

A Large-Scale Dataset for Benchmarking Elevator Button Segmentation and Character Recognition

Jianbang Liu¹, Yuqi Fang¹, Delong Zhu^{1*}, Nachuan Ma¹, Jin Pan¹, and Max Q.-H. Meng^{2*}

Abstract—Human activities are hugely restricted by COVID-19, recently. Robots that can conduct inter-floor navigation attract much public attention, since they can substitute human workers to conduct the service work. However, current robots either depend on human assistance or elevator retrofitting, and fully autonomous inter-floor navigation is still not available. As the very first step of inter-floor navigation, elevator button segmentation and recognition hold an important position. Therefore, we release the first large-scale publicly available elevator panel dataset in this work, containing 3,718 panel images with 35,100 button labels, to facilitate more powerful algorithms on autonomous elevator operation. Together with the dataset, a number of deep learning based implementations for button segmentation and recognition are also released to benchmark future methods in the community. The dataset is available at https://github.com/zhudelong/elevator_button_recognition

I. INTRODUCTION

Affected by the coronavirus, COVID-19, human activities are hugely restricted, e.g., people in those severe epidemic areas are forced to stay at home. The autonomous robot systems draw more public attention because they can substitute human workers and conduct many service work. Take the hospital as an example, if a mobile robot can move freely and safely across floors, it can replace the healthcare workers to deliver drugs to the infectious patients, greatly reducing the risk and burden on the medical personnel. However, fully autonomous inter-floor navigation of the service robot is still quite challenging nowadays.

To navigate the robot to travel across floors, traditional approaches either rely on assistance from the outside world or are simply based on the hand-craft features under specific environments [1]–[4], which make them not generally applicable. As the very first and fundamental step of autonomous inter-floor navigation, elevator button segmentation and recognition play an important role, which can enormously affect the performance of the robot system (see Fig. 1). However, there exists no large-scale elevator panel dataset for the researchers to verify their button segmentation

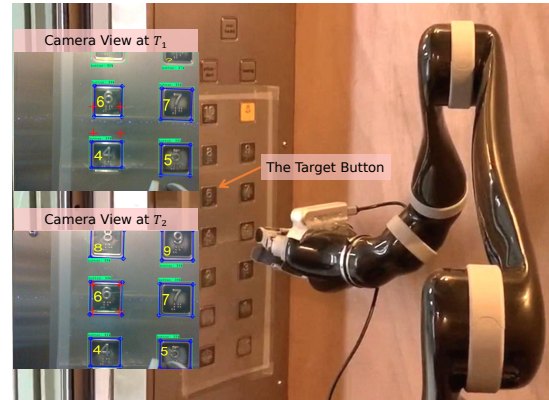


Fig. 1. Demonstration of elevator operation. The robot arm is tuning the pose of its end-effector and trying to press the target button.

and recognition approaches. As shown in Table I, existing elevator datasets [3], [5]–[7] either contain a few panels images or do not make data available. Therefore, we in this work release a large-scale elevator panel dataset to facilitate related studies on autonomous elevator operation. Moreover, benefiting from deep learning techniques that automatically mine informative features, a number of learning-based network implementations on button segmentation and recognition are also released to benchmark future methods in the community.

The contributions of this paper are summarized as follows:

- The first large-scale publicly available elevator panel dataset is released in this work, containing high-quality 3,718 panel images with 35,100 button labels.
- The baseline implementations on button segmentation and recognition, as well as evaluation metrics, are also released, for benchmarking future methods and facilitating related studies on autonomous elevator operation.

The remainder of this paper is organized as follows. Section II briefly surveys existing elevator panel datasets and recent studies on button operation. The details and characteristics of the released dataset are described in Section III. The evaluation metrics are established in Section IV, and network implementations for button segmentation and recognition in Section V. The paper is concluded in Section VI.

II. RELATED WORK

A. Elevator Panel Dataset

In the field of computer vision, some large-scale image datasets have been released to encourage the researchers to

*The corresponding author of this paper.

