2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Intelligent Disinfection Robot with High-Touch Surface Detection and Dynamic Pedestrian Avoidance

Yunfei Luan†, Muhang He†, Yudong Tian, Chengjie Lin, Yunhan Fang, Zihao Zhao, Jianxin Yang, Yao Guo

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610836

***Abstract*— The increasing awareness of public health issues has highlighted the need for effective disinfection of crowded indoor public areas, leading to the development of automated disinfection robots. However, most of the existing robots spray disinfectant in all areas, and they are still immature to navigate in densely populated environments. Hence, in this paper, we design a new disinfection robotic system consisting of a mobile platform, an RGB-D camera, and a robotic arm with a spray disinfection device. To address the above challenges, we propose a vision-based method for accurately detecting high-touch areas in the surroundings, enabling the disinfection robot to achieve superior disinfection efficiency. In addition, we propose a dynamic pedestrian avoidance method, namely Socially Aware APF (SA-APF), which can predict the movement trend of pedestrians and plan the path in real-time. Both simulated and real-world experiments are conducted to demonstrate the effectiveness of our disinfection robot system, especially highlighting the ability to detect high-touch areas and navigate in the environment while avoiding dynamic pedestrians.**

I. Introduction

The emergence of the COVID-19 pandemic coupled with an escalation in public health conscientiousness, has underscored the urgent necessity of proficiently disinfecting densely populated environments to mitigate the onset of infectious diseases [1]. Public interiors, notably hospitals, hospitality establishments, and civic hubs, are particularly vulnerable to the dispersal of viral contagions, compounded by the confluence of heavy pedestrian traffic and inherent ventilation impediments. Although conventional manual disinfection methods are prevalent, their protracted time requirements and limited efficacy persist as notable drawbacks [2]. Nonetheless, such exposure and potential transmission may imperil the health and safety of the well-being of the sanitation workforce.

In the wake of technological advancements, a slew of disinfection robots have been engendered and deployed [3]–[7], showing remarkable efficacy under their elevated levels of robotic automation. Nonetheless, it merits attention that these robots indiscriminately discharge antiseptics or directly activate ultraviolet light, which is profligate and inefficient.

This work was supported by the Science and Technology Commission of Shanghai Municipality under Grant 20DZ2220400. †These authors contributed equally to this work.

Y. Luan, M. He, Y. Tian, C. Lin, and Y. Fang, J. Yang, and Y. Guo are with the School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, China. Z. Zhao is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.
Corresponding authors: *Jianxin Yang, Yao Guo*.

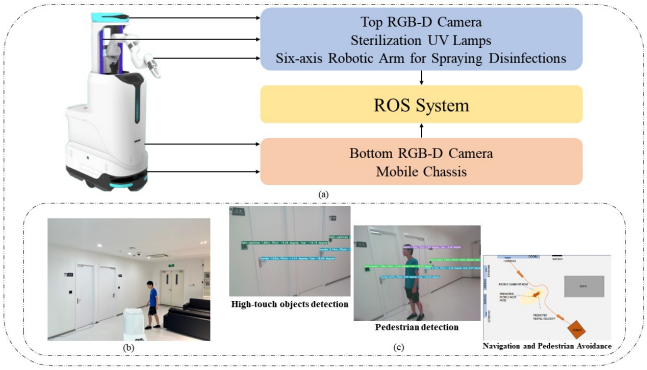

Fig. 1. Disinfection robot system overview. (a) the overall composition of the system, (b) the robot in a real scenario, (c) an illustration of the function modules.

Moreover, these robots often perceive pedestrians as immobile hindrances, neglecting the intricate dynamics of human mobility.

To address the aforementioned challenges, we design an autonomous disinfection robot that can navigate in complex indoor environments and especially focus on disinfecting high-touch areas. As shown in Fig. 1, our system is built on a mobile platform equipped with 2D LiDAR sensors and RGB-D cameras. The robot comprises four functional modules: visual recognition, navigation, robot control, and sterilization. These modules interact with each other via the Robot Operation System (ROS).

Specifically, the vision recognition module detects high-touch objects such as doorknobs and switches from an embedded RGB-D camera. By fusing the depth information, the relative positions and orientations between the robot and the target can be derived. Furthermore, the vision recognition module exhibits the capacity to identify pedestrians in the robot's vicinity, thereby providing spatial information about dynamic pedestrians with respect to the robot's position [8].

