2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2022)
October 23-27, 2022, Kyoto, Japan

A Deep-Learning-based System for Indoor Active Cleaning

Yike Yun†, Linjie Hou†, Zijian Feng†,
Wei Jin, Yang Liu, Heng Wang, Ruonan He, Weitao Guo, Bo Han, Baoxing Qin, Jiaxin Li*,

***Abstract*— Cleaning public areas like commercial complexes is challenging due to their sophisticated surroundings and the vast kinds of real-life dirt. Robots are required to distinguish dirts and apply corresponding cleaning strategies. In this work, we proposed an active-cleaning framework by utilizing deep-learning methods for both solid wastes detection and liquid stains segmentation. Our system consists of 4 components: a Perception module integrated with deep-learning models, a Post-processing module for projection, a Tracking module for map localization, and a Planning and Control module for cleaning strategies. Compared with classic approaches, our vision-based system significantly improves cleaning efficiency. Besides, we released the largest real-world indoor hybrid dirt cleaning dataset (HD10K) containing 10K labeled images, together with a track-level evaluation metric for better cleaning performance measurement. The proposed deep-learning based system is verified with extensive experiments on our dataset, and deployed to Gaussian Robotics's robots operating globally. Dataset is available at: https://gaussianopensource.github.io/projects/active_cleaning.**

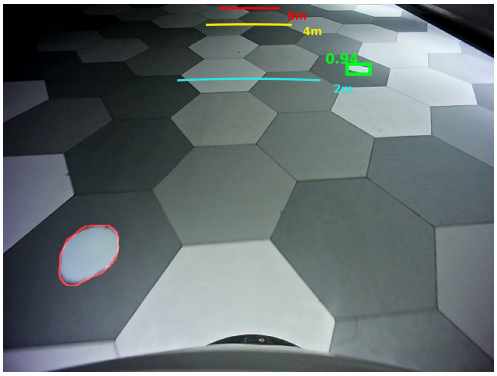

Fig. 1: Visualization of inferencing results from our active-cleaning system in production. Solid wastes (green box) and liquid stains (red contour) are detected via YOLOv5 [14] and DDRNet [28] backbones respectively.

I. INTRODUCTION

The demand for floor-cleaning robots has been booming in recent years, and many camera-based robots have been developed for better dirt detection. One of the main interests is active-cleaning, which requires the robot to actively identify dirt using vision system and correspondingly generate strategies to clean dirty areas. Compared with traditional cleaning approaches like S-path or wall-following [24], active-cleaning speeds up the overall procedure significantly.

Existing systems designed for active-cleaning like office cleaning robots [8] or outdoor garbage collecting robots [19] can be categorized into 3 stages: Detection, Localization, and Control. For detection, early approaches [5], [6], [8] utilized depth information with spectral residual filtering or GMM [12] for dirt & background separation. In recent years, more and more studies [7], [11], [16], [19] have started to employ deep-learning methods as their detection backbones, yielding more accurate detection results. For localization, a homography matrix is commonly applied to project the identified dirt from the image plane to the world frame. Followed by tracking and corresponding control strategies, the robot is capable of active-cleaning.

In the framework mentioned above, the detection module plays a vital role. However, most solutions only target solid wastes while the remaining few solutions [9], [29] for hybrid dirt solely apply object detection. Liquid stains usually occupy a much smaller area than the bounding box resulting a waste of cleaning effort after projection. Moreover, training a dirt detection model is challenging due to insufficient data in the field. Most of the work [2], [7] heavily use synthesized and augmented data. These generated data ignore spatial relationships between objects, which leads to poor performance for items at a far distance or in small size.

