



OPEN

Optimal selective floor cleaning using deep learning algorithms and reconfigurable robot hTetro

Balakrishnan Ramalingam^{1✉}, Anh Vu Le¹, Zhiping Lin², Zhenyu Weng²,
Rajesh Elara Mohan¹ & Sathian Pookkuttath¹

Floor cleaning robots are widely used in public places like food courts, hospitals, and malls to perform frequent cleaning tasks. However, frequent cleaning tasks adversely impact the robot's performance and utilize more cleaning accessories (such as brush, scrubber, and mopping pad). This work proposes a novel selective area cleaning/spot cleaning framework for indoor floor cleaning robots using RGB-D vision sensor-based Closed Circuit Television (CCTV) network, deep learning algorithms, and an optimal complete waypoints path planning method. In this scheme, the robot will clean only dirty areas instead of the whole region. The selective area cleaning/spot cleaning region is identified based on the combination of two strategies: tracing the human traffic patterns and detecting stains and trash on the floor. Here, a deep Simple Online and Real-time Tracking (SORT) human tracking algorithm was used to trace the high human traffic region and Single Shot Detector (SSD) MobileNet object detection framework for detecting the dirty region. Further, optimal shortest waypoint coverage path planning using evolutionary-based optimization was incorporated to traverse the robot efficiently to the designated selective area cleaning/spot cleaning regions. The experimental results show that the SSD MobileNet algorithm scored 90% accuracy for stain and trash detection on the floor. Further, compared to conventional methods, the evolutionary-based optimization path planning scheme reduces 15% percent of navigation time and 10% percent of energy consumption.

Generally, human traffic is heavy on commercial premises like shopping malls, food courts, community clubs, and hospitals. Therefore, the chances of accumulating trash, stains (footprint stains and other forms of stains), and debris are quite high, raising safety and hygiene concerns. Hence, frequent cleaning is essential in commercial premises in pandemic situations like COVID-19. Due to long working hours, low wages, and unwillingness to work as a cleaner, workforce shortage has been a constant problem for cleaning and maintenance tasks in commercial premises^{1,2}.

In recent years, mobile cleaning robots have been widely used to clean and maintain commercial premises and help overcome the workforce shortage. It performs the cleaning task in public premises using two cleaning modes: auto cleaning/scheduled cleaning mode and selective area cleaning/spot cleaning mode²⁻⁵. Here, the auto cleaning/scheduled cleaning mode cleans the given workspace autonomously once a day or at a scheduled time interval using a pre-build floor map. In selective area cleaning/spot cleaning mode, the robot will clean only dirty areas instead of cleaning the whole region. In contrast with auto cleaning/scheduled cleaning mode, the selective area cleaning/spot cleaning mode could save energy by avoiding non-dirty areas, reducing the cleaning time, and enhancing the lifetime of the robot⁶. However, the selective area cleaning/spot cleaning mode hasn't been designed with autonomous operation, where the robot requires human assistance to perform the selective area cleaning/spot cleaning. Moreover, human-assisted selective area cleaning or spot cleaning is labor-intensive and time-consuming for larger workspaces. The cleaning supervisor needs to continuously monitor or inspect the environment to enable selective area cleaning/spot cleaning mode (according to human traffic in the environment or trash and stains detection).

Computer vision-based autonomous selective area cleaning/spot cleaning is a viable solution for overcoming the shortcomings mentioned above. Recently, Machine Learning (ML) and Deep Learning (DL) based computer vision algorithms are widely used for mobile cleaning robots for stain and trash detection and path planning tasks⁷⁻⁹. However, selective area cleaning/spot cleaning using onboard robot vision systems and generating the

¹ROAR Lab, Engineering Product Development, Singapore University of Technology and Design, Singapore 487372, Singapore. ²School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, Singapore. ✉email: balakrishnan@sutd.edu.sg

Author	Algorithm	Application
Bormann et al. ^{11,12}	Spectral residual filter	Dirt and Mud detection on floor
Andersen et al. ¹⁰	multivariable statistical approach	Dirt detection
Milinda et al. ¹³	Spectral residual filter + Maximally Stable Extremal Regions	Dirt and mud detection
Martinez et al. ¹⁶	median filter and the thresholding technique	Dust detection
Bala et al. ⁶	periodic pattern detection filter	Dirt region mapping
Zhihong et al. ²¹	(RPN) and VGG-16	Trash sorting
Rad et al. ²⁰	OverFeat GoogleNet	Trash detection
Gaurav et al. ²¹	AlexNet	Trash detection
Ramalingam et al. ⁷	SSD MobileNet	Debris detection and classification
Yin et al. ²	16 layer CNN	Food strains and trash detection
Chen et al. ²¹	Fast RCNN	Trash sorting
Proposed system	Deep SORT and SSD MobileNet	Selective area cleaning/spot cleaning

Table 1. Summary of related work.

optimal path generation has a lot of practical challenges. Searching and finding the dirty region itself increases the energy consumption and processing time by scanning the entire area of operation. Further, monitoring human traffic from a robot perspective and computing the selective area cleaning/spot cleaning region took more computation time and power.

Hence, there is a need for a novel and scalable approach to overcome the challenges mentioned above in the selective area cleaning/spot cleaning method. Therefore, this article proposes a novel selective area cleaning/spot cleaning framework for indoor floor cleaning robots using RGB-D vision sensor based Closed Circuit Television (CCTV) network, deep learning algorithm, and optimal complete waypoints path planning algorithm. Furthermore, the main contributions of this paper are divided threefold:

- Design and development of selective area cleaning/spot cleaning using deep learning-based computer vision algorithm and RGB-D vision sensor network. The deep learning algorithms find the selective area cleaning/spot cleaning region based on detecting stains and trash on the floor and tracing the human traffic patterns (common space: hall, entrance, and hallway/entryway).
- Compute the 3D location of stain and trash in world coordinates to estimate the complete waypoint coverage path planning by evolutionary methods to drive the cleaning robot to a selective area cleaning/spot cleaning.
- Validate the efficiency of selective area cleaning/spot cleaning framework in the real workspace with in-house developed reconfigurable floor cleaning robot hTetro.

The manuscript is organized as follows—section “[Related work](#)” elaborates the related works, and the proposed system is presented in section “[Overview of the proposed system](#)”. Finally, section “[Experimental results](#)” provides the experimental results, followed by the conclusion and remarks of this work.

Related work

This section presents the various dirt, stain, and trash detection schemes reported in the literature with detail summary (Table 1) and the optimal path planning methods used for cleaning robots. In¹⁰ Andersen et al. proposed a floor cleaning robot with a cleanness sensor using a computer vision algorithm and LED illumination. The sensor identifies dirty areas by analyzing surface images and producing pixel-by-pixel cleanness maps, applying a multivariable statistical approach. Borman et al.^{11,12} proposed a spectral residual filtering-based method for dirt detection. The dust threshold for various floor patterns is fixed based on maximal and minimal filter response. However, the dirt detection ratio is varied upon surface types. Mud and dirt on the floor are identified and classified for cleaning robots using Maximally Stable Extremal Regions (MSER) and Minimal Canny edge detection algorithms¹³. Lee and Banerjee¹⁴ suggested a solution for the overall cleaning schedule, which also generates an effective path for each cleaning cycle. Simulation-based Optimisation (SO) technique is applied for floor cleaning to avoid redundant paths and maintain tolerable dirt particle levels. In⁶ vision-based dirt detection was proposed for floor cleaning robot hTetro. Here, the authors use a periodic pattern detection filter to identify the dirt region on the floor and a tiling algorithm for generating the cleaning pattern. Grunauer et al.¹⁵ proposed an algorithm to detect dust spots based on Gaussian Mixture Model (GMM) applied to RGB images. This supports distinguishing the dust spots from the floor surface. Martinez et al.¹⁶ proposed a vision-based dust detection technique for the planar surface. The median filter and the thresholding technique were applied to distinguish the dust and the floor surface. Further, RANSAC algorithm was adopted to detect plane surfaces. However, as mentioned above, the scheme has significant practical limitations; for instance, the heavy detection of dirt depends on the floor background and cannot distinguish the dirt classes like trash or stain^{11–13}.

