as follows: The SFM can neither predict the mentality of the volunteers, nor capture the pedestrian positions in all frames. In addition, the initial speed inputted into the SFM is not the expected speed.

TABLE 4. Comparison of evacuation time and pedestrian speed.

	Actual results	Our method	SFM
Evacuation Time	23s	25s	29s
Mean speed	0.94m/s	0.90m/s	0.84m/s

As shown in Table 4, both our method and the SFM produced longer evacuation time and slower mean speed than the actual results, but the results of our method were closer to the actual data.

V. CONCLUSION

In this paper, we proposed a crowd model combining deep learning and social force model simulation. For sparse and dense crowd, we selected the optimal pedestrian detection algorithm respectively and did the migration learning according to the dataset of this paper. Then deep learning classification algorithm is used to evaluate the drop point where the pedestrian's gravity center and acquire the pedestrian's trajectory. Pedestrian detection and trajectory calculation are used to make up for the deficiency of social force models in human body differences and random events, and social force models are used to make up for the problem of personnel occlusion and the lack of accuracy in crowded people. Through our experiment, compared with the traditional social force model, a more accurate crowd evacuation process, pedestrians' trajectory and evacuation time can be acquired through simulation algorithm. The results also suggest that the more familiar pedestrians are with the scene, the less influence the wall will have on them. The accuracy of the Faster R-CNN selected for the current experiment is low in dense crowds, and there is only one exit of the data. In the future study, the pedestrian detection accuracy will be further improved by employing other algorithms to discuss evacuation path selection, and explore the impact of multiple exits for evacuation.

REFERENCES

[1] T. Weiss, A. Litteneker, C. Jiang, and D. Terzopoulos, "Position-based multi-agent dynamics for real-time crowd simulation," in *Proc. ACM SIGGRAPH / Eurograph. Symp. Comput. Animation*, Los Angeles, CA, USA, Jul. 2017, Art. no. 27.

[2] Q. Wang, H. Liu, K. Gao, and L. Zhang, "Improved multi-agent reinforcement learning for path planning-based crowd simulation," *IEEE Access*, vol. 7, pp. 73841–73855, 2019.

[3] Z. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, and H. He, "Robot-assisted pedestrian regulation based on deep reinforcement learning," *IEEE Trans. Cybern.*, to be published. doi: 10.1109/TCYB.2018.2878977.

[4] M. Zsifkovits and T. S. Pham, "Modelling and parameterizing pedestrian behaviour in public places: A review," *Int. J. Simul. Model.*, vol. 16, no. 4, pp. 630–643, 2017.

[5] Y. Niu, Y. Zhang, J. Zhang, and J. Xiao, "Running cells with decision-making mechanism: Intelligence decision P system for evacuation simulation," *Int. J. Comput., Commun. Control*, vol. 13, no. 5, pp. 865–880, 2018.

[6] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, pp. 4282–4286, 1995.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[8] X. Huang, T. Zhang, Z. Deng, and Z. Li, ''Design of moving target detection and tracking system based on cortex-A7 and openCV,'' *Traitement Signal*, vol. 35, no. 1, pp. 61–73, 2018.

[9] Y. Jiang, M. Zhao, C. Hu, L. He, H. Bai, and J. Wang, ''A parallel FP-growth algorithm on world ocean atlas data with multi-core CPU,'' *J. Supercomput.*, vol. 72, no. 2, pp. 732–745, 2019.

[10] J. Fruin, *Pedestrian Planning and Design*. New York, NY, USA: Metropolitan Association of Urban Designers and Environmental Planners, 1971.

[11] T. Kretz, ''Pedestrian traffic: On the quickest path,'' *J. Stat. Mech., Theory Exp.*, vol. 3, no. 12, 2009, Art. no. P03012.

[12] D. Hartmann, ''Adaptive pedestrian dynamics based on geodesics,'' *New J. Phys.*, vol. 12, no. 4, 2010, Art. no. 043032.

[13] R.-Y. Guo and H.-J. Huang, ''Route choice in pedestrian evacuation: Formulated using a potential field,'' *J. Stat. Mech., Theory Exp.*, vol. 4, Apr. 2011, Art. no. P04018.