It is paramount to guarantee the real-time execution of detection algorithms. The Nvidia Jetson platform stands out prominently owing to its capability to expedite deep-learning algorithms while maintaining a low power budget and high efficiency with TensorRT [9]. Its compact form factor, which allows seamless integration within a robot, is an added boon. YOLO [10], a distinguished one-stage object detection framework noted for its harmony of speed and precision, which is supported by NVIDIA's TensorRT SDK, can significantly enhance detection efficiency thus making

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.

it an apt choice for deployment on this edge computing platform. Subsequently, we employ the YOLO as the backbone network for object detection, which exhibits excellent deployability on the embedded platform, enabling real-time multi-object detection while upholding satisfactory accuracy.

The navigation module guarantees the robot reaches its destination quickly and safely. Rapidly-Exploring Random Trees (RRT) [11] and its variants [12] [13] are a kind of global path planner that effectively avoids static obstacles and tries to find the shortest path. However, its disadvantage lies in being computationally complex and unable to achieve real-time performance. Therefore we use it for pre-travel path planning to ensure a high-quality global path. Since the robot is likely to encounter temporary obstacles when traveling that do not exist on the map, a local path planner with low compute complexity is needed for real-time path correction. Although the Artificial Potential Field (APF) [14] algorithm meets our needs, it is designed for static obstacles, not considering the social attributes of robots. Without sufficient prediction of pedestrian mobility, the APF local planner will lead to unnecessary deceleration, stagnation, or collision with pedestrians. Thus, we propose a novel local planner based on the APF algorithm, namely Socially Aware APF (SA-APF).

In summary, the main contribution of this paper is threefold:

- We design an automated disinfection robot system, which has high disinfection efficiency and robust navigation skills and is thus more suitable for crowded scenarios with high disinfection needs such as hospitals.
- A dedicated deep model for high-touch area detection is introduced to detect the target objects with high disinfection needs (such as door handles and switches).
- Deep learning-based pedestrian detection and improved dynamic pedestrian avoidance are combined to predict the movement trend of pedestrians, to achieve real-time path planning and dynamic obstacle avoidance.

II. System Overview

A. Hardware System

The disinfection robot is built upon a mobile platform consisting of differential drive wheels, with an overall volume of $850mm \times 400mm \times 1130mm$. It comprises a 6DOF robotic arm, furnished with a dual-liquid disinfection setup, alongside four UV disinfection lamps, RGB-D cameras, and 2D LiDAR detectors. The rear of the robot features wireless charging modules and the front of the robot has a touchscreen.

B. Functional Modules

Our disinfection robot mainly consists of four functional modules. (1) *Robot Control*: the various modules of the robot can work together through the ROS system. An industrial computer located at the bottom of the robot acts as the host, which is responsible for controlling the velocity of the robot chassis. (2) *Vision Recognition*: this module demonstrates swift identification of high-touch objects and pedestrians, furnishing the essential information for other modules; performs real-time identification and information-return tasks by an RGB-D camera and a combination of algorithms; and efficiently provides distance and angle information for other modules to collaborate with, by deploying on the edge computer and establishing ROS communication with the host computer. (3) *Navigation and Obstacle Avoidance*: this module can perform path planning tasks under densely populated environments; build the map of disinfection space and locate the robot itself by LiDAR sensors and SLAM algorithm; plan a global path before traveling, avoiding collisions with static obstacles and trying to find the shortest path by RRT* algorithm; and flexibly correct its path in real-time to prevent collisions with pedestrians by SA-APF algorithm. This module deploys algorithms on the host computer. It receives pedestrian information from the vision recognition module and publishes velocity information to the robot control module. (4) *Disinfection*: the robot has two sets of disinfection functions. The first is to use the irradiation of ultraviolet lamps, which has the advantages of high efficiency and wide disinfection range. The second is to use the disinfectant spray, which has the advantage of flexibility. This module works after the robot successfully navigates to the high-touch area.

III. Vision Recognition

To accomplish effective path planning and precise disinfection, it is critical for the vision recognition module to discern the high-touch objects and relay distance data to the robot. In pursuit of this objective, we select YOLOv8s, which has an equilibrium between speed and precision, effectively supported by the Nvidia TensorRT SDK. Subsequently, we implement an attention module along with a lightweight convolution module to augment performance further. Additionally, depth information procured from an RGB-D camera confers three-dimensional spatial hints of the identified objects. Coordinated with navigation and disinfection modules, the robot is proficient in executing pedestrian avoidance and disinfection tasks with efficacy.