To improve cleaning efficiency, we build our system by framing the hybrid dirt cleaning task into detection and segmentation for solid wastes and liquid stains, respectively. To our best knowledge, we are the first to publish a deep-learning-based active-cleaning system with proven performance, and it has been deployed onto our robots worldwide. In addition, to alleviate the scarcity of data, we release the largest hybrid dirt dataset in the field, containing 10K images of solid wastes and liquid stains obtained from 3 cities. Aside from classic image-level precision and recall, we propose a track-level metric for a more comprehensive active-cleaning performance evaluation on our dataset. To sum up, our contributions are:

- We are the first to propose an indoor active-cleaning framework powered by deep-learning methods, which effectively detect and segment solid wastes and liquid stains.
- We release a Hybrid-Dirt-10K (HD10K) dataset with 10K images covering 3 real-world scenes. It is also the largest dataset so far in the field for floor-cleaning tasks.
- Apart from image-level evaluation, we present a track-level evaluation metric for solid wastes and liquid stains

†Equal contribution.
*Corresponding author.
All authors are from Gaussian Robotics Pte. Ltd., and correspondingly, emails are {assassin, houlinjie, fengzijian, carsonlee}@gs-robot.com.

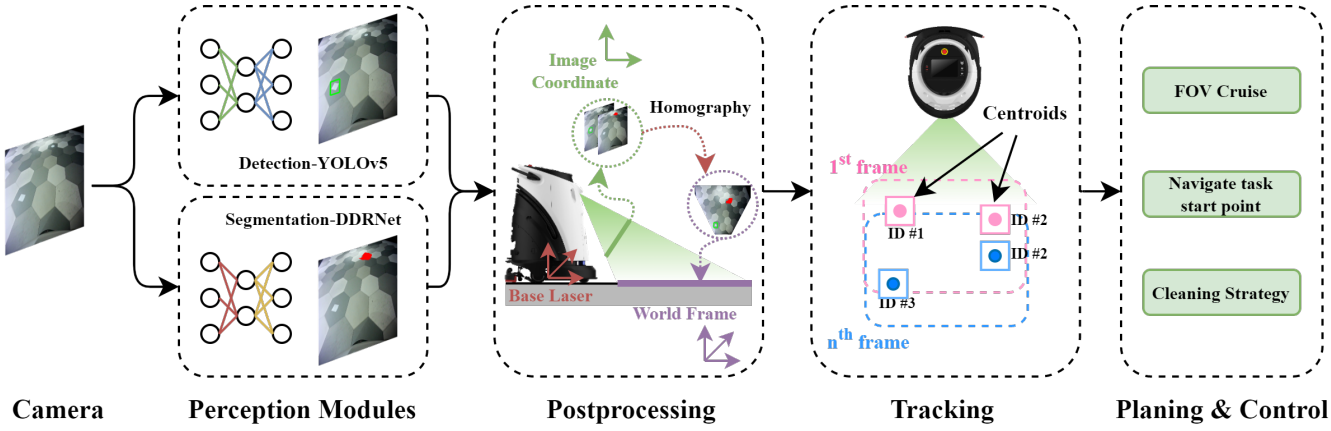

Fig. 2: Architecture for our active-cleaning framework. YOLOv5 [14] and DDRNet [28] are used for solid wastes detection and liquid stains segmentation, respectively. Inference results are further fed into post-processing module for image-world coordinate transformation and dirt size estimation. A tracking algorithm is developed for object re-identification and together with our Planning & Control(PNC) module, our robot is able to perform hybrid dirt active-cleaning in real world environment.

recognition.

II. RELATED WORK

Early solutions first try to separate dirts from the background, the detected dirts are then projected from image plane to ground surface, after which dirts are cleaned with different control strategies. For detection, 2D saliency-based methods [5], [8] using canny edge detector [10] and GMM [12] are adopted to filter out targets in the image. The identified dirts are then projected and transformed from the image plane to the ground surface using homography transformations. Followed by tracking algorithms like Adaptive Color Matching [22] or Kalman filters [15], detected objects can be traced and corresponding control options will be applied. Though such approaches are learning-free, they usually have a high false-positive rate, resulting in frequent human-machine interactions.

Powered by Convolutional Neural Networks (CNNs), deep-learning methods have greatly improved the efficiency of cleaning robots. In [29], Yin et al. developed their system using a VGG-alike [23] network as the detection module for table liquid stains detection and classification. After dirt are spotted, perspective transformation is applied for projecting the point of bounding boxes in the image plane to 3D points in the world frame. Then a grouping algorithm is applied for more efficient cleaning. Based on the spatial distribution of the stains, corresponding cleaning strategies will be executed based on the grouping results. In addition, [2] propose their active-cleaning system with a SegNet [1] model for coarse ground segmentation and a ResNet [13] model for solid wastes classification. Instead of using projection, they track the bounding boxes in the image plane and estimate their location in the real world.