¹The authors are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China. email: {henryliu, fangyuqi, zhudelong, manachuan, jpan}@link.cuhk.edu.hk

²Max Q.-H. Meng is with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China, and also with the Shenzhen Research Institute of the Chinese University of Hong Kong, Shenzhen, China, on leave from the Department of Electronic Engineering of the Chinese University of Hong Kong, Hong Kong e-mail: max.meng@ieee.org. This project is partially supported by the Hong Kong RGC GRF grants #14200618 and Hong Kong ITC ITSP Tier 2 grant #ITS/105/18FP awarded to Max Q.-H. Meng.

TABLE I: The Comparison between our released elevator panel dataset and existing panel datasets. “-” denotes *not reported*.

Dataset	Dataset Statistic				Dataset Feature			
	Images	Buttons	panels	Classes	Video Sequence Image	GT Pose	Segmentation Map	public
Klingbeil <i>et al.</i> [3]	150	686	60	-	×	×	×	×
Zakaria <i>et al.</i> [5]	50	-	15+	-	×	×	×	×
Liu and Tian [6]	1,000	-	-	-	×	×	✓	×
Yang <i>et al.</i> [7]	260,560	-	8	26	×	×	×	×
Ours	3,718	35,100	2,100+	297	✓	✓	✓	✓






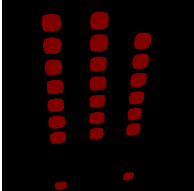
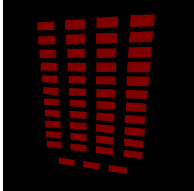

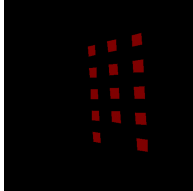
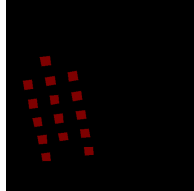




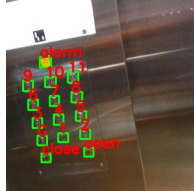
	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
Original					
Segmentation GT					
Recognition GT					

Fig. 2. Examples of the elevator panel samples with pixel-wise segmentation masks (row 2) and button recognition labels (row 3).

develop deep learning based solutions, such as ImageNet [8], KITTI [9], and PASCAL VOC [10]. As we know, with more sufficient training data, the generalization ability and robustness of a deep learning neural network can be improved. However, in autonomous elevator operation scenarios, there exists no public large-scale elevator panel dataset for button segmentation and recognition. For instance, Klingbeil *et al.* [3] collect only 150 panel images from more than 60 distinct elevators for button localization and optical character recognition (OCR). The training and testing sets are only 100 and 50 panel images, respectively. Zakaria *et al.* [5] evaluate their proposed button recognition and detection system using 50 high-resolution images including the internal and external elevator panels. Liu *et al.* [6] collect 1,000 panel images captured from both inside and outside of the elevators, to train the deep learning based method for pixel-wise button segmentation. The biggest panel dataset to date is collected by Yang *et al.* [7], which contains 260,560 images that are composed of 8 different panels, and many data augmentation strategies have been utilized to further enlarge the data scale, e.g., blurring, sharpening, and histogram equalization. However, Yang *et al.* [7] do not release their collected dataset, thus no comparison studies can be conducted. This paper releases the first public large-scale elevator panel dataset, which contains 3,718 panel images and 35,100 button labels, covering the majority of

existing button categories. Furthermore, our dataset contains both static images and video sequences, as well as pose information between the panels and the vision camera. We believe that this dataset can benchmark future methods on autonomous elevator operation.