[14] M. Haghani and M. Sarvi, ''Stated and revealed exit choices of pedestrian crowd evacuees,'' *Transp. Res. B, Methodol.*, vol. 95, pp. 238–259, Jan. 2017.

[15] Y. Jiang, T. Zhang, Y. Gou, L. He, H. Bai, and C. Hu, ''High-resolution temperature and salinity model analysis using support vector regression,'' *J. Ambient Intell. Hum. Comput.*, to be published. doi: 10.1007/s12652-018-0896-y.

[16] D. Nilsson and A. Johansson, ''Social influence during the initial phase of a fire evacuation—Analysis of evacuation experiments in a cinema theatre,'' *Fire Saf. J.*, vol. 44, no. 1, pp. 71–79, 2009.

[17] S. Heliövaara, J.-M. Kuusinen, T. Rinne, T. Korhonen, and H. Ehtamo, ''Pedestrian behavior and exit selection in evacuation of a corridor—An experimental study,'' *Saf. Sci.*, vol. 50, no. 2, pp. 221–227, 2012.

[18] R. Lovreglio, A. Fonzone, L. Dell'Olio, and D. Borri, ''A study of herding behaviour in exit choice during emergencies based on random utility theory,'' *Saf. Sci.*, vol. 82, pp. 421–431, Feb. 2016.

[19] P. H. Bovy and E. Stem, ''Wayfinding: Choice and search,'' in *Route Choice: Wayfinding in Transport Networks*. Dordrecht, The Netherlands: Springer, 1990, ch. 1, pp. 1–14.

[20] R. L. Hughes, ''A continuum theory for the flow of pedestrians,'' *Transp. Res. B, Methodol.*, vol. 36, no. 6, pp. 507–535, 2002.

[21] Y.-Q. Jiang, S.-G. Zhou, and F.-B. Tian, ''A higher-order macroscopic model for bi-direction pedestrian flow,'' *Phys. A, Stat. Mech. Appl.*, vol. 425, pp. 69–78, May 2015.

[22] J. A. Carrillo, S. Martin, and M.-T. Wolfram, ''An improved version of the Hughes model for pedestrian flow,'' *Math. Models Methods Appl. Sci.*, vol. 26, no. 4, pp. 671–697, 2016.

[23] D. Helbing, I. Farkas, and T. Vicsek, ''Simulating dynamical features of escape panic,'' *Nature*, vol. 407, no. 6803, pp. 487–490, 2000.

[24] H. Qin, H. Chen, and Y. Sun, ''Distributed finite-time fault-tolerant containment control for multiple ocean bottom flying nodes,'' *J. Franklin Inst.*, to be published. doi: 10.1016/j.jfranklin.2019.05.034.

[25] F. Johansson, A. Peterson, and A. Tapani, ''Waiting pedestrians in the social force model,'' *Phys. A, Stat. Mech. Appl.*, vol. 419, pp. 95–107, Feb. 2015.

[26] W. Li, J. Gong, P. Yu, S. Shen, R. Li, and Q. Duan, ''Simulation and analysis of congestion risk during escalator transfers using a modified social force model,'' *Phys. A, Stat. Mech. Appl.*, vol. 420, no. 6, pp. 28–40, 2015.

[27] B. Liu, H. Liu, H. Zhang, and X. Qin, ''A social force evacuation model driven by video data,'' *Simul. Model. Pract. Theory*, vol. 84, pp. 190–203, May 2018.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ''Rich feature hierarchies for accurate object detection and semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[29] S. Ren, K. He, R. Girshick, and J. Sun, ''Faster R-CNN: Towards real-time object detection with region proposal networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[30] H. Qin, C. Wang, Y. Jiang, Z. Deng, and W. Zhang, ''Trend prediction of the 3D thermocline's lateral boundary based on the SVR method,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2018, p. 252, Dec. 2018.

[31] C. Szegedy, A. Toshev, and D. Erhan, ''Deep neural networks for object detection,'' in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 2553–2561.