Deep Learning (DL) based object detection is an emerging technique. Nowadays, it has been widely utilized in the autonomous robot vision system, abandoned object detection in surveillance applications^{8,9}, human and vehicle tracking applications^{17,18}, to detect stains and trash in cleaning robots, classify garbage in waste management. In¹⁹ author evaluate the efficiency of various deep learning-based object detection framework for marine trash detection. The authors report that the Single Shot Detector (SSD) architecture has high detection accuracy

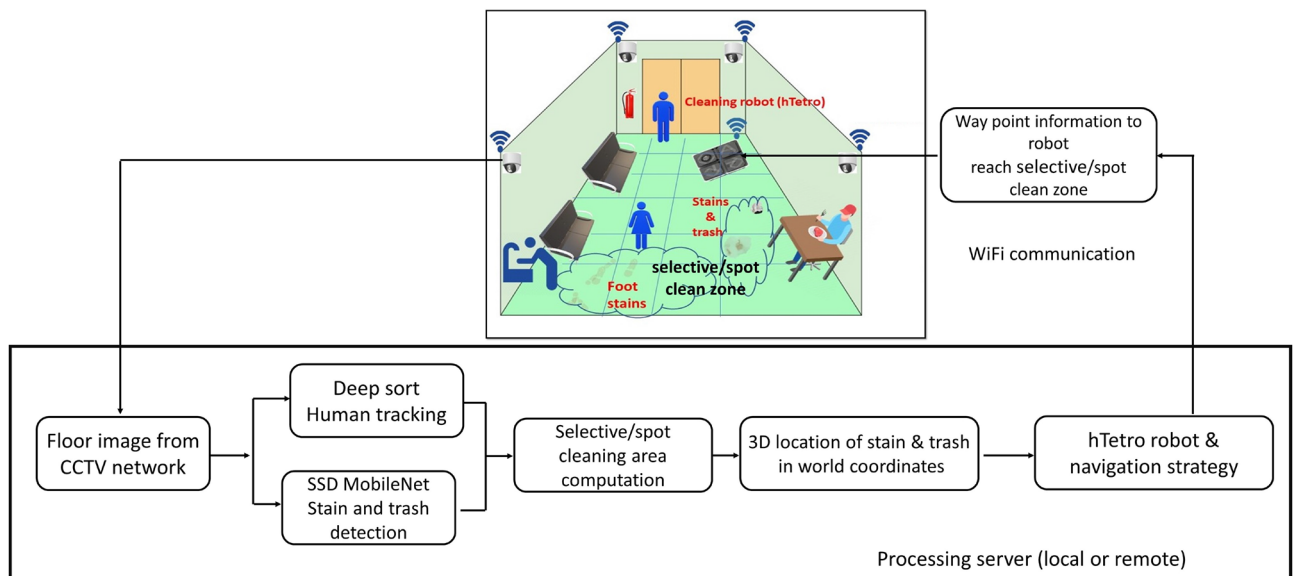


Figure 1. Overview of the proposed system.

than the Yolo v2 and Tiny Yolo object detection models. GoogLeNet-Overfeat object detection model was trained for outdoor trash detection proposed by Rad et al.²⁰. Here, the detection framework was trained with 18000 trash images containing waste papers, metal/plastic cans, and leaves. The trained model obtained a maximum of 77.3 % precision for detecting trash on the street. Ramalingam et al.⁷ proposed a debris detection and classification framework for floor cleaning robot application. The authors⁷ train the SSD MobileNet object detection algorithm with debris image dataset to detect and classify the trash and stains on the floor. Zhihong et al.,²¹ proposed an automatic trash sorting system using Region Proposal Network (RPN) and VGG-16 deep learning model. The authors report that the trained model had 3% miss detection and 9% false detection. Gaurav et al.²¹, proposed a smartphone app-based outdoor garbage localization technique where AlexNet CNN was trained to detect the outdoor garbage and detect the outdoor trash with 87% detection accuracy. Yin et al.² proposed Human Support Robot (HSR) based on deep learning for cleaning the tables. A darknet framework-based 16-layer CNN framework was developed. The model achieved 95% accuracy in detecting food stains and trash on the surface of the table. Further, a similar autonomous robot was developed using deep learning architecture, which has the capability to produce two-dimension trajectories from human kinesthetic instances^{22,23}.

Apart from dirt, stain, and trash detection, path planning is also a key function for floor cleaning robots to achieve efficient cleaning. In the literature, path planning for cleaning robots was widely studied. Generally, spiral and zigzag patterns are widely used to generate the path for cleaning robots, aiming to maximize the area's coverage. However, these patterns have significant limitations in covering the constrained areas, such as narrow space, and are only suitable for fixed morphology platforms⁷. Further, computing the optimal shortest path is another key challenge for the selective area cleaning/spot cleaning task. Reconfigurable morphology robots can cover the maximum area and arbitrary workspace compared with a fixed morphology robot and are more suitable for selective/spot-cleaning tasks. Self-reconfigurable robots have attracted significant attention, and the development of autonomy systems mostly focused on autonomous motion control and mechanism design. There are only a few works in path planning for self-reconfigurable robots, e.g., the work in²⁴ proposes a complete coverage path planning model trained using deep reinforcement learning (RL) for the tetromino based reconfigurable robot platform to simultaneously generate the optimal set of shapes for any pretrained. Research on path planning so far focuses on fixed-morphology robots and considered robots as a single point. But, representative path planning algorithms can be broadly classified as classical graph search algorithms (Dijkstra, A*, D*, and D* Lite)^{25–28}, bio-inspired search (genetic algorithm, ant colony optimization, and particle swarm optimization)^{29–35}, geometric algorithms (probabilistic road map, visibility graph, rapid-exploring random tree, cell decomposition, Voronoi graph)^{36–41}. However, selective area cleaning/spot cleaning path planning for re-configurable floor cleaning robots is a new approach. It has not been explored yet. This research describes the optimal shortest waypoint coverage path planning method using evolutionary-based optimization technique for re-configurable floor cleaning robot hTetro to execute the selective area cleaning/spot cleaning task.

Overview of the proposed system

Figure 1 shows the overview of our selective area cleaning/spot cleaning framework with a functional block diagram. It is developed for floor cleaning robot indoor selective area cleaning/spot cleaning task. Generally, high human traffic areas are more likely to accumulate more trash and foot stains. Hence, two strategies are combined to efficiently find the indoor environment's selective area cleaning/spot cleaning regions. It includes tracing the human traffic patterns and detection of stains, foot stains, and trash on the floor.

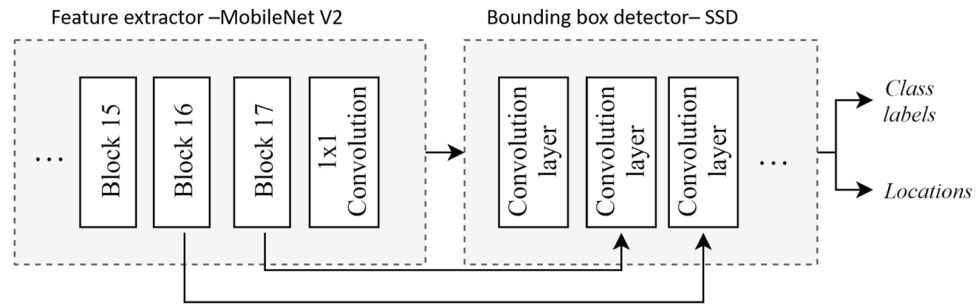


Figure 2. SSD MobileNet V2.