B. Autonomous Elevator Operation

Research on elevator button segmentation and recognition has received more attention in recent years. The existing methods can be categorized into traditional methods and deep learning based methods. For traditional methods, as one of the earliest studies on autonomous elevator operation, Klingbeil *et al.* [3] perform button detection using a sliding-window detector first, and then apply the Expectation Maximization (EM) algorithm to remove false positives and infer missing buttons. In button recognition stage, the OCR model is adopted, and then a hidden markov model (HMM) is leveraged to take into consideration the arrangement of button labels in order of floors. Their method correctly detects and labels 86.2% of the buttons in a small-scale dataset with 50 test panels. Kim *et al.* [4] propose a robust vision-based button recognition method, which is comprised of feature extraction, initial button recognition, and post-processing modules. Moreover, considering that a button object has a convex quadrilateral boundary, the authors design the specific features to represent the button contour

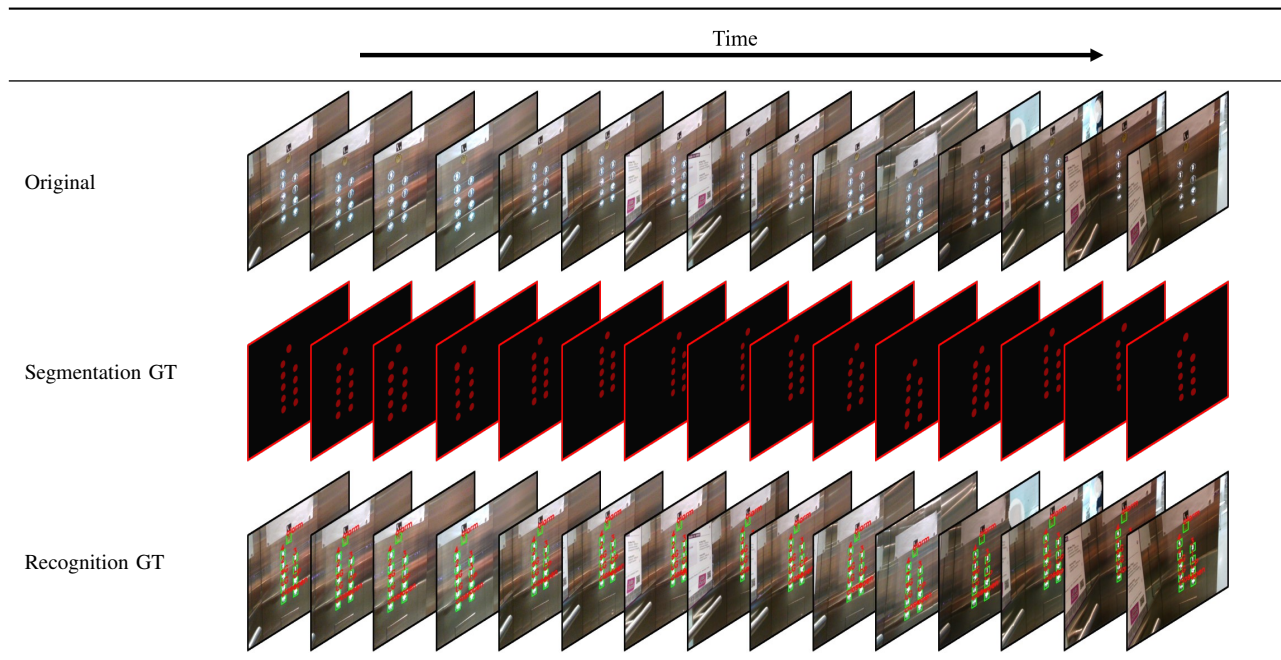


Fig. 3. The video sequence samples with ground-truth segmentation masks and button recognition labels in our dataset.

under different perspective distortions, which overcomes the difficulty caused by the reflective walls or the partial occlusion and greatly improves button segmentation accuracy. Zakaria *et al.* [5] develop an efficient button detection and recognition framework based on the sobel operator [11] for button edge detection and Wiener filter for reflection noise removal. These traditional algorithms highly rely on the domain knowledge and manual design, and most of them are only evaluated in some specific environments, hence their performance and robustness remain unsatisfying.