A. Overview of YOLOv8s

The YOLO framework has achieved remarkable success in the field of computer vision. YOLOv8s [15] is a progressive, state-of-art model supported by Ultralytics. YOLOv8s encompasses five basic models that diverge in terms of model depth and parameters, yet they all adhere to the same fundamental framework. As model size escalates, so does its discernment precision, although this may incur a longer inference duration on the Nvidia Xavier NX edge computing platform due to an overabundance of parameters. Our examination of all basic models reveals that although YOLOv8s falls slightly short of its counterparts in terms of performance accuracy, it demonstrates superior inference speed after training. As a result, we have chosen this model for its commendable balance in recognition performance.

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.

B. Object Detection Framework

To attain rapid computation and lightweight deployment while improving accuracy, we incorporate the Convolutional Block Attention Module (CBAM) [16] and C3Ghost Module into the backbone and the head of the YOLOv8s. The C3Ghost module comprises GhostNet [17] as its fundamental building block. GhostNet utilizes a limited number of convolutional kernels to extract features from input. Thereafter, these features are processed by linear layers, rather than complex convolutional layers. The final feature maps are generated through concatenation. The diagram of the C3Ghost module is shown in Fig. 2. CBAM is a lightweight convolutional attention module that includes two sub-modules, the Channel Attention Module (CAM) and the Spatial Attention Module (SAM), which conduct channel and spatial attention computation respectively.

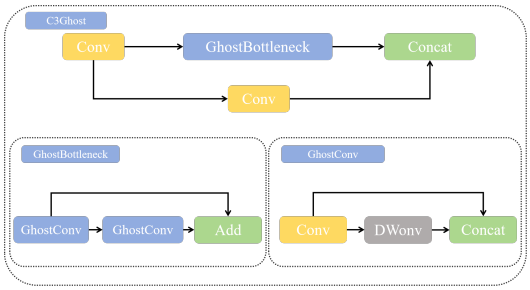

Fig. 2. The architecture of the C3Ghost module.(1) GhostConv obtains the feature map by concatenating the Conv layer output and the output after Conv and DWConv [18];(2) GhostBottleneck is formed by using two GhostConvs in series, using jump connection pathways;(3) C3Ghost module output consists of concatenating GhostBottlenek and Conv layer outputs

The CAM entails performing average pooling and maximum pooling operations on the input features. Subsequently, the average-pooled and maximum-pooled features are combined in a MultiLayer Perceptron (MLP) architecture to yield the final channel attention feature mapThe computation of the channel attention feature map is given by the equation:

$$M_C(F) = \sigma(MLP(\text{AvgPool}(F) + MLP(\text{Max}\,Pool(F))) = \sigma\left(W_1\left(W_0\left(F^c_{\text{avg}}\right)\right) + W_1\left(W_0\left(F^c_{\max}\right)\right)\right) \quad (1)$$

where $M_c(F)$ is channel attention features, and F^c_{avg}, F^c_{max} represent the average pooling and max pooling features.

To calculate the spatial attention, the module applies average pooling and maximum pooling operations along the channel dimensions. Subsequently, the resulting feature maps are utilized to perform convolution on the concatenated feature maps, resulting in the generation of the ultimate spatial attention feature maps. The computation of the spatial attention feature map is expressed as follow:

$$M_s(F) = = \sigma\left(f^{7\times7}\left(F^s_{\text{avg}}; F^s_{\max}\right)\right) \quad (2)$$

where $M_s(F)$ stands for channel attention features and $f^{7\times7}$ is a convolutional kernel of size 7x7.

The CBAM module operates on the feature map by assigning an attention weight to each channel and spatial location. Fig. 3 is the CBAM structure diagram and Fig. 4 shows the improved YOLOv8s framework diagram.

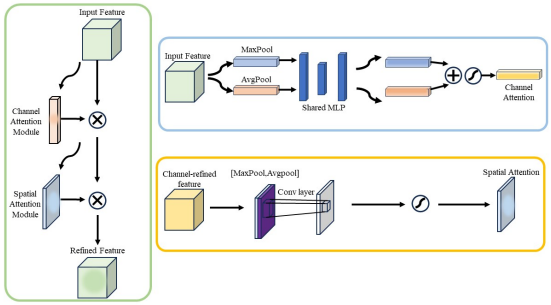

Fig. 3. The structure of the CBAM. It contains two main components, channel attention and spatial attention. Both of the two sub-modules use average-pooling and max-pooling outputs with a shared network [16].