The proposed system functional block diagram is composed of an image data collection module from the RGB-D vision sensor network, deep learning-based computer vision algorithms, and an optimal path planning module for the hTetro robot. WiFi-enabled surveillance camera setup (Realsense D345 fitted in roof corners) captures human movements indoors and continuously captures the floor images. The camera system is fixed in the corners of the roof, and the Field of View (FoV) is calibrated with the 3D map built by the hTetro robot base frame. The cameras are connected to an external server through a WiFi router to transfer the captured video streams. A deepSORT human tracking algorithm was used to trace the human traffic pattern. SSD MobileNet object detection algorithm was trained to detect foot stains on the high human traffic region, common stains, and trash on the floor. The selective area cleaning/spot cleaning algorithm is run on an external server built with a high-performance GPU. The server collects the indoor images continuously through a WiFi router and performs the human traffic pattern detection analysis, stains, and trash detection task. According to the human traffic pattern tracing, stains, and trash detected on the floor, the framework generates the navigation strategy and send the way point coordinates to hTetro robot to perform the selective area cleaning/spot cleaning task. The robot uses waypoint information (computed through image coordinate to robot footprint coordinate) and odometry data to reach the selective area cleaning/spot cleaning location and perform the cleaning task. The details of each component and integration methods are described as follows.

Deep SORT human tracking. The Deep SORT is an improved version of the Simple Online and Real-time Tracking (SORT) algorithm. Generally, deep sort is used for human tracking applications. In our work, Deep SORT is used to monitor the human traffic pattern tracing in indoors through RGB-D vision sensor collected image and map the high traffic area to identify the selective area cleaning/spot cleaning region. The Deep SORT tracking framework was built using the hypothesis tracking technique with the Kalman filtering algorithm and DL-based association metric approach. Further, the Hungarian algorithm is utilized to resolve the uncertainty between the estimated Kalman state and the newly received measuring value. The tracking algorithm uses the appearance data to improve the performance of sort⁴². In this work, the bounding box coordinates in image plane (X_{min} , Y_{min} and X_{max} and Y_{max}) of human detection are passed to Deep SORT with image frame for tracking the human movements. According to bounding box coordinates and object appearance, deep SORT assigns each human detection ID and performs the tracking. A selective area cleaning/spot cleaning algorithm used the tracked information to compute the cleaning spot.

Stain and trash detection framework. A Single Shot Detector (SSD) MobileNet V2 end-to-end object detection framework is adopted to detect the stains, foot stains, and trash on the floor. The functional block diagram of SSD MobileNet is shown in Fig. 2. Here, feature extraction is done using the MobileNet model. Bounding box prediction is made using a meta-architecture SSD. While training, RMSprop algorithm⁴³ is used for loss optimization. The framework was trained using a transfer learning scheme; the last layer of the detector head is modified to detect the stains, foot stains, and trash.

Feature extractor. MobileNet v2 is used as the base network that performs feature extraction tasks suitable for mobile and edge devices. Figure 2 shows the MobileNet v2 architecture. It comprises multiple residual bottleneck layers. These layers use 3×3 in depthwise convolution layers, which are more efficient than standard convolution layers. They also employ 1×1 convolutions instead. The architecture also uses ReLU6 layers, which are ReLU layers with an upper limit of 6. The upper limit prevents the outputs from scaling to a huge value, reducing the computation cost.

SSD architecture. Single-Shot Detector (SSD) is an object detection algorithm. It outputs the object's location through a bounding box in an image with the confidence of the prediction. The bounding box coordinates have been translated into real-world coordinates and fed to a trajectory planning algorithm to generate the selective area cleaning/spot cleaning path. The SSD algorithm uses the MobileNet feature map to generate the anchor boxes. The final few layers (feature map) from the feature extractor are used to predict trained objects' locations. In the SSD architecture, the input is passed through different convolution layers of different sizes. These layers

decrease in size progressively through the network. The purpose of these layers is to enable the network to detect objects of different shapes and sizes.

While training, confidence loss $L_{confidence}$ and localization loss $L_{location}$ Eq. (1) was calculated for each prediction. The prediction error of class and confidence is given by confidence loss. The localization loss is a smooth loss between the predicted bounding box value and the ground truth value. The value α is used to reduce the overall loss's influence and balance confidence and location loss.

Root Mean Squared gradient descent algorithm (RMS) is used for optimizing both the losses. The weights w_t are computed at any time t using the gradient of loss L , g_t and gradient of the gradient v_t Eqs. (2)–(4). The hyperparameter $\beta = 0.9$, $\eta = 0.02$ are utilized for estimating the momentum and gradient values, while a small value near to zero ϵ is used for avoiding divide-by-zero errors.

$$L = \frac{1}{N} (L_{confidence} + \alpha L_{location}) \quad (1)$$

$$v_t = \beta v_{t-1} + (1 - \beta) g_t^2 \quad (2)$$

$$\Delta w = \frac{-\eta}{\sqrt{v_t + \epsilon}} \times g_t \quad (3)$$

$$w_{t+1} = w_t + \Delta w \quad (4)$$

Algorithm 1: Selective area cleaning/spot cleaning algorithm

Input:

S is a selective area cleaning/spot cleaning time slot given by user for execute the selective area cleaning/spot cleaning

Output:

s_map selective area cleaning/spot cleaning map

Initialize:

C is an Camera node

F is an image frame

$dirt_map$ is an array used to store the detected dirty region bounding box coordinates

$human_foot_map$ is an array used to store the human track bounding box coordinates

Initialize End**Begin:****while** (1) **do**

F = Capture Frame (C)

B = SSDMobileNet(F)

time-slot = time-slot + 1 //time-slot is variable increment only every one minute

if (human detection in B) **then**

Deep_Sort_human_tracking (Capture Frame (C)) // track until human exits from the frame

Update (human-foot-map) // Update the human tracking coordinates on $human_foot_map$

else if (stain | foot stain| trash in B) **then**

Read the detection (F_t) // F_t : current frame

Compare_detection (F_t , F_{t-1}) //exclude duplicate detection F_t : current frame, F_{t-1} : previous frame

Update (dirt-map) // update the coordinates in dirt-map

else if ((time slot == S) && (update in human-foot-map | dirt-map)) **then**

w_foot_map = convert image coordinates to world coordinates (human-foot-map) // w_foot_map hold world coordinates

w_dirt_map = Convert image coordinates to world coordinates (dirt-map) // w_dirt_map hold world coordinates

s_map = Generate trajectory plan (w_dirt_map , w_foot_map) // w_dirt_map hold selective area cleaning/spot cleaning world coordinates

robot_navigation = Active the cleaning sequence (waypoint_sequence)

else

goto → *Begin*

end if**end while**

3D location of stain and trash in world coordinates. After finding the 2D pixel coordinates in the RGB image frame of stain, trash, and human traffic pattern, the system will derive the world frame's corresponding 3D position. Since, the perception system is a stereo camera with both an RGB image sensor and a depth image sensor, they are of the same resolution, and each pixel in the RGB image corresponds to the pixel in the depth image. This enables us to map the dirt object's localized pixel coordinates to their depth value. When the depth images are subscribed, the image will be filtered via an adaptive directional filter⁴⁴ to remove noise. After filtering the depth image, the depth value of the dirt objects, d_o , has been derived by Eq. (5) as follows:

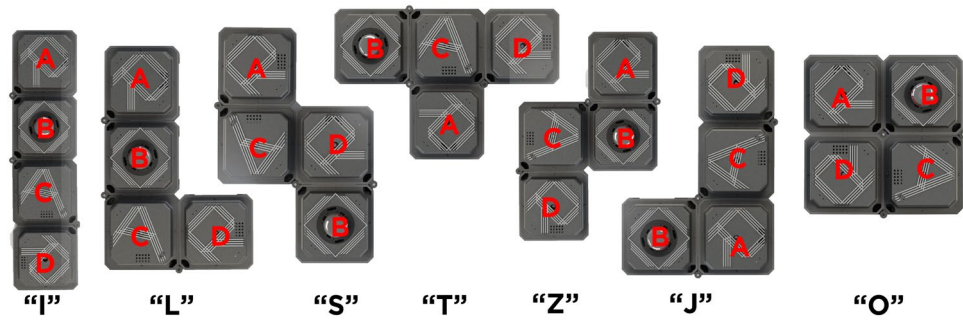


Figure 3. Seven configurations of hTetro.