[32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, ''OverFeat: Integrated recognition, localization and detection using convolutional networks,'' in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–16. [Online]. Available: https://arxiv.org/abs/1312.6229

[33] K. He, X. Zhang, S. Ren, and J. Sun, ''Spatial pyramid pooling in deep convolutional networks for visual recognition,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[34] R. Girshick, ''Fast R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[35] H. Qin, H. Chen, Y. Sun, and L. Chen, ''Distributed finite-time fault-tolerant containment control for multiple ocean bottom flying node systems with error constraints,'' *Ocean Eng.*, vol. 189, Oct. 2019, Art. no. 106341. doi: 10.1016/j.oceaneng.2019.106341.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ''SSD: Single shot MultiBox detector,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 9905, 2015, pp. 21–37.

[37] 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. (CVPR)*, Jun. 2016, pp. 779–788.

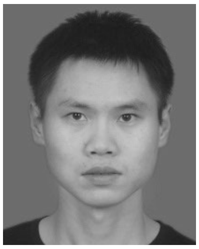
[38] H. Law and J. Deng, ''CornerNet: Detecting objects as paired keypoints,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.

[39] R. Neelapu, G. L. Devi, and K. S. Rao, ''Deep learning based conventional neural network architecture for medical image classification,'' *Traitement Signal*, vol. 35, no. 2, pp. 169–182, 2018.

[40] J. J. Yang, Y. L. Yuan, X. Zhang, L. F. Shao, L. H. Gong, J. Mi, and T. Xu, ''A deep learning-based image recognition algorithm for fecal shape of domestic rabbits,'' *Rev. d'Intell. Artif.*, vol. 32, no. S1, pp. 67–78, 2018.

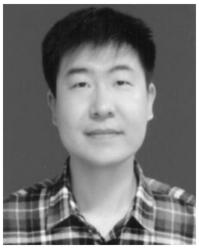
XIN LI received the master's degree in software engineering from the College of Software, Jilin University, Changchun, China, in 2010. He is currently pursuing the Ph.D. degree in artificial intelligence and machine learning with the College of Computer Science and Technology. His research interests include information extraction and machine learning.

YANCHUN LIANG received the Ph.D. degree in applied mathematics from Jilin University, Changchun, China, in 1997. He was a Visiting Scholar with The University of Manchester, U.K., from 1990 to 1991, a Visiting Professor with the National University of Singapore, from 2000 to 2001, a Guest Professor with the Institute of High Performance Computing, Singapore, from 2002 to 2004, a Guest Professor with Trento University, Italy, from 2006 to 2010, and a Guest Professor with the University of Missouri, USA, from 2011 to 2017. He is currently a Professor with the College of Computer Science and Technology, Jilin University. He has published more than 400 articles. His research interests include computational intelligence, machine learning methods, text mining, MEMS modeling, and bioinformatics. He was a recipient of several grants from NSFC and EU. His research was featured in Bioinformatics, the IEEE TSMC, the IEEE TKDE, the *Journal of Micromechanics and Microengineering*, *Physical Review E*, *Neural Computing and Applications*, *Smart Materials and Structures*, *Artificial Intelligence in Medicine*, and *Applied Artificial Intelligence*.



MINGHAO ZHAO received the B.S. degree in computer science and technology from Jilin University, China, in 2017, where he is currently pursuing the master's degree with the College of Computer Science and Technology. His current research interests include the Internet of Things, industrial automation, machine learning, and machine vision.

CHONG WANG received the Ph.D. degree in design and construction of naval architecture and ocean structure from Harbin Engineering University, Heilongjiang, China, in 2017. Since, he has been a Postdoctoral Fellow with Jilin University, Jilin, China. His research interests include marine data analysis, deep learning, autonomous underwater vehicle, and energy consumption management.



HONGTAO BAI received the B.S., M.E., and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 1998, 2003, and 2010, respectively. He is currently the Vice Professor with the Center for Computer Fundamental Education, College of Computer Science and Technology, Jilin University. His research interests include high performance computing and computer vision.

YU JIANG received the M.E. and Ph.D. degrees from the College of Computer Science and Technology, Jilin University, China, in 2005 and 2011, respectively. He is currently the Vice Professor with the College of Computer Science and Technology, Jilin University. His research interests include wireless sensor networks and data mining.