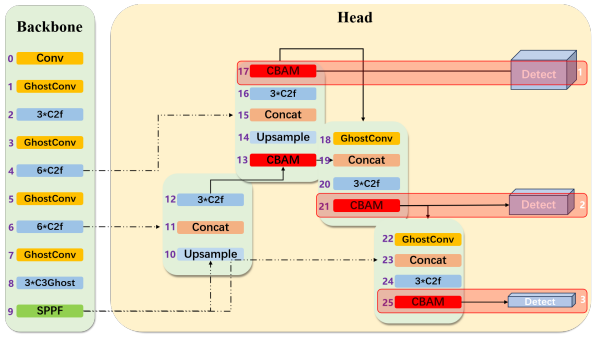

Fig. 4. Structure of the improved YOLOv8s method. Compared with the original YOLOv8s method, the method has two structural improvements. (1) The G3Ghost module and the attention module are introduced into the backbone layer; (2) the attention module is added to the feature fusion layer.

IV. Navigation and Pedestrian Avoidance

To perform disinfection tasks in a wide indoor area, the robot needs to build maps of disinfection scenarios, and then with the given disinfection sites on a map, the robot gets to them one by one. Particularly, when faced with human beings who are standing or walking, the robot should escape them flexibly in a social manner, instead of just stopping and waiting for pedestrians to pass. To tackle this problem, a two-wheel differential mobile platform, several LiDAR sensors, and RGBD cameras are used together. Additionally, we integrate several existing algorithms and propose SA-APF as the local path planner based on traditional APF.

A. Mapping and Repositioning

Two 2D LiDAR sensors are fixed in front and to the side of the robot. Based on the SLAM algorithm, the 2D map of the indoor scenarios can be acquired. Every time the robot moves in the same scenario, its LiDAR sensors and inertial measurement unit can work together to re-localize the robot on the existing map.

B. Global Planner: RRT*

Before the robot starts to move from one disinfection site to another, a global path plan is made in advance. The global planner helps to shorten planning time cost and improve path quality during the robot's actual moving.

We choose RRT* [12] as the global planner. The RRT* algorithm is an extension of the RRT [11], which can

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.

generate a collision-free path from the start to the goal. The RRT algorithm works in the following processes. (1) Randomly sample a node X_{rand}. (2) Find the nearest node from X_{rand} in the existing node list as X_{near}. (3) Connect X_{rand} and X_{near} as the direction of tree growth. X_{new} is decided based on a fixed step length. If the line segment between X_{new} and X_{near} is collision-free, we consider this steer as successful. (4) For the nodes within a specified radius of X_{new}, if the cost between the node and X_{new} is lower than that between X_{new} and its parent node and collision-free, rewrite the node as the new parent node thus the parent node is updated. (5) Iterate the process above until the distance between X_{new} and the goal is less than the fixed step length and free of collision.

Finally, we get a tree consisting of a series of nodes, which forms the path we need. Compared to RRT, RRT* adds a rewiring operation every time after finding X_{new}. For the nodes within a specified radius of X_{new}, it updates the parent node if the new path is better than the original path. Hence, RRT* can generate a shorter path than RRT in the case of ensuring safety.

C. Local Planner: SA-APF

To deal with the dynamic obstacles that may appear in the real scene, we propose a novel local planner based on the APF [14] algorithm, namely Socially Aware APF(SA-APF). APF abstracts a scene to an artificial potential field, where the obstacles generate a repulsive field, and the targets generate an attractive field. In addition, the robot's velocity is controlled by the force it receives in the potential field.

Here is a brief description of traditional APF. The attractive force is defined by :

$$F_{att}(q) = -\eta\rho\left(q, q_g\right)\mathbf{r}(q, q_g) \tag{3}$$

where q, q_g represent the positions of the robot and the goal. η is the proportional gain coefficient of attractive force. $\rho\left(q, q_g\right)$ represents the Euclidean distance between the robot's position q and the goal point's position q_g.$\mathbf{r}(q, q_g)$ is the unit vector from goal to robot.

Similarly, the repulsive force is given by:

$$F_{rep}(q) = \begin{cases} k\left(\frac{1}{\rho(q,q_0)} - \frac{1}{\rho_0}\right)\frac{\mathbf{r}(q,q_0)}{\rho^2(q,q_0)} & 0 \le \rho < \rho_0 \\ 0 & \rho \ge \rho_0 \end{cases} \tag{4}$$

where q, q_0 represent the positions of the robot and the obstacle. k is the proportional gain coefficient of repulsion. $\rho\left(q, q_0\right)$ is the Euclidean distance between the robot's position q and the obstacle's position q_0. ρ_0 is the minimum safety distance between the robot and the obstacle, which is the sum of the obstacle radius and robot radius. $\mathbf{r}(q, q_0)$ is the unit vector from obstacle to the robot.