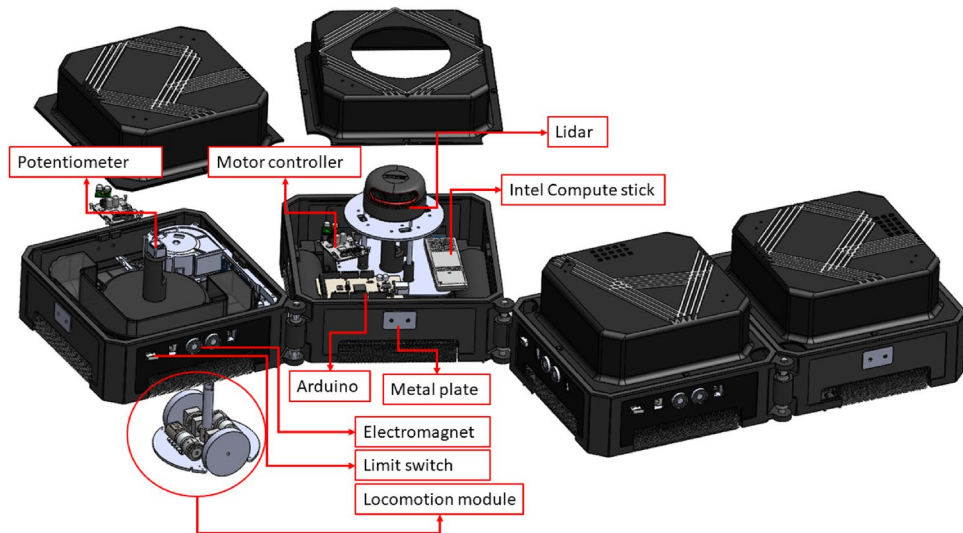


Figure 4. hTetro hardware components.

$$d_o = \frac{\sum_{\Omega} d_i}{\Omega} \tag{5}$$

where, Ω represents the sum of pixels within a 10×10 window, and d_i represents the value of the depth in the pixel. The dirt objects in real-world coordinates will be estimated by converting the X and Y pixel coordinate system to x and y world coordinate system built by the Robot Lidar sensor.

RGB-D Sensor parameters are derived through camera calibration techniques to give the camera an intrinsic and extrinsic matrix⁴⁵. Camera calibration methods give us the focal length, f_x , and f_y , of the RealSense RGB-D sensor and the optical center, c_x , and c_y , of the RealSense RGB-D sensor in its respective x and y coordinate to get the intrinsic camera matrix, I . The RealSense RGB-D sensor in-built functions also provide us with the translation vector, K , and rotation vector, Q . The extrinsic matrix, S , can be derived where $S = [QK]$. Further, it will convert from the pixel coordinate system to the world coordinate system through Equation (6).

$$p = I * [QK] * P = I * S * P. \tag{6}$$

where P is the world coordinates, p is the pixel coordinates.

Overview of hTetro robot and navigation strategy. hTetro is an in-house developed re-configurable floor cleaning robot (Fig. 4). This robot consists of a chain-type with four cube shape modules connected by three hinges at the corner of blocks, as shown in Fig. 3. In hTetro, blocks 1, 3, and 4 each hold a cleaning module, while block 2 contains the primary electronic circuits. The robot blocks could be easy to remove and add a degree of freedom. The hinges permit the hTetro robot in shape-transferring and a couple of configurations. The shape-shifting allows the robot to cover the constrained space more efficiently than fixed-form robots. For instance, the robot can cover the bigger space with the default O shape and change shapes flexibly from the O to I shape to cover the narrow spaces. The three hinges connected between robot blocks are used for shape-shifting during the reconfiguration.

The locomotion modules at each block are programmed to operate synchronously with others according to the robot's transformation, and each of the modules is amenable for steering and moving the hTetro. Each

block contains two geared DC motors with encoders for balance locomotion. The operation voltage rating of the DC motors is 7.4V. In hTetro, a potentiometer type absolute encoder was used in each block to sense the shape change action and block's heading angle. When the hTetro is involved in the transformation actions, the servos located at the hinges drive the modules by rotation clockwise and anticlockwise to do the reconfiguration into seven distinct shapes - 'I, L, S, T, Z, J, O', as shown in Fig. 3. The low-level controller (Arduino mega) was used to measure the output current from the absolute encoder through the A/D Input pin and fine-tunes the shape change action. This sensing data helps ensure that the units are aligned and moving in the same direction. The servo motors run at 14.8V, and the high torque is 77 kg. cm. The servo has enough torque to hold the hTetro block masses in morphologically changing and fastening the blocks in locomotion.

Hardware and software components. hTetro hardware architecture was developed with a single-board computer Intel compute stick and low-level micro-controller Arduino mega 2560 (Fig. 4). The Intel compute stick is powered with an Intel Core m3 processor, 4GB of memory, and 64GB of eMMC flash storage. It was used as an hTetro central processing unit, runs with an operating system of Ubuntu 20.04, and uses ROS Melodic⁴⁶ as middleware. The hTetro navigation algorithms run alongside the ROS middleware in the compute stick and control all the autonomy-related operations of the hTetro. The navigation algorithm receives the waypoint information from the remote/local server through on board WiFi module and generates the control command to the low-level control unit to execute the selective area cleaning/spot cleaning task.

Arduino-Mega was used as a low-level controller. It works at 5 volt supply voltage and operates at 16 Mhz clock frequency. The controller receives the commands from the central processing unit through ROS serial bridge function and generates the appropriate actions for the motor drives and other local electronic components. The rosserial ROS package uses Arduino's universal asynchronous receiver/transmitter (UART) communication and converts the board to a ROS node that can output ROS messages and subscribe to messages as well.

Optimization of trajectory. The 3D points at the centers of detected objects are defined as waypoints the robot needs to visit to do the appropriate selective area cleaning/spot cleaning region. The robot will derive the optimal navigation trajectories defined as Complete Waypoint Coverage Planning (CWCP) to visit all the waypoints. With N unsequenced waypoints detected from the perception unit, the navigation trajectory η is defined as follows: $\eta = \{(W_1^s, W_1^g), (W_2^s, W_2^g), \dots, (W_N^s, W_N^g)\}$. The trajectory η consists of pairs of waypoints with the directions which indicate the order of navigation. The initial waypoint is labeled W_1^s , and the final waypoint is labeled W_N^g . This trajectory representation visits each waypoint among N waypoints once. The cost to navigate from source waypoint to destination waypoint equals to summation of mass of each robot m_i block multiplied with 3D displacement to move blocks from source (x_i^s, y_i^s, z_i^s) to goal (x_i^g, y_i^g, z_i^g) as shown in Eq. (7). The total cost of the trajectory can be summarized by all the waypoint pair components as shown in Eq. (8). The Pareto trajectory is defined as $\hat{\eta}$, and the objective of the proposed CWPP problem is derived by Eq. (9).

$$C(W_n^s, W_n^g) = \sum_{i=1}^4 m_i \sqrt{(x_i^g - x_i^s)^2 + (y_i^g - y_i^s)^2 + (z_i^g - z_i^s)^2} \quad (7)$$

$$C_{total} = \sum_{(W_n^s, W_n^g) \in \eta} C(W_n^s, W_n^g) \quad (8)$$

$$\hat{\eta} = \underset{\eta}{\operatorname{argmin}} C_{total} = \underset{\eta}{\operatorname{argmin}} \sum_{(W_n^s, W_n^g) \in \eta} C(W_n^s, W_n^g) \quad (9)$$

There are many possibilities to form the trajectories and find the optimal route to minimize the fitness function; the classic Travelling Salesman Problem (TSP) model can be considered an NP-hard problem. Similarly, brute force search can be used for iterating every possible path with factorial time complexity $O(n!)$, but performance gradually decreases with the large input data size. Dynamic programming through Held-Karp algorithm⁴⁷ reduces the complexity of TSP to $O(n^2 2^n)$ time. However, the method is still inappropriate for the presented scenario where multiple detected waypoints are in the working workspace. Evolutionary approach Ant Colony Optimization (ACO)^{48,49} is adopted to find the optimal trajectory connecting the waypoints, $\hat{\eta}$ in Eq. (9) within a reasonable time. Optimization based on probabilistic techniques are the main idea of ACO. By tuning the ant colony decisions at the waypoint and the constant updates on the pheromones left along the temporary trajectory, the ACO algorithm has proven to be a feasible method to derive the Pareto-optimal solution of the TSP while eliminating the calculation burden.