Benefiting from deep learning technology, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown superior performance over traditional methods in different applications. Researchers have applied deep learning techniques to solve the button detection and recognition problem. Liu *et al.* [6] adopt the fully convolutional network (FCN) for pixel-level segmentation of elevator buttons, and integrate the single-shot detector (SSD) with the convolutional recurrent neural network for button localization and recognition. Dong *et al.* [12] first propose the proper button candidates based on R-CNN [13], and then a fine-tuned CNN is developed for elevator button recognition, which achieves a reliable and promising recognition performance. Based on the system proposed in [12], Zhu *et al.* [14], [15] further modify the system by combining OCR network and Faster R-CNN network [16] into a single architecture, and thus button detection and recognition can be performed simultaneously, enabling an end-to-end training scheme. Yang *et al.* [7] utilize the YOLO v2 network [17] for elevator button detection, and this 2D YOLO detector can produce the 3D coordinates of the target button based on a coordinate transform neural network. In this work, along with the released dataset, we introduce many classic and popular

segmentation and recognition approaches, benchmarking and facilitating related studies on autonomous elevator operation.

III. ELEVATOR BUTTON DATASET

The released elevator panel dataset contains 3,718 panel images with 35,100 button labels. The images are comprised of two types: static images (Fig. 2) and video sequences (Fig. 3). The static images, which contains inner and outer control panels, are mostly collected from the Internet using a web crawler, and part of them are captured by the authors. We also manually clean the images to ensure there are no repetitive samples for the same elevator panel. The collected images are diverse in panel types, shooting angles, and lighting conditions. The video sequences are captured by a RealSense depth camera D435 mounted on the end effector of a Kinova robot arm (see Fig. 4). Specifically, we control the robot arm to move smoothly along some predefined trajectories, e.g., moving towards or away from the target panels while keeping them within the perception field. Images in the video sequences show slight blurs due to the movement of the robot arm, but this can increase the diversity of the training set, improving the generalization

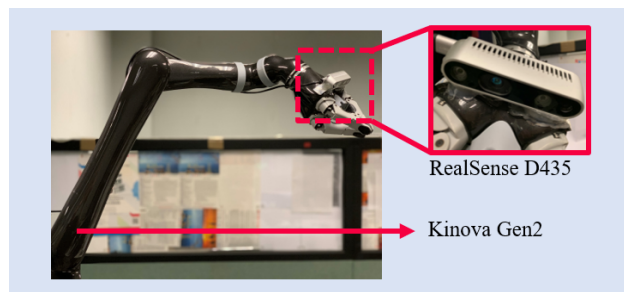


Fig. 4. The hardware system for video sequence acquisition.

ability and robustness of the models.

Furthermore, during image sequence acquisition, an ArUco marker [18] is stucked on each elevator panel, based on which the poses between the camera and the elevator panel can be performed with the help of OpenCV libraries. In [19], we utilize the pose information to assess the accuracy of the proposed distortion removal algorithm. The pose information is also released along with the elevator panel dataset, but in this work, it has not been exploited to assist button segmentation and recognition. To the best of our knowledge, this is the first dataset that integrates the pose information of elevator panels. We encourage future studies to leverage this prior knowledge to further improve the performance in autonomous button segmentation and recognition. The pose information also provides ground truth for single-image based pose estimation algorithms, which is the key for manipulation control.

A. Button Segmentation

The collected elevator buttons have various shapes and appearances, e.g., circle, ellipse, rectangle, and trapezium, under different camera views. All buttons are carefully annotated. Specifically, we first draw a contour exactly outlining each button, and then generate the mask accordingly. All the mask labels are checked by colleagues in this research field and ascertain the correctness. In Fig 2, the segmentation ground truth is represented by binary masks, i.e., the dark red areas. Since the button class is annotated simultaneously, multi-class segmentation techniques can also be leveraged to address the problem.

B. Button Recognition

The released dataset contains 297 classes of buttons, covering the majority of existing button categories, which are labeled carefully by the authors. The distribution of button samples in top-fifty classes is demonstrated in Fig. 5, indicating that there exists a severe class imbalance problem in the dataset. For example, some buttons (e.g., open, close, alarm), are commonly seen on most elevator panels, the number of which is unsurprisingly larger than buttons that are specifically designed, e.g., podium floor, rooftop, and ballrooms. Moreover, low-rise buildings are always more than high-rise ones, hence buttons with small numbers (e.g., floor 1, 2, 3) are significantly more than those with large numbers (e.g., floor 100).