The combined force is the superposition of the attractive force and the repulsive force. For attractive and repulsive forces to work well together, the ratio of proportional gain coefficients η and k should be carefully adjusted to maintain balance. Based on the basic obstacle avoidance APF algorithm, we further consider pedestrians as a special kind of mobile obstacle. Thus, we design the SA-APF algorithm by taking obstacles' mobility into account and hence is more suitable for social conditions between humans and robots.

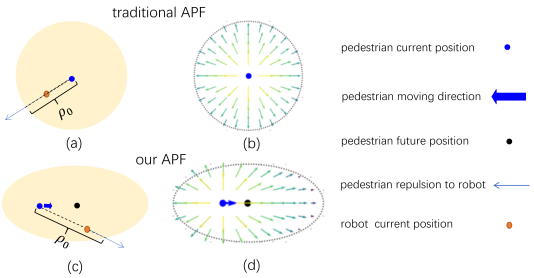

Fig. 5. Repulsion distribution comparison between traditional APF and our SA-APF. We redefine the central point of the potential field and our potential field shape is elliptical while that of traditional APF is circular.

Firstly, we predict the velocity state of a pedestrian in order to get mobility information. An RGBD camera fixed on the front of the robot detects pedestrians and locates them in the camera coordinate. So their pose T^{human} including position and moving direction in the world coordinate system can be obtained in every camera frame from

$$T^{\text{human}} = T^{\text{human}}_{\text{camera}} * T^{\text{camera}}_{\text{robot}} * T^{\text{robot}}_{\text{map}} \tag{5}$$

Then, we observe real-time velocity based on the pedestrian's poses from the two neighboring frames.

$$\widetilde{V}_{\text{human},k} = \frac{T^{\text{human}}_{k} - T^{\text{human}}_{k-1}}{\sigma} \tag{6}$$

where σ is the sampling interval between frames. Next We estimate the real-time velocity state by calibrating the real-time observation with the last pedestrian velocity state.

$$V_{\text{human},k} = \alpha * V_{\text{human},k-1} + (1-\alpha) * \widetilde{V}_{\text{human},k} \tag{7}$$

We next modify the repulsive potential field and repulsion model mentioned above, making it more suitable for mobile obstacles. As is shown in Fig. 5, traditional APF builds a round repulsive potential field, which means a static obstacle gives radial repulsion to the robot around it, whose value is solely dependent on distance in between. By contrast, our SA-APF expects a moving obstacle to give larger repulsion where there is a higher likelihood of obstructing its moving. So, we develop an elliptical potential field with an offset center point. In the elliptical potential field, the relationship between long and short axis lengths is defined as follows

$$l_{\text{shortAxis}} = \beta * l_{\text{longAxis}} \tag{8}$$

In addition, the long axis is parallel to the velocity direction of the robot, so the greatest collision avoidance weight is given in this direction. The sum of $l_{\text{shortAxis}}$ and l_{longAxis} should be twice as large as the diameter of the original round potential field. The center of the potential field is also modified as

$$T_{\text{center}} = T_{\text{human}} + V_{\text{human},k} * predictTime \tag{9}$$

The offset of the center point gives the front of the pedestrian more weight in collision avoidance than the rear. In this

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.

modification, the repulsion function(4) still works. The only change is the definition of ρ_0 as is shown in Fig. 5(c).

D. Motion Control

Every time the robot starts to move from one disinfection site to another, it makes a global RRT* plan before moving. To make the path more smooth, we use bezier interpolation on the original global path. After that, the robot travels with the SA-APF local planner working. In detail, the points that are on a global path and have not been achieved by the robot provide an attractive force. In addition, all the temporary obstacles, including pedestrians and so on, provide repulsive force. The velocity of the robot is proportional to its received force in the potential field.

V. Experiment

A. Implementation Details of Vision Recognition

The training data comprises two sources: a pedestrian dataset from an open-source dataset [19], and an indoor high-touch object dataset developed by our team. The new dataset includes four categories: pedestrians, doorknobs, switches, buttons with a total of 1688 images.These objects are subject to frequent manual contact, thereby presenting an elevated risk of contagion. We enhanced the data diversity by utilizing data augmentation techniques such as random angle rotations, noise addition, contrast elevation, and cropping to increase the resilience of the model. For training, the dataset was randomly partitioned into a training set of 1519 images and a test set of 169 images, with a ratio of 9:1.