Experimental results

This section presents the experimental procedure and outcome of the selective area cleaning/spot cleaning framework tested on the hTetro robot. The algorithm has two pre-processing steps: training the trash and stain detection framework using a custom dataset and mapping the indoor environment for hTetro autonomous navigation.

The dataset was prepared in-house with different image resolutions. The dataset has four classes (Stain, Foot stain, Trash and Human); each class has 1200 images. Most of the images are created in-house with various floor backgrounds, and few images are collected online through Bing image search. Here, in-house collected images are captured in an overhead perspective with a Realsense RGBD camera with a different angle and

different illumination conditions. Then, the captured images are labeled for training the detection framework (SSD MobileNet). Further, to reduce the CNN learning rate and avoid over-fitting. The data augmentation (rotation, scaling, and flipping) is applied in collected images.

The standard method of the K-fold (here K = 10) cross-validation process is used to evaluate the detection model. In K-fold cross-validation, the labeled images are split into ten sets, and nine sets were used to train the detection network, and one set is utilized for evaluation. These steps run for ten rounds to ensure all the data were used for both training and testing to eliminate the biasing conditions. This process is done to remove any bias due to the particular training and testing data set to split. The experimental results are obtained from the detection model with reasonable accuracy. The deep learning models were developed in Tensor-flow 1.9 Ubuntu 18.04 version and trained using the following hardware configuration Intel core i7-8700k, 64 GB RAM, and Nvidia GeForce GTX 1080 Ti Graphics Cards. Standard statistical measures such as accuracy (10), precision (11), recall (12) and $F_{measure}$ (13) were used to evaluate the proposed algorithm through confusion matrix elements.

$$Accuracy(A) = \frac{tp + tn}{tp + fp + tn + fn} \quad (10)$$

$$Precision(P) = \frac{tp}{tp + fp} \quad (11)$$

$$Recall(R) = \frac{tp}{tp + fn} \quad (12)$$

$$F_{measure}(F_1) = \frac{2 \times precision \times recall}{precision + recall} \quad (13)$$

Here, tp , fp , tn , fn represents the true positives, false positives, true negatives, and false negatives, respectively, as per the standard confusion matrix.

Validation of selective area cleaning/spot cleaning. The selective area cleaning/spot cleaning framework was validated in three different indoor environments with the sizes of 10×10 , 8×8 , 15×8 feet. The overhead cameras were fixed corner of the roof in the testbed for tracing the human movements and capturing the floor images. The captured floor images are transferred to the external server, running a selective area cleaning/spot cleaning framework. The experiment was carried out with different lighting conditions, and results are shown in Figs. 5 and 6. In this trial, the overhead camera run at 30fps, and the image resolution was 640×480 pixels. The floor image was captured at 7 feet height from the ground, and the camera head is inclined at an angle of 45 degrees downwards.

The stain is detected by a light turquoise color bound box, foot stains were marked as white rectangle box and trash marked by the sky blue bounding box. Figures 5 and 6 shows the human traffic pattern detection results. Here, the human movements were tracked by bounding boxes, and the traffic pattern is indicated in the orange line. The experiment results indicate that the selective/ spot cleaning framework detects the human, stain, and trash with a higher confidence level of 92%, 90%, 95%, and 98%, respectively. Further, the model detection accuracy was estimated by statistical measures, which are given in Table 2.

The statistical measurement result shows that the framework has identified the stain with an average of 93.44%, foot stains with 91.69%, trash with 93.0 %, and human with 97.0%, respectively. In this experiment, we observe that stain and foot stain detection confidence level and statistical measure values are relatively low compared to other classes due to less visibility of objects and complex texture shape. Further, the computation cost of the proposed system was estimated via inference time; in this test, the framework took 0.05 seconds to process the 640×480 images. The statistical study ensures that the proposed system was stable for different lighting conditions and also work in realtime.

Comparison with other object detection frameworks. The efficiency of the SSD mobilenet stain trash detection was evaluated with other benchmark object detection algorithms include Faster RCNN ResNet⁵⁰, Faster RCNN Inception⁵¹, and Yolo v2 frameworks. The same dataset was used for estimating the detection and classification accuracy of three networks for a period at the same time. Statistical measure parameter was used to compare the performance model. Table 3 shows statistical measure results of each framework. In addition to that, timing analysis was performed to calculate the detection time taken by each network. NVIDIA Quadro P4000 graphic card was used for testing the proposed algorithm, as shown in Table 4.

Best detection accuracy was obtained using Faster RCNN ResNet and Faster RCNN inception compared to other models. Yolo v2 inference time is comparatively low but showed poor detection accuracy. SSD-MobileNet has obtained a better trade-off between detection accuracy and inference time. Hence, SSD- MobileNet shows optimum results when used in realtime applications for object detection. Optimized SSD MobileNet is highly suitable for low-power resource-constrained embedded systems, making it suitable for realtime applications.

Complete Waypoints Coverage Planning (CWCP) evaluation. We validated the efficiency and performance of the selective area cleaning/spot cleaning method in terms of navigation efficiency by deploying the cleaning robot system to clean optimally the detected tracked human traces in the selected indoor testbed sites. Specifically, depending on the human interaction activity in the environment, a location map is constructed



Figure 5. Real-time detection of stain, foot stain, trash.

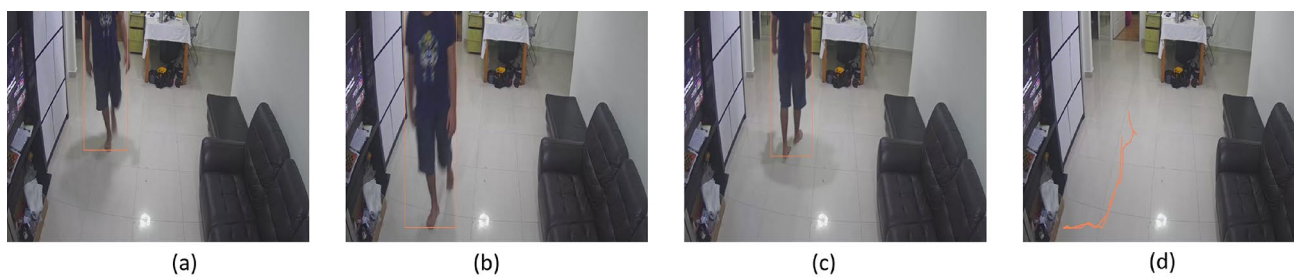


Figure 6. Human movement tracking, (a–c) human movements and (d) tracking result.

Class	P	R	F1	A
Stain	93.07	92.0	91.15	93.44
Foot stain	92.11	90.22	90.04	91.69
Trash	94.55	93.89	93.32	93.0
Human	97.02	96.3	96.03	97.0

Table 2. Statistical measure results.

Class	Faster RCNN ResNet				Faster RCNN Inception				Yolo V2			
	P	R	F_1	A	P	R	F_1	A	P	R	F_1	A
Stain	95.96	94.34	94.11	95.04	94.41	94.38	94.11	94.0	89.24	88.11	87.66	89.15
Foot stain	94.45	94.13	94.01	94.24	93.50	92.07	92.06	92.00	85.00	84.33	84.09	84.50
Trash	96.63	96.29	96.17	96.44	96.20	95.91	95.43	96.03	89.04	89.25	89.06	90.00
Human	97.35	96.92	96.74	97.00	96.87	96.43	96.14	96.54	93.01	91.25	91.03	92.00

Table 3. Comparison with other object detection frameworks.

Model	Inference time for 60 images (s)
Faster RCNN ResNet	122
Faster RCNN Inception	110
Yolo V2	48
SSD MobileNet	54

Table 4. Execution time comparison.