The class imbalance problem makes button recognition models easily overfitted to classes with sufficient training samples. To tackle this problem, we formulate the recognition task as an OCR problem instead of a classification problem. An OCR framework is developed in [15], which can decompose the text on a button into a series of characters. For instance, floor 102 can be split into (1, 0, 2), and each character can be recognized individually. A character dictionary is specially designed for this dataset (see Table II), based on which the OCR framework can recognize hundreds of button categories without changing the network design.

TABLE II: The designed character dictionary for the dataset. The functional buttons are encoded by the special characters, e.g., >< for *close*, \$ for *alarm*, # for *stop*, and & for *call*.

'0': 0	'1': 1	'2': 2	'3': 3	'4': 4	'5': 5
'6': 6	'7': 7	'8': 8	'9': 9	'A': 10	'B': 11
'C': 12	'D': 13	'E': 14	'F': 15	'G': 16	'H': 17
'I': 18	'J': 19	'K': 20	'L': 21	'M': 22	'N': 23
'O': 24	'P': 25	'R': 26	'S': 27	'T': 28	'U': 29
'V': 30	'X': 31	'Z': 32	'<': 33	'>': 34	'(': 35
'\)': 36	'\$': 37	'#': 38	'&': 39	's': 40	'-': 41
'*': 42	'%': 43	'?': 44	'!': 45	'\phi': 46	

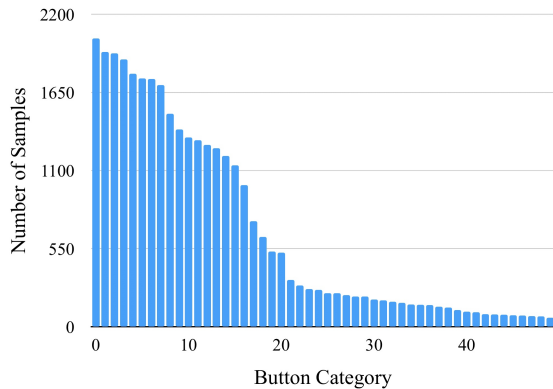


Fig. 5. The class distribution of top 50 button categories.

IV. BENCHMARK METRICS

In this section, several commonly used metrics are introduced to evaluate the performance of elevator button segmentation and recognition among different algorithms.

A. Metrics for Button Segmentation

Four measurement metrics are adopted to validate the button segmentation performance, i.e., *Intersection over Union*, *Precision*, *Recall*, and *Parameter Size*, each of which is detailed as follows.

Intersection over Union, also known as Jaccard Index, is the most popular metric for evaluation of the segmentation performance, which aims to measure the overlap between the predicted segmentation masks and the ground-truth areas. For a given input panel image, IoU is calculated as follows:

$$IoU = \frac{TP}{TP + FP + FN}, \quad (1)$$

where True Positive (TP) indicates the number of pixels that are correctly classified as button pixels; False Positive (FP) means the number of pixels that are wrongly classified as button pixels; False Negative (FN) denotes the number of pixels that are wrongly classified as the background.

Precision and Recall are also two common metrics for segmentation quality measurement. Precision is sensitive to the over-segmentation, while recall is sensitive to the under-segmentation, which are denoted as follows:

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}. \quad (2)$$

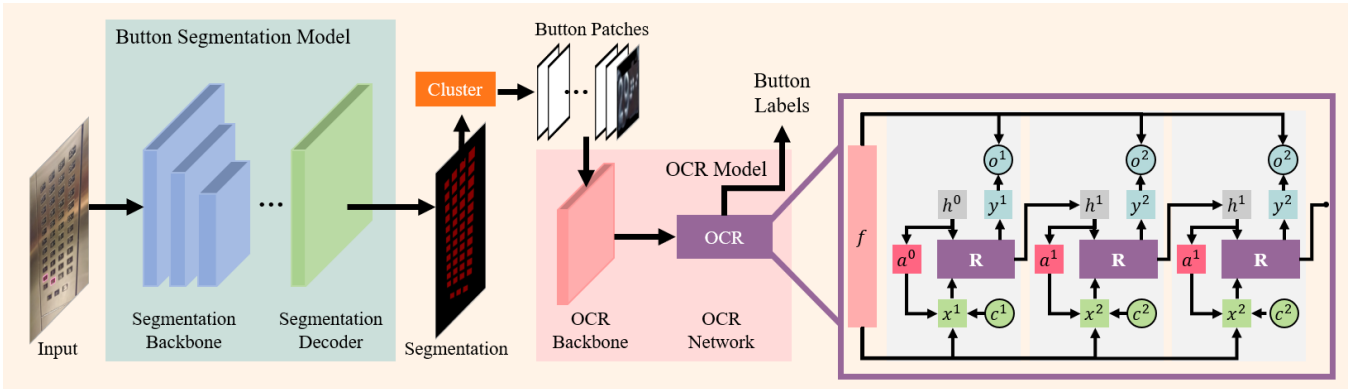


Fig. 6. The proposed button segmentation and recognition framework. The segmentation model takes a RGB panel image as the input, and generate the button segmentation mask. A clustering module (DBSCAN) aims to group pixels close to each other as an independent button, and frames the button with a tight bounding box. Based on each box, the OCR model implemented via an attention RNN network (right) recognizes its button character in sequence. The details of the OCR model can refer to [14].

Parameter Size, measuring the number of training parameters, is an extremely important metric in button segmentation task. Although computation memory is usually sufficient, it can be a limiting factor in some scenarios, especially those requiring the real-time manipulation. In these situations, models with smaller parameter size can be extraordinarily helpful, enabling more efficient elevator operation. However, it is possible that these light-weighted models produce less satisfactory performance compared with the heavy ones, hence a trade-off between the efficiency and accuracy needs to be made in practice.

B. Metrics for Button Recognition

For button recognition, *Average Accuracy (ACC)* is adopted to evaluate the overall recognition performance. *ACC* is the percentage of the correctly recognized buttons out of the total amount of buttons involved, which is:

$$ACC = \frac{\text{No. of button recognized correctly}}{\text{No. of button involved}}. \quad (3)$$

V. EXPERIMENTAL RESULTS

In this section, we detail the basic network implementations in terms of button segmentation and recognition, which benchmark future methods and facilitate related studies in this research field.

TABLE III: The button segmentation results based on various algorithms. Abbreviations: IoU: intersection over union; Params: parameter size; Res50, Mob2, Mob3, and Gh represent ResNet50, MobileNetV2, MobileNetV3, and GhostNet, respectively.

	IoU(%)	Precision(%)	Recall(%)	Params(M)
DeeplabV3+(Res50)	53.40	55.16	94.36	40.35
DeeplabV3+(Mob2)	34.78	35.68	93.25	5.82
DeeplabV3+(Mob3)	47.30	48.84	93.76	9.64
DeeplabV3+(Gh)	53.34	56.25	91.16	8.25
FCN32s	51.57	64.24	72.33	134.27
FCN16s	62.45	74.01	80.00	134.27
FCN8s	62.72	73.75	80.74	134.27
ICNet	46.93	55.94	74.44	7.73
ENet	42.36	43.25	95.39	0.35
U-Net	57.65	67.36	80.00	1.94
PSPNet	56.54	60.10	90.54	68.06

A. Button Segmentation

For elevator button segmentation, several popular semantic segmentation algorithms are adopted and implemented in this work. Ronneberger *et al.* [20] design a U-shape convolutional neural network (U-Net), in which the contracting path aims to capture low-level features in shallow layers and high-level features in deeper layers, and the expansive path maps back the extracted features to the same size of the input data to reconstruct the pixel-wise segmentation mask. Long *et al.* [21] propose an encoder-decoder network architecture (FCN) for segmentation tasks, which replaces the fully connected (FC) layers of CNNs with convolutional layers. Based on the basic FCN, several modified versions (FCN8s, FCN16s, FCN32s