The vision recognition experiments were conducted on the Ubuntu 18.04 operating system, utilizing an Intel(R) Xeon Processor CPU and 56GB of RAM. The PyTorch 1.8.1 framework was employed for training, leveraging an Nvidia Tesla T4 graphics card with 16GB of video memory. The Python version used was 3.8.10, and the CUDA version employed was 12.0. The model training process involved 320 epochs, with a batch size of 32. The optimizer is Adam weight decay regularization (AdamW) [20]. The momentum factor is 0.937, the weight decay value is 0.0005, and both the initial momentum and IOU thresholds are set to 0.01.

We employed our algorithm on the Nvidia Xavier NX with jetson SDK 4.6.1, in conjunction with TensorRT 8.2 and the torch 1.8 environment. During the conversion of the PyTorch model to TensorRT model, we utilized the 'fp=16' parameter to achieve better acceleration.

B. Comparison Methods and Ablation Study

To verify the effectiveness of our framework, experiments are conducted to compare with other object detection algorithms, and ablation experiments are conducted for CBAM and C3Ghost modules. The above experiments are each conducted on the same dataset as mentioned above for model training, and the experimental conditions are kept consistent.

To verify the effectiveness of our proposed SA-APF algorithm, we compare it with the traditional APF algorithm. First, we build a map of disinfection space and use RRT* global planner to generate a path between disinfection sites with bezier interpolation in advance. Additionally, we set pedestrians along the path, who are traveling in front of the robot slowly, face-to-face encountering with the robot, or crossing in front of the robot with a randomized traveling velocity. Next, we use SA-APF and APF planners to navigate the robot moving to the aimed disinfection site in real-time 70 times, respectively, recording the time cost and whether collision happens or not. Besides, we also list important parameters in the SA-APF algorithm and find out their optimal values by grid searching.

TABLE I
Comparison with baselines and ablation study

Method	mAP 0.5:0.95	mAP @0.5	FPS	FPS TensorRT
YOLOv8s (baseline)	0.457	0.689	28.16	95.23
YOLOv8s-CBAM	0.492	0.722	27.77	96.15
YOLOv8s-C3Ghost	0.435	0.681	32.67	98.03
YOLOv8s-CBAM-C3Ghost	0.483	0.716	29.12	97.26
YOLOv8m	0.504	0.732	27.0	72.3
YOLOv8m-CBAM	0.512	0.740	24.69	74.07
YOLOv8m-C3Ghost	0.453	0.718	27.02	72.46
YOLOv8m-CBAM-C3Ghost	0.465	0.706	21.97	78.125
YOLOv8n	0.423	0.676	29.25	153.84
YOLOv8n-CBAM	0.439	0.707	29.23	119.04
YOLOv8n-C3Ghost	0.402	0.638	26.45	117.35
YOLOv8n-CBAM-C3Ghost	0.405	0.647	24.21	116.27

C. Evaluation Metrics

In this paper, we choose mean average precision (mAP),inference speed as the evaluation indices for visual recognition module. For navigation module, we use arrival time (the time for the robot to travel from the start point to the end point) and collision rate(the number of turns when collision happens among all turns) as the evaluation indices.

VI. Results and Analysis

A. Results of Vision Recognition

The comparison and ablation results are listed in Table I. In comparison to the baseline yolov8s model, our yolov8s-CBAM-C3Ghost model demonstrates a 1.6% increase in frames per second (FPS) by incorporating the attention and lightening modules, along with an additional 2.9% FPS improvement when utilizing TensorRT. Moreover, our method exhibits a enhancement of 2.7% in mAP@0.5 accuracy and a 2.6% improvement in mAP_0.5:0.95 over the baseline, highlighting the effectiveness of our approach.

Next, we test the model with the addition of CBAM alone, and the mAP@0.5 improved by 3.3% from baseline. The ablated model with the C3Ghost module alone revealed a 0.8% decrease in mAP@0.5, which suggests that the attention mechanism contributes to the improved accuracy, while the lightweight module decreases accuracy; when combined, they are able to have performance with lower model parameters as well as improved accuracy.