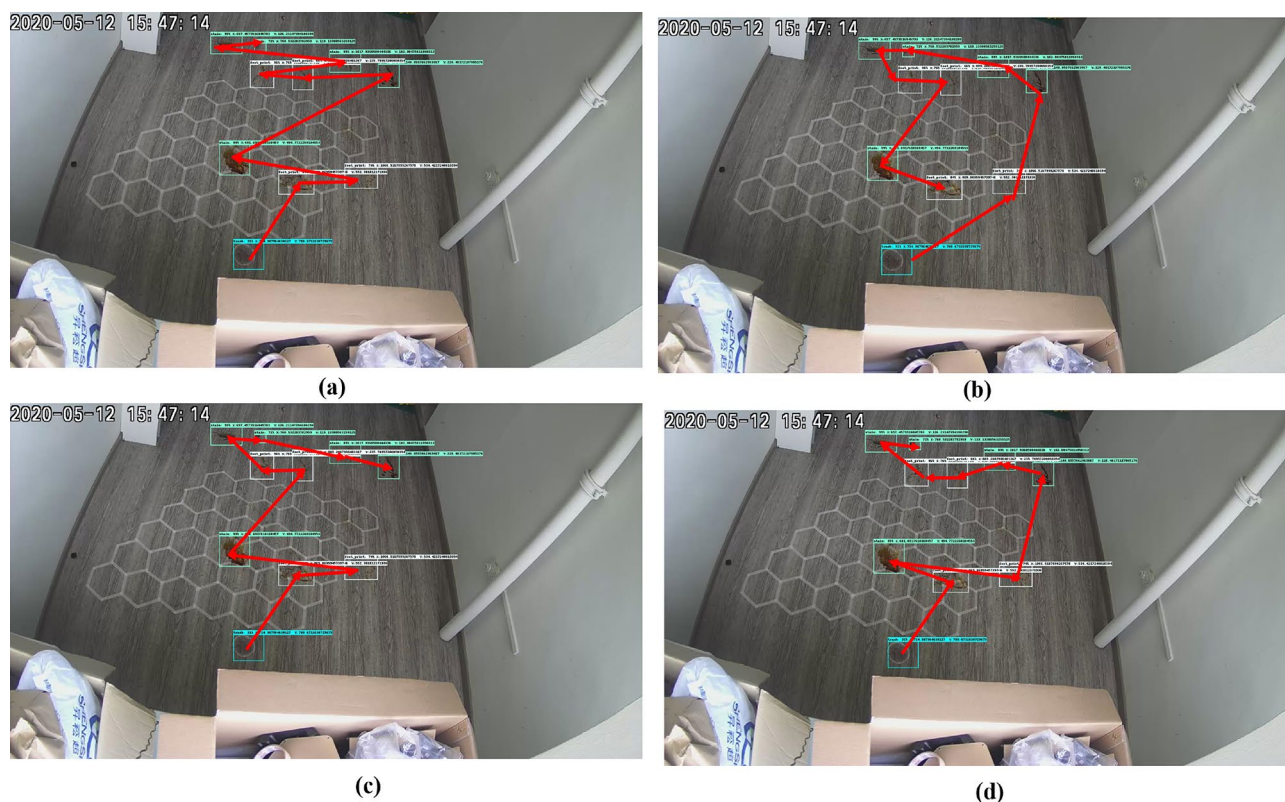


Figure 7. The trajectories of the tested method for workspace-1. (a) zigzag; (b) spiral; (c) greedy search; (d) ACO.

on the grid cells by the proposed framework. The location map distribution result is used as input for a hTetro reconfigurable cleaning robot to decide navigation strategies optimally.

The Complete Waypoints Coverage Planning (CWCP) framework derives the smallest costweight and energy consumption for real environments testbed. The trajectories denoted as red linking arrows generated by all tested methods including zigzag, spiral, greedy search with 100 iterations, and ACO algorithms for all testbed workspace-1 with 9 detected waypoints and workspace-2 with 11 detected waypoints are shown in Figs. 7 and 8, respectively. In the experiment, the best parameter values are found by the experimental approach by 200 trials. After tuning, the ACO parameters set is evaporation probability = 0.9, and ant agents = 220. The terminal's criteria are triggered when the costweight during execution is not improved for ten iterations or the execution loop runs over the 500 iterations.

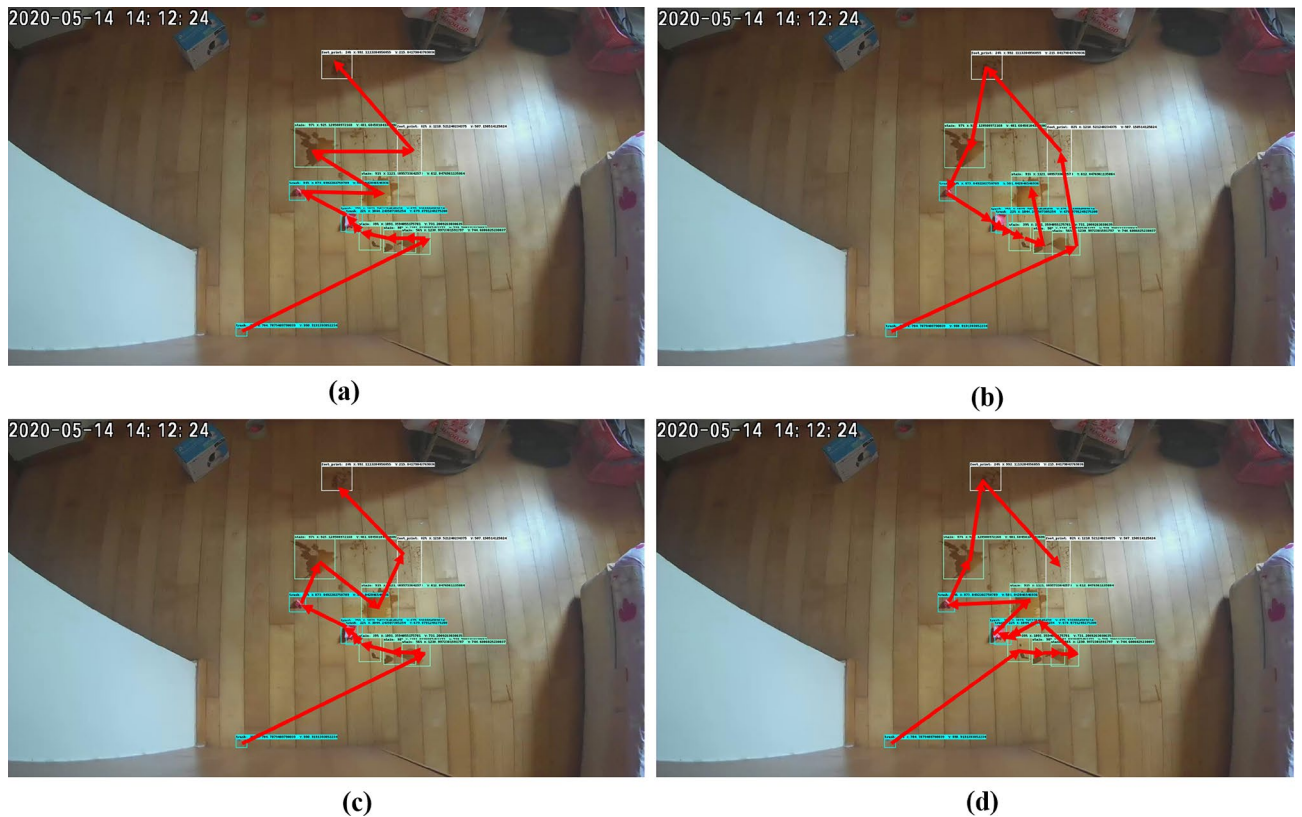


Figure 8. The trajectories of the tested method for workspace-2. (a) zigzag; (b) spiral; (c) greedy search; (d) ACO.

Approach	Cost weight (Kgcm)	Consumed energy (Wh)	Path generation time (s)	Running time (s)
Zigzag	1951.69	0.014344444	0.01	182.25
Spiral	1962.28	0.016172222	0.05	194.16
Greedy search	1829.63	0.010866667	5.25	135.95
ACO	1725.91	0.0098916667	0.92	125.15

Table 5. The numerical comparisons for tested CWPP methods.