III. SYSTEM ARCHITECTURE

The overall architecture of the proposed framework is illustrated in Fig. 2. We divide our system into 4 stages: Perception module, Post-processing, Tracking, and Planning & Control (PNC). The perception module receives camera input and generate inference results. In order to accurately locate liquid stains, we adopt a segmentation network instead of object detection. Followed by a post-processing module, we project the detected dirt from image plane to the ground plane. Then the tracking module tracks the projected dirt in each frame to continuously locate and update target's position. Lastly, a PNC module is applied for robot navigation and cleaning strategy planning. A cleaning task is marked as done once the defined working area is covered according to the FOV of the robot's camera.

IV. SYSTEM MODULES

A. Detection and Segmentation

1) Modified YOLOv5 for Detection: We employ YOLOv5 [14] for solid wastes detection. Due to the installation angle of the camera, as shown in Fig. 7, most of the solid wastes at distant, e.g., $> 2m$, are squeezed into a few pixels in the image, which are prone to false positives (FP) and false negatives (FN). FPs will trigger our robot to move to a non-existing target while FNs will make our robot short-sighted. In YOLO, detecting small objects relies on shallow features extracted by $F1-F3$ while the detection of big objects are based on semantic features in $F4-5$. However, as shown in Fig.3, we noticed that layer $B3$ also contains semantic features from $B4$ and $B5$, which might affect the performance of $P3$. We further visualize the feature maps of $B3$ by plotting the weights of convolution layers and found out only some of the features passed to layer $P3$ are useful, as illustrated in Fig.4. Thus we adopted a channel-wise spatial attention block [27] to guide $P3$ and integrated it at the bottom of PAN [20] to make $P3$ focus more on important features rather than treat them equally. The modified network achieves better results compared to the original YOLOv5.

2) DDRNet for Segmentation: As shown in Fig.5, DDRNet is used for liquid stains segmentation due to its superior performance on both accuracy and inference speed. Segmentation is preferred instead of object detection because

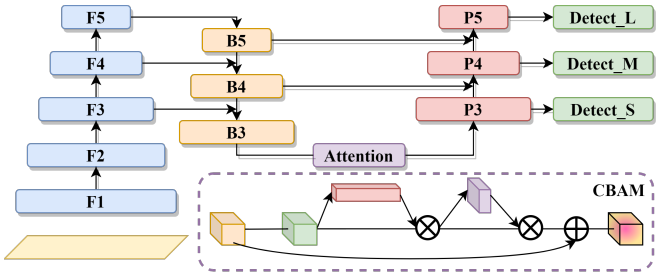

Fig. 3: Illustration of modified YOLOv5 [14] for solid wastes detection in our proposed system. Spatial and channel-wise attention are applied at the bottom of the PAN [20] module for better small object detection.

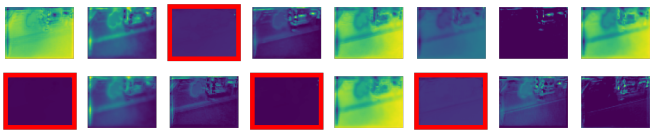

Fig. 4: Feature maps passed into $P3$ block for small object detection. Empty feature maps are highlighted by red boxes.

liquid stain is usually in irregular shape and size, i.e., the actual stain can be a small portion of its outer bounding box. Such effect will be amplified after projection onto the ground plane, and results in significant waste of cleaning effort.

Similar to [2], [7], we apply data augmentation at training. For detection, we copy-paste the solid wastes to random positions with random flipping and rotation, and we mix liquid stains images as negative samples. For segmentation, solid wastes images are also used as negative samples for better generalization.