B. Results of Dynamic Pedestrian Avoidance

As shown in Fig. 6, the path produced by the global path planner is effective in avoiding complex static obstacles. The robot encounters three different types of pedestrians when

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.

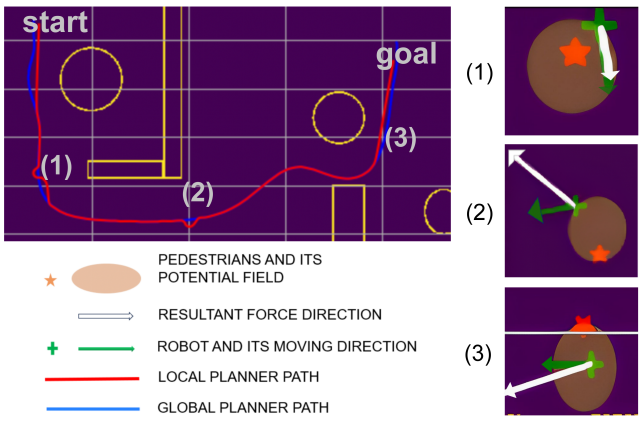

Fig. 6. Qualitative result of navigation and pedestrian avoidance. The global planner path and local planner path are observed in blue and red colored lines, and pedestrians are divided into three categories: (1) slow pedestrian in front; (2) intersect-with pedestrian; (3) face-to-face-with pedestrian.

TABLE II

COMPARISON BETWEEN OUR SA-APF AND TRADITIONAL APF

Methods	Arrival Time	Collision Rate
SA-APF	92.6sec	10%
Traditional APF	96.1sec	24.3%
p Value	0.029	0.044

traveling. It can be observed that deviations are made to avoid them, but when the risk of collision is removed, the robot returns to the global path.

Table II shows the quantitative comparison between our algorithm and the traditional APF. Our algorithm shortens arrival time by 3.6% with p value 0.029 by Independent Samples t test. In addition, it reduces collision rate by 58.3% with p value 0.044 by chi-squared test. The result shows our SA-APF is significantly safer and more efficient than traditional APF.

As mentioned in Section IV, the APF model is related to hyper-parameters, including α, β, and $predictTime$. So we will discuss how to choose these hyper-parameters next. We define hyper-parameters and optimization ranges as below. According to (7), $\alpha \in [0, 1]$ is the proportion of the last pedestrian velocity state in the prediction of the current velocity state. According to (8), $\beta \in (0, 1]$ is the ratio of the short axis to the long axis of the pedestrian potential field. According to (9), $predictTime$ is the prediction time of pedestrians. The larger the value of $predictTime$, the farther away the centers of the potential fields formed by pedestrians are from them. We use a grid-searching method to optimize hyper-parameters. We divide the value range of hyper-parameters into multiple discrete values and repeat experiments 10 times on each value candidate of each parameter, as illustrated in Fig. 7. As α increases, arrival time and collision rate (or called collision time) first surge and then stabilize. We infer that α is reasonable in the range of 0.6 to 1, which means the pedestrian motion model can be approximately simplified to a uniform-velocity model. As for β, arrival time and collision rate are lowest when β is 0.9, which means the potential field of a proper ellipse

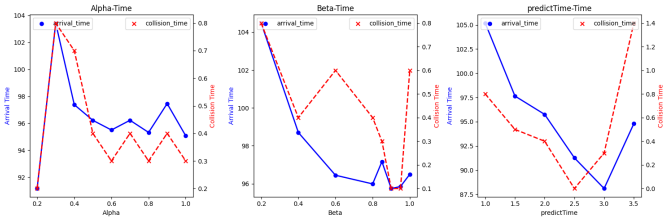

Fig. 7. Quantitative result of navigation and pedestrian avoidance determined by different hyper-parameters.

can result in shorter paths and fewer collision times than traditional round field. As for $predictTime$, the optimal value is between 2.5 and 3.0, which means a proper offset amount of center point position can improve navigation efficiency and safety.

C. Results of Real-World Experiment

We test the robot to perform a sterilization task through a real-world experiment. The experiment is conducted in a 20m$\times$15m corridor, in which there are 13 door handles and 9 switches, as well as 3 pedestrians traveling at different speeds and directions. The disinfection process is as follows: first, the robot scans the entire corridor to create a map. Then, we set up disinfection sites on the map and the robot reaches these sites in order. At Every site, the robot identifies high-touch areas and disinfects them.