The trajectories with found costweight and the time to generate as well as the running time to complete these found trajectories of the tested method for the two testbed workspaces are represented in Table 5, respectively. Because of the straightforward approach in generating the paths, the path generation time of the zigzag and the spiral are smaller than Greedy Search and ACO, but the zigzag and spiral methods link the defined locations by position-wise orders; hence the costweight is greater than greedy search methods and significantly greater than ACO based Complete Waypoints Path Planning (CWPP). Noted that a greedy algorithm is an algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage. Since it may be stuck at local minimal so we run the greedy algorithm for 100 iteration with random beginning waypoint and select the best solution. That makes that greed search requires more path generation time.

In terms of running time when deploying the found trajectory to the real robot, the Zigzag and Spiral need more time to cover the defined waypoint than the greedy search. Together with the least costweight, the running time of ACO-based CWPP is the shortest one.

After the perception system has defined the waypoint locations in real testbed sites, the energy efficiency during the selective area cleaning/spot cleaning process is validated by recording in realtime the energy consumption. The current sensors connected to the main lithium-ion battery with the specs of 14.4 V, 3000 mAh are logged during robot operation. These current values are integrated with the running time to find power consumption. In the selective area cleaning/spot cleaning mode, the hTetro is set at O shape and to clear these waypoints in order. The robot can change to I shape to access the narrow space area by the environment evaluation from the Lidar sensor. The action of shape-changing is adapted to a dynamic environment. The robot modules such as perception, locomotion, and shape-shifting communicate under the controls by the ROS system⁵². To archive the smooth locomotion with the different kinematic design of hTetro, the moving command generated by the PID controller is sent to motor drivers to issue appropriate velocity for the DC motors. Specifically, the robot

should keep the planned trajectories and overcome the disturbances while colliding with static and dynamic obstacles. These tasks are significant challenges since the synchronization between hTetro blocks locomotion must be accomplished. To fulfill the kinematic constraints, the control method is based on the Instantaneous Center of Rotation (ICR). The placement of ICR makes use of the desired positions and velocities information provided by the central computing unit based stains pattern. After that, the corresponding steering angle and required individual velocity are calculated based on the ICR location. A velocity regulator is then used to limit the velocity of each motor. Due to the placement of LIDAR on the second module's chassis, the orientation of the whole robot is defined according to it. The instantaneous center of rotation's desired location is defined based on the desired position, current position, and the desired velocity. It is represented in the polar coordinate using to avoid numerical singularity. The hTetro locates itself within the pre-built map using enhanced extended Kalman filter(EKF)⁵³ for multi-sensor fusion makes sure hTetro can navigate from source and destination goad correctly even if sensors noise is received, or any perception sensors struggle the malfunction issues.

From Table 5, if the robot clears the found path of the method with a smaller costweight value, the less energy spent is observed. Specifically, since producing the longest costweight, the zigzag draws the highest energy, then the spiral follows the following position with a slight amount lower. The best CWCP method in both energy and time saving is ACO. It yields about 10% and 15% lower than the greedy search in energy and time spent, respectively. The results ensure that the proposed CWCP with ACO is possible for selective area cleaning/spot cleaning by hTetro.

Conclusions

This work proposed a selective area cleaning/spot cleaning technique for reconfigurable floor cleaning robot hTetro using deep learning-based computer vision algorithms and an optimal complete waypoints path planning method. SSD MobileNet object detection algorithm and Deep SORT real-time human tracking algorithm were used to find the human traffic patterns and detect the stains, foot stains, and trash on the floor. The proposed scheme was tested with two different testbeds with hTetro reconfigurable floor cleaning robot and evolutionary optimization based optimal trajectory generation algorithm. The experimental results showed that the trained SSD model detects the stains, foot stains, and trash on the floor with a high confidence score, and its result is used to find the selective area cleaning/spot cleaning region. Further, the SSD MobileNet framework's stain and trash detection efficiency was evaluated with other CNN-based object detection algorithms and observed that SSD MobileNet obtained a better trade-off between detection accuracy and low inference time for detection of stains and trash. The overall experimental results show that the proposed framework is more suitable for enabling the autonomous selective area cleaning/spot cleaning function for the reconfigurable floor cleaning robot hTetro.

Data availability

'The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.' Correspondence and requests for materials should be addressed to B.R.

Received: 26 January 2022; Accepted: 26 August 2022

Published online: 24 September 2022

References

1. The 5 most unwanted jobs in singapore, accessed 6 Jan 2020; <https://sg.finance.yahoo.com/news/5-most-unwanted-jobs-singapore-160000715.html>.
2. Yin, J. *et al.* Table cleaning task by human support robot using deep learning technique. *Sensors* **20**, 5. <https://doi.org/10.3390/s20061698> (2020).
3. Murdan, A. P. & Ramkissoon, P. K. A smart autonomous floor cleaner with an android-based controller. In *2020 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*, 235–239, <https://doi.org/10.1109/ELECOM49001.2020.9297006> (2020).
4. Deepa, G. *et al.* Robovac-automatic floor cleaning robot. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, 1–4, <https://doi.org/10.1109/ICIRCA51532.2021.9544914> (2021).
5. Ramalingam, B. *et al.* Deep learning based pavement inspection using self-reconfigurable robot. *Sensors* **21**, 5. <https://doi.org/10.3390/s21082595> (2021).
6. Ramalingam, B., Veerajagadheswar, P., Ilyas, M., Elara, M. R. & Manimuthu, A. Vision-Based Dirt Detection and Adaptive Tiling Scheme for Selective Area Coverage. *Journal of Sensors*. <https://doi.org/10.1155/2018/3035128> (2018).
7. Ramalingam, B., Lakshmanan, A.K., Ilyas, M., Le, A.V. & Elara, M.R. Cascaded machine-learning technique for debris classification in floor-cleaning robot application (2018).
8. Shin, M., Paik, W., Kim, B. & Hwang, S. An iot platform with monitoring robot applying cnn-based context-aware learning. *Sensors* **19**, 2525. <https://doi.org/10.3390/s19112525> (2019).
9. Koo, J. H., Cho, S. W., Baek, N. R., Kim, M. C. & Park, K. R. Cnn-based multimodal human recognition in surveillance environments. *Sensors* **18**, 3040. <https://doi.org/10.3390/s18093040> (2018).
10. Andersen, N.A., Braithwaite, I.D., Blanke, M. & Sorensen, T. Combining a novel computer vision sensor with a cleaning robot to achieve autonomous pig house cleaning. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on* 8331–8336 (IEEE, 2005).
11. Bormann, R., Fischer, J., Arbeiter, G., Weisshardt, F. & Verl, A. A visual dirt detection system for mobile service robots. In *ROBOTIK 2012; 7th German Conference on Robotics* 1–6 (2012).
12. Bormann, R., Weisshardt, F., Arbeiter, G. & Fischer, J. Autonomous dirt detection for cleaning in office environments. In *2013 IEEE International Conference on Robotics and Automation* 1260–1267 (2013).
13. Milinda, H.G.T. & Madhusanka, B.G.D.A. Mud and dirt separation method for floor cleaning robot. In *2017 Moratuwa Engineering Research Conference (MERCCon)* 316–320 (2017).
14. Lee, H. & Banerjee, A. Intelligent scheduling and motion control for household vacuum cleaning robot system using simulation based optimization. In *2015 Winter Simulation Conference (WSC)* 1163–1171 (2015).