B. Post-processing

For robot planning and control, detected objects should be transformed from the image plane to the ground plane with homography transformation. Unlike [8], which uses an estimated homography matrix, we calibrate our camera for more accurate transformation. The calibration setup is shown in Fig.6 where aruco QR codes are pasted on the box perpendicular to the ground. Through the camera, we can get the coordinates of intersections between the QR code, ground, and the box, denoted as C_{1-4}. Similarly, we can also obtain their coordinates through lidar, denoted as L_{1-4}. Finally, by using the PNP algorithm [18], we can calculate the homography matrix using C_{1-4} and L_{1-4}. Formally, given the inference results in the image coordinate $\boldsymbol{\mathcal{X}}_{image}$ and a homography matrix $\boldsymbol{\mathcal{H}}^{lidar}_{image}$ from plane *image* to *lidar*, the projected coordinate is given by:

$$\boldsymbol{\mathcal{X}}_{lidar} = \boldsymbol{\mathcal{H}}^{lidar}_{image} \boldsymbol{\mathcal{X}}_{image} \tag{1}$$

and the position in the world coordinate can be easily calculated based on the transformation between baselink and world frame.

C. Tracking

We denote a detected object as $\boldsymbol{\mathcal{O}}$, $\boldsymbol{\mathcal{O}}$ is a tuple consisting of the object center (c_x, c_y), the object id ID, its occurrences

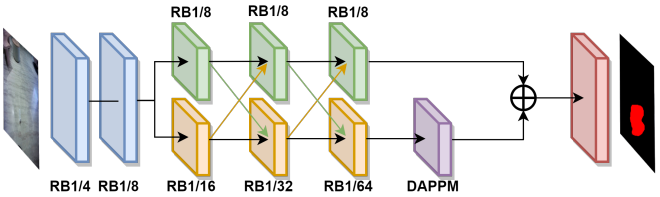

Fig. 5: DDRNet [28] for liquid stains segmentation in our proposed framework.

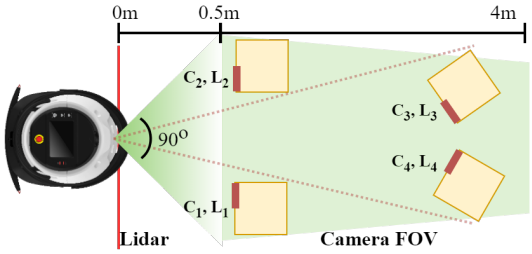

Fig. 6: Camera calibration. Calibration boxes (yellow) with QR code (red) are used for precise calculation of homography matrix from image plane to world frame. Distances shown in graph is not in scale.

in the last consecutive k frames N, and its life span T. Each detected object is initialized as $\boldsymbol{\mathcal{O}} = (c_x, c_y, 0, 0, T)$.

At timestamp t, for each observation, we compute the Euclidean distances between its center (c_x, c_y) and the centers of all existing objects in a list L. For an observation o_t, we denote the closest object to it in L as $\boldsymbol{\mathcal{O}}_{closest}$ and distance between o_t and $\boldsymbol{\mathcal{O}}_{closest}$ as d. If d is less than a threshold, we associate o_t with $\boldsymbol{\mathcal{O}}_{closest}$ - we update N_{cloest} accordingly and reset T_{cloest}. Otherwise, we initialize a new object $\boldsymbol{\mathcal{O}}_{new}$ and add it to L. For each object with no newly associated observation, we will decrease its life span by 1.

An object is marked as *Confirmed* if its occurrences N is larger than a pre-defined threshold δ, and is deleted from $\boldsymbol{\mathcal{L}}$ once its life span is decreased to 0. The confirmed objects are sent to the next module for downstream processing.

D. Planning and Control

Once assigned an area on the map, the robot will mark the region that has been covered by its camera's FOV and navigate itself to the unexplored areas. During the process, if dirts are detected and confirmed, the sizes of their projection on the ground are calculated and different cleaning strategies are applied accordingly: small dirts are cleaned by directly running over their object centers, while dirts larger than the robot's brush size are cleaned in S-path. The aforementioned process repeats until the whole target area is covered. Since camera's FOV is much larger than the size of our robot, compared to traditional S-path cleaning, our strategy can cover the area much faster, thus greatly boost the cleaning efficiency.

V. DATASET

In this section, we describe the details on data collection and annotation, and evaluation protocols of our dataset.

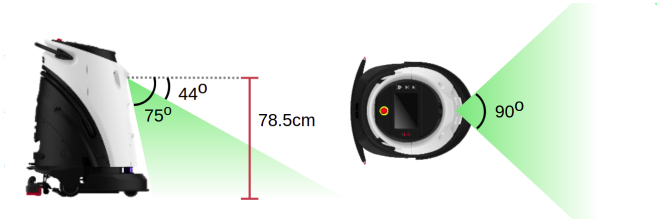

Fig. 7: Robot specifications. A RGB camera is installed in front of the robot at height of 78.5cm with an angle of 44° looking downwards.

Fig. 8: Examples of different types of solid wastes and liquid stains in our dataset.

A. Collection & Annotation.

We use our robot's front RGB camera to record the data. The camera has 90° of horizontal FOV and is located at 78.5cm in height with 44° offset downwards, as shown in Fig.7. Common dirt in shopping malls such as flyers/tickets, and coffee/tea stains are targeted as solid wastes and liquid stains correspondingly, as shown in Fig.8. During collection, dirt are randomly placed on the floor mimicking the real-life scenes and the robot was manually pushed using random routines. To ensure the variety of our data, 3 cities are selected, resulting in 3 different floor patterns (Fig.9). The collected data are in the form of video sequences in 10FPS; key-frames are extracted for annotation by thresholding frame similarity at 2Hz. Bounding boxes and segmentation masks are applied accordingly for solid wastes and liquid stains. Only key-frames are labeled for training data while the entire sequence is labeled for testing data.

B. Dataset statistics.

We build our dataset based on scenes, as shown in Table I. For training, 2 scenes are included each with 2000 extracted key-frames, while 2 video sequences with 1000 frames each are provided for testing. Besides, camera parameters are also included as they will be used in our proposed evaluation methods in Sec. V-C. For training data, solid wastes and liquid stains are separately collected and labeled, while for testing data, all dirt are collected and annotated together. In addition, test video 0 shares the same scene with scene 1, while test video 1 is an independent clip from all other 2 scenes in the training set.

Shown in Table II, our dataset is not only the largest in the field compared with other benchmark datasets [12], [17], [21], [26], but also covers the most comprehensive scenarios with both segmentation masks and bounding boxes presented. In total, our dataset consists of 10,000 images with 10,118 bounding boxes and 6,694 polygons for solid wastes and liquid stains, respectively.

Fig. 9: Examples of floor patterns and dirts in our HD10K Dataset. Unlike most synthesized data, our images comprise a good mixture of dirt in various size, and their perspective information is well preserved.

TABLE I: Number of images and labeled bboxes / polygons under each scene in HD10K Dataset.

Types	**Training**		**Testing**	
Solid wastes	**Scene 0**	**Scene 1**	**Scene 1**	**Scene 2**
Image	2,000	2,000	1,000	1,000
Annotations	2,625	3,691	1,883	1,919
Liquid stains	**Scene 0**	**Scene 1**	**Scene 1**	**Scene 2**
Image	2,000	2,000	1,000	1,000
Annotations	854	2,996	1,990	854
Mixed	✗	✗	✓	✓

TABLE II: Comparison between HD10K Dataset and other benchmark datasets in the field.

	HD10K Dataset	ACIN [12]	MJU [26]	Ext. TACO [21]	UAVVaste [17]
Size	**10,000**	969	2,475	4,562	772
Scenes	indoor	indoor	indoor	outdoor	outdoor
Resolution	480x640	480x640	480x640	various sizes	various sizes
Solid wastes	✓	✓	✓	✓	✓
Liquid stains	✓	✓	✗	✗	✗
Seg. masks	**6,694**	0	2,475	4,784	0
Bounding boxes	**10,118**	2,286	0	4,784	3,718

C. Sequence-based Evaluation Metrics

Current metrics like Precision, Recall and Mean IOU for object detection and semantic segmentation only evaluate each individual module, they do not reflect the overall performance of an end-to-end system. To this end, we propose a sequence-based end-to-end metric that jointly evaluates these modules and focuses on instance-level recognition performance. Specifically, we propose an MOT-alike metric to evaluate track-level precision and recall for solid wastes and liquid stains respectively. Given a set of bounding boxes and segmentation masks, we first generate a track ID for each solid / liquid stains, the track-level precision and recall can be computed based on the tracks of ground truth and predictions. To aid practical analysis, we also break down our metric based on the physical distance to the robot of each solid / liquid stains. We detail the metric calculation process for a single image sequence **s**:

1) Generating track IDs from ground truth annotations and model predictions: given the annotated bounding boxes and segmentation masks for all frames in the sequence, we use an offline process to generate the ground truth tracks to improve the quality of the track IDs. We first extract the liquid stains from the masks with *cv2.findContours* [25], the extracted polygons are then converted to bounding boxes by finding their minimum bounding rectangles. We run the SORT [3] algorithm on image coordinates for solid wastes and liquid stains respectively to generate the initial tracks

$\mathbf{T^s_{SORT}}$ and $\mathbf{T^l_{SORT}}$.

Assuming the locations of objects do not change over time, we perform hierarchical clustering on $\mathbf{T^s_{SORT}}$ and $\mathbf{T^l_{SORT}}$ respectively, where the distance d between a track $\mathbf{t_1}$ and another track $\mathbf{t_2}$ is defined as the mean Euclidean distance of all pairwise distances of the bounding box centres between the two tracks, formally:

$$d = \frac{1}{|\mathbf{P^1_2}|} \sum_{box1 \in \mathbf{t_1}} \sum_{box2 \in \mathbf{t_2}} \sum |c1 - c2| \tag{2}$$

where P^1_2 denotes all possible bounding box pairs between **t1** and **t2**, $c1$, $c2$ are the box centres of $box1$, $box2$ respectively.

We set the distance threshold for hierarchical clustering to 0.1 metres and denote the final ground truth tracks as $\mathbf{T^s_{gt}}$ and $\mathbf{T^l_{gt}}$ for solid wastes and liquid stains respectively. We perform the same process on the predicted bounding boxes and segmentation masks to generate the predicted tracks $\mathbf{T^s_{pred}}$ and $\mathbf{T^l_{pred}}$.

2) Track-level precision and recall: to match the predictions with ground truths, we first perform linear assignment between the ground truth boxes and predicted boxes and remove the matches whose IoU is less than 0.5. We keep track of the match status of each box in all the tracks in $\mathbf{T_{gt}}$ and $\mathbf{T_{pred}}$. A successful match is defined as more than half of the ground truth boxes are matched with predictions. A predicted track is marked as a true positive (TP) for a successful match. We perform this step for solid wastes and liquid tracks respectively. The track level precision and recall can then be computed as:

$$p^s = TP^s / |\mathbf{T^s_{pred}}|, r^s = M^s / |\mathbf{T^s_{gt}}| \tag{3}$$

$$p^l = TP^l / |\mathbf{T^l_{pred}}|, r^l = M^l / |\mathbf{T^l_{gt}}| \tag{4}$$

where TP^s and TP^l denote the number of true positive predicted solid wastes and liquid tracks, $|\mathbf{T^s_{pred}}|$ and $|\mathbf{T^l_{pred}}|$ denote the predicted tracks for solid wastes and liquid stains, M^s and M^l denote the number of matched ground truth solid / liquid tracks respectively.

3) Performance break-down based on physical distance to the robot: bad recognition results at different distances from the robot will have different impacts on the final system performance, for instance, if the pipeline fails to recognize liquid stains within 2 metres from the robot, the robot might run over of it instead of cleaning it, resulting in undesired consequences. Therefore, we break down our metrics into distance intervals for better performance evaluation. Specifically, for each ground truth object, we use homography transformation to project the lower centre of the bounding box to the ground plane to estimate its physical distance to the robot, then for each distance interval, we remove all out-of-bound objects before computing the metrics. We report the track-level recalls and precisions for distance intervals $[0,2)$, $[2,4)$ and $[4,\infty)$, all in metres.

The final metrics for distance interval $[d1,d2)$ are obtained by averaging the metrics of all sequences S:

$$P^s_{d1,d2} = \frac{1}{|\mathbf{S}|} \sum_{s \in \mathbf{S}} p^s_{d1,d2}, R^s_{d1,d2} = \frac{1}{|\mathbf{S}|} \sum_{s \in \mathbf{S}} r^s_{d1,d2} \tag{5}$$

$$P^l_{d1,d2} = \frac{1}{|\mathbf{S}|} \sum_{s \in \mathbf{S}} p^l_{d1,d2}, R^l_{d1,d2} = \frac{1}{|\mathbf{S}|} \sum_{s \in \mathbf{S}} r^l_{d1,d2} \tag{6}$$

VI. EXPERIMENTS

In this section, we briefly introduce our implementation details and report our evaluation results.

A. Implementation Details

We train the networks using our labeled HD10K Dataset. Data are augmented as mentioned in Sec.V-B. During training, images are firstly resized to 512×512, then mosaic augmentation [4] and random horizontal flipping are adopted. We train the network using a batch size of 16 with 100 epochs on a single RTX 2080Ti. We obtain our inferencing results using Intel OpenVINO platform and implement projection, tracking, and PNC algorithms in Robot Operating System (ROS).

B. Evaluation Results

Image-level Evaluation. As standard detection and segmentation evaluation practices, we report mAP (mean Average Precision) and mIOU (mean Intersection over Union) respectively on our HD10K dataset, as shown in Table III. For detection evaluation, we further compute precision & recall by setting a confidence threshold that achieves best F1 score. For segmentation, detection-like precision & recall can be evaluated similarly, by assigning confidence score as the mean pixel-wise confidence in the predicted stain mask, and computing mask-to-mask overlap with IoU.

Additionally, we present our ablation study in Table III, where *Raw, Blend, Aug.* represents training YOLOv5 on original dataset, dataset mixed with liquid stains, and data augmentation on solid wastes; while *Attn.* refers to modified yolo model in Sec. IV-A.1.

Track-level Evaluation Table IV shows the precision and recall of our system under different distance intervals. As mentioned before, objects at distant are small in the image, resulting in significant performance drop.

System-level Evaluation We further evaluate the end-to-end cleaning efficiency in real-world by comparing the proposed system with traditional S-path cleaning with the same speed and control configurations. Our proposed system boosts hourly cleaning efficiency from $500m^2$ to $1800m^2$.

TABLE III: Evaluation results of YOLOv5 and DDRNet backbones on the HD10K Dataset.

HD10K Dataset	Raw	Attn.	Blend	Aug.	P	R	mAP50	mAP95	mIOU
YOLOv5s	✓				0.606	0.617	0.538	0.420	-
		✓			0.622	0.632	0.543	0.442	-
		✓	✓		0.643	0.636	0.561	0.455	-
		✓	✓	✓	0.691	0.690	0.609	0.480	-
DDRNet	✓				0.701	0.693	-	-	0.847
			✓		0.726	0.718	-	-	0.858

TABLE IV: Recognition performance as per distance intervals.

Distance	Solid Wastes		Liquid Stains	
	Precision	Recall	Precision	Recall
0-2m	0.746	0.724	0.739	0.696
2-4m	0.607	0.524	0.701	0.735
4-6m	0.301	0.248	0.597	0.645

VII. CONCLUSIONS

We propose an high performance active-cleaning framework for indoor cleaning tasks, powered by deep-learning models for solid wastes detection and liquid stains segmentation. In addition, we release the HD10K dataset, which is the largest in the field so far. Furthermore, we propose comprehensive evaluation metrics for better cleaning efficiency measurement. Our proposed system can accurately detect hybrid dirt and effectively boost cleaning efficiency.

REFERENCES

[1] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *CoRR*, abs/1505.07293, 2015.

[2] Jinqiang Bai, Shiguo Lian, Zhaoxiang Liu, Kai Wang, and Dijun Liu. Deep learning based robot for automatically picking up garbage on the grass. *IEEE Trans. Consumer Electron.*, 64(3):382–389, 2018.

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

[5] Richard Bormann, Jan Fischer, Georg Arbeiter, Florian Weisshardt, and Alexander Verl. A visual dirt detection system for mobile service robots. In *ROBOTIK 2012; 7th German Conference on Robotics*, pages 1–6, 2012.

[6] Richard Bormann, Joshua Hampp, and Martin Hägele. New brooms sweep clean - an autonomous robotic cleaning assistant for professional office cleaning. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4470–4477, 2015.

[7] Richard Bormann, Xinjie Wang, Jiawen Xu, and Joel Schmidt. Dirtnet: Visual dirt detection for autonomous cleaning robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1977–1983, 2020.

[8] Richard Bormann, Florian Weisshardt, Georg Arbeiter, and Jan Fischer. Autonomous dirt detection for cleaning in office environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1260–1267, 2013.

[9] Daniel Canedo, Pedro Fonseca, Petia Georgieva, and António J. R. Neves. A deep learning-based dirt detection computer vision system for floor-cleaning robots with improved data collection. *Technologies*, 9(4), 2021.

[10] John F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[11] Michael Fulton, Jungseok Hong, Md Jahidul Islam, and Junaed Sattar. Robotic detection of marine litter using deep visual detection models. In *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, pages 5752–5758. IEEE, 2019.

[12] Andreas Grünauer, Georg Halmetschlager-Funek, Johann Prankl, and Markus Vincze. The power of gmms: Unsupervised dirt spot detection for industrial floor cleaning robots. In Yang Gao, Saber Fallah, Yaochu Jin, and Constantina Lekakou, editors, *Towards Autonomous Robotic Systems - 18th Annual Conference, TAROS 2017, Guildford, UK, July 19-21, 2017, Proceedings*, volume 10454 of *Lecture Notes in Computer Science*, pages 436–449. Springer, 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[14] Glenn Jocher, Alex Stoken, Ayush Chaurasia, Jirka Borovec, NanoCode012, TaoXie, Yonghye Kwon, Kalen Michael, Liu Changyu, Jiacong Fang, Abhiram V, Laughing, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Jebastin Nadar, imyhxy, Lorenzo Mammana, AlexWang1900, Cristi Fati, Diego Montes, Jan Hajek, Laurentiu Diaconu, Mai Thanh Minh, Marc, albinxavi, fatih, oleg, and wanghaoyang0106. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support, October 2021.

[15] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[16] Jaeseok Kim, Nino Cauli, Pedro Vicente, Bruno Damas, Filippo Cavallo, and José Santos-Victor. "icub, clean the table!" a robot learning from demonstration approach using deep neural networks. In *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 3–9, 2018.

[17] Marek Kraft, Mateusz Piechocki, Bartosz Ptak, and Krzysztof Walas. Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle. *Remote Sensing*, 13(5), 2021.

[18] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal Of Computer Vision*, 81:155–166, 2009.

[19] Jingyi Liu, Pietro Balatti, Kirsty Ellis, Denis Hadjivelichkov, Danail Stoyanov, Arash Ajoudani, and Dimitrios Kanoulas. Garbage collection and sorting with a mobile manipulator using deep learning and whole-body control. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pages 408–414, 2021.

[20] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8759–8768. Computer Vision Foundation / IEEE Computer Society, 2018.

[21] Pedro F. Proença and Pedro Simões. TACO: trash annotations in context for litter detection. *CoRR*, abs/2003.06975, 2020.

[22] Lin Rui, Du Zhijiang, and Sun Lining. Moving object tracking based on mobile robot vision. In *2009 International Conference on Mechatronics and Automation*, pages 3625–3630, 2009.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[24] Saurabh Singh, Vinay K Shetkar, Faisal Siddiqui, and Dawnee Soman. A novel hybrid navigation algorithm for autonomous robotic vacuum cleaners. In *2017 6th International Conference on Computer Applications In Electrical Engineering-Recent Advances (CERA)*, pages 491–496, 2017.

[25] Satoshi Suzuki and KeiichiA be. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[26] Tao Wang, Yuanzheng Cai, Lingyu Liang, and Dongyi Ye. A multilevel approach to waste object segmentation. *Sensors*, 20(14), 2020.

[27] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2018.

[28] Shi Yan, Chenglei Wu, Lizhen Wang, Feng Xu, Liang An, Kaiwen Guo, and Yebin Liu. Ddrnet: Depth map denoising and refinement for consumer depth cameras using cascaded cnns. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, volume 11214 of *Lecture Notes in Computer Science*, pages 155–171. Springer, 2018.

[29] Jia Yin, Koppaka Ganesh Sai Apuroop, Yokhesh Krishnasamy Tamilselvam, Rajesh Elara Mohan, Balakrishnan Ramalingam, and Anh Vu Le. Table cleaning task by human support robot using deep learning technique. *Sensors*, 20(6):1698, 2020.