In the experiment, our robot can successfully disinfect all the handles and switches, and disinfect only these areas, which means it is more efficient and cost-effective than existing disinfection robots. In addition, it takes active and flexible avoidance when encountering pedestrians, showing more socially friendly than existing disinfection robots.

VII. CONCLUSIONS

There is an increasing need for autonomous disinfection of densely populated indoor spaces. However, the majority of existing disinfection robots spray disinfectant everywhere evenly and are still immature to maneuver through congested, dynamic environments. Therefore, in this study, we build a new robotic system for disinfection to solve the mentioned challenges. We first propose a vision-based approach for precisely identifying high-touch locations in the environment, allowing the disinfection robot to achieve improved disinfection efficiency and reduce disinfection cost. Additionally, we propose SA-APF, a dynamic socially aware algorithm for pedestrian avoidance, which can anticipate pedestrian movement trends and avoid collisions with pedestrians more flexibly. However, there are still some unresolved issues in this work: in terms of visual recognition, the pedestrian's motion has a certain degree of uncertainty and the recognition rate on edge computing devices should be improved. In terms of navigation and obstacle avoidance, we have not considered humans' active avoidance of robots, which is also important in human-robot interaction. Although there are still problems to be solved, our work proposes a more automated and intelligent disinfection process, contributing to safeguarding public health in the post-pandemic era.

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] A. Gao, R. R. Murphy, W. Chen, G. Dagnino, P. Fischer, M. G. Gutierrez, D. Kundrat, B. J. Nelson, N. Shamsudhin, H. Su *et al.*, "Progress in robotics for combating infectious diseases," *Science Robotics*, vol. 6, no. 52, p. eabf1462, 2021.

[2] Y.-L. Zhao, H.-P. Huang, T.-L. Chen, P.-C. Chiang, Y.-H. Chen, J.-H. Yeh, C.-H. Huang, J.-F. Lin, and W.-T. Weng, "A smart sterilization robot system with chlorine dioxide for spray disinfection," *IEEE Sensors Journal*, vol. 21, no. 19, pp. 22 047–22 057, 2021.

[3] j. Zhao, Yu-Lin and Huang, Han-Pang and Chen, Tse-Lun and Chiang, Pen-Chi and Chen, Yi-Hung and Yeh, Jiann-Horng and Huang, Chien-Hsien and Lin, Ji-Fan and Weng, Wei-Ting, "A smart sterilization robot system with chlorine dioxide for spray disinfection," vol. 21, no. 19, pp. 22 047–22 057, 2021.

[4] L. Tiseni, D. Chiaradia, M. Gabardi, M. Solazzi, D. Leonardis, and A. Frisoli, "Uv-c mobile robots with optimized path planning: Algorithm design and on-field measurements to improve surface disinfection against sars-cov-2," *IEEE Robotics & Automation Magazine*, vol. 28, no. 1, pp. 59–70, 2021.

[5] D.-E. Schahawi, W. Zingg, M. Vos, H. Humphreys, L. Lopez-Cerero, A. Fueszl, J. R. Zahar, E. Presterl *et al.*, "Ultraviolet disinfection robots to improve hospital cleaning: Real promise or just a gimmick?" *Antimicrobial Resistance & Infection Control*, vol. 10, no. 1, pp. 1–3, 2021.

[6] Y. Guo, W. Chen, J. Zhao, and G.-Z. Yang, "Medical robotics: opportunities in china," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 361–383, 2022.

[7] A. IBHAZE and F. M. ADEKOGBE, "An ultraviolet germicidal irradiation autonomous robot," *Firat University Journal of Experimental and Computational Engineering*, vol. 2, no. 3, pp. 134–148, 2023.

[8] Y. Guo, F. Deligianni, X. Gu, and G.-Z. Yang, "3-d canonical pose estimation and abnormal gait recognition with a single rgb-d camera," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3617–3624, 2019.

[9] S. Mittal, "A survey on optimized implementation of deep learning models on the nvidia jetson platform," *Journal of Systems Architecture*, vol. 97, pp. 428–442, 2019.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[11] S. Silva, D. Paillacho, N. Verdezoto, and J. D. Hernández, "Towards online socially acceptable robot navigation," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 707–714.

[12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[13] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 2997–3004.

[14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[15] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[16] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[17] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1580–1589.

[18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[19] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *ACCV*, 2012.

[20] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

Authorized licensed use limited to: UNIVERSITY NOTRE DAME. Downloaded on November 07,2024 at 23:12:27 UTC from IEEE Xplore. Restrictions apply.