15. Grünauer, A., Halmetschlager-Funek, G., Prankl, J. & Vincze, M. The power of gmms: Unsupervised dirt spot detection for industrial floor cleaning robots. In (eds. Gao, Y. *et al.*) *Towards Autonomous Robotic Systems* 436–449 (Springer International Publishing, 2017).
16. Martínez, D., Alenyà, G. & Torras, C. Planning robot manipulation to clean planar surfaces. In *Engineering Applications of Artificial Intelligence* **39** 23–32. <https://doi.org/10.1016/j.engappai.2014.11.004> (2015).
17. Park, J., Chen, J., Cho, Y. K., Kang, D. Y. & Son, B. J. Cnn-based person detection using infrared images for night-time intrusion warning systems. *Sensors* **20**, 34. <https://doi.org/10.3390/s20010034> (2020).
18. Goma, A., Abdelwahab, M. M., Abo-Zahhad, M., Minematsu, T. & Taniguchi, R.-I. Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow. *Sensors* **19**, 4588. <https://doi.org/10.3390/s19204588> (2019).
19. Fulton, M., Hong, J., Islam, M. J. & Sattar, J. In *Robotic Detection of Marine Litter Using Deep Visual Detection Models*. [arXiv:1804.01079](https://arxiv.org/abs/1804.01079) (2018).
20. Rad, M.S. *et al.* A computer vision system to localize and classify wastes on the streets. In *ICVS* (2017).
21. Zhihong, C., Hebin, Z., Yanbo, W., Binyan, L. & Yu, L. A vision-based robotic grasping system using deep learning for garbage sorting. In *Control Conference (CCC), 2017 36th Chinese* 11223–11226 (IEEE, 2017).
22. Cauli, N. *et al.* Autonomous table-cleaning from kinesthetic demonstrations using deep learning. In *2018 Joint IEEE 8th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* 26–32 (2018).
23. Farooq, A., Farooq, F. & Le, A. V. Human action recognition via depth maps body parts of action. *TIIS* **12**, 2327–2347 (2018).
24. Lakshmanan, A. K. *et al.* Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot. *Autom. Constr.* **112**, 103078 (2020).
25. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271. <https://doi.org/10.1007/BF01386390> (1959).
26. Hart, P. E., Nilsson, N. J. & Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **4**, 100–107. <https://doi.org/10.1109/TSSC.1968.300136> (1968).
27. Stentz, A. Optimal and efficient path planning for partially-known environments. In *Proceedings of the International Conference on Robotics and Automation* 3310–3317. (1993).
28. Koenig, S. & Likhachev, M. Fast replanning for navigation in unknown terrain. *IEEE Trans. Robot.* **21**, 354–363. <https://doi.org/10.1109/TRO.2004.838026> (2005).
29. Nazarahari, M., Khanmirza, E. & Doostie, S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Syst. Appl.* **115**, 106–120. <https://doi.org/10.1016/j.eswa.2018.08.008> (2019).
30. Cheng, K. P., Mohan, R. E., Nhan, N. H. K. & Le, A. V. Multi-objective genetic algorithm-based autonomous path planning for hinged-tetro reconfigurable tiling robot. *IEEE Access* **8**, 121267–121284. <https://doi.org/10.1109/ACCESS.2020.3006579> (2020).
31. Patle, B. K., Parhi, D. R. K., Jagadeesh, A. & KumarKashyap, S. Matrix-binary codes based genetic algorithm for path planning of mobile robot. *Comput. Electr. Eng.* **67**, 708–728. <https://doi.org/10.1016/j.compeleceng.2017.12.011> (2018).
32. Thabit, S. & Mohades, A. Multi-robot path planning based on multi-objective particle swarm optimization. *IEEE Access* **7**, 2138–2147. <https://doi.org/10.1109/ACCESS.2018.2886245> (2018).
33. Das, P. K., Behera, H. S. & Panigrahi, B. K. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm Evol. Comput.* **28**, 14–28. <https://doi.org/10.1016/j.swevo.2015.10.011> (2016).
34. PortaGarcia, M., OscarMontiel, Castillo, O., Sepúlveda, R. & Melin, P. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Appl. Soft Comput.* **9**, 1102–1110. <https://doi.org/10.1016/j.asoc.2009.02.014> (2009).
35. Zeng, M.-R., Xi, L. & Xiao, A.-M. The free step length ant colony algorithm in mobile robot path planning. *Adv. Robot.* **30**, 1509–1514. <https://doi.org/10.1080/01691864.2016.1240627> (2016).
36. Clark, M. C. Probabilistic road map sampling strategies for multi-robot motion planning. *Robot. Auton. Syst.* **53**, 244–264. <https://doi.org/10.1016/j.robot.2005.09.002> (2005).
37. Niewola, A. & Podsedkowski, L. L* algorithm-a linear computational complexity graph searching algorithm for path planning. *J. Intell. Robot. Syst.* **91**, 425–444. <https://doi.org/10.1007/s10846-017-0748-6> (2018).
38. Karaman, S. & Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **30**, 846–894. <https://doi.org/10.1177/0278364911406761> (2011).
39. Qureshi, A. H. & Ayaz, Y. Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments. *Robot. Auton. Syst.* **68**, 1–11. <https://doi.org/10.1016/j.robot.2015.02.007> (2015).
40. Li, Y., Chen, H., Er, M. J. & Wang, X. Coverage path planning for uavs based on enhanced exact cellular decomposition method. *Mechatronics* **21**, 876–885. <https://doi.org/10.1016/j.mechatronics.2010.10.009> (2011).
41. Candeloro, M., Lekkas, A. M. & Sørensen, J. A. A voronoi-diagram-based dynamic path-planning system for underactuated marine vessels. *Control Eng. Pract.* **61**, 41–54. <https://doi.org/10.1016/j.conengprac.2017.01.007> (2017).
42. Bewley, A., Ge, Z., Ott, L., Ramos, F. & Uppcroft, B. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)* 3464–3468 (2016).
43. Tieleman, T. & Hinton, G. *Lecture 6.5-rmsprop, coursera: Neural networks for machine learning*. University of Toronto, Technical Report (2012).
44. Le, A. V., Jung, S.-W. & Won, C. S. Directional joint bilateral filter for depth images. *Sensors* **14**, 11362–11378 (2014).
45. Bradski, G. The OpenCV Library. In *Dr. Dobbs Journal of Software Tools* (2000).
46. Le, A. V. & Choi, J. Robust tracking occluded human in group by perception sensors network system. *J. Intell. Robot. Syst.* **90**, 349–361 (2018).
47. Malandraki, C. & Dial, R. B. A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *Eur. J. Oper. Res.* **90**, 45–55 (1996).
48. Dorigo, M. & Di Caro, G. Ant colony optimization: A new meta-heuristic. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 2, 1470–1477 (IEEE, 1999).
49. Le, A. V., Nhan, N. H. K. & Mohan, R. E. Evolutionary algorithm-based complete coverage path planning for tetramond tiling robots. *Sensors* **20**, 445 (2020).
50. Targ, S., Almeida, D. & Lyman, K. *Resnet in resnet: Generalizing residual architectures*. [arXiv:1603.08029](https://arxiv.org/abs/1603.08029) (2016).
51. Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. *Inception-v4, inception-resnet and the impact of residual connections on learning*. [arXiv:1602.07261](https://arxiv.org/abs/1602.07261) (2016).
52. Quigley, M. *et al.* Ros: An open-source robot operating system. In *ICRA workshop on open source software*, vol. 3, 5 (Kobe, Japan, 2009).
53. González, J. *et al.* Mobile robot localization based on ultra-wide-band ranging: A particle filter approach. *Robot. Auton. Syst.* **57**, 496–507 (2009).

Author contributions

Conceptualization, R.E.M.; Methodology, R.E.M, Z.L.; Software, B.R, A.L., S.P., Z.W; Validation, R.E.M, Z.L.; Formal analysis, R.E.M, Z.L, B.R., S.P.; Investigation, B.R., A.L.; Resources, M.R.E.; Data curation, S.P., Z.W, Z.L; Writing---original draft, B.R., A.L , Z.W; Writing---review & editing, B.R., Z.W, Z.L; Visualization, S.P;

Supervision, M.R.E, Z.L; Project administration, M.R.E.; Funding acquisition, M.R.E. All authors have read and agreed to the published version of the manuscript.

Funding

This research is supported by the National Robotics Programme under its Robotics Enabling Capabilities and Technologies (Funding Agency Project No. 192 25 00051), National Robotics Programme under its Robot Domain Specific (Funding Agency Project No. 192 22 00058), National Robotics Programme under its Robotics Domain Specific (Funding Agency Project No. 192 22 00108), and administered by the Agency for Science, Technology and Research.

Competing interests

The authors declare no competing interests

Additional information

Correspondence and requests for materials should be addressed to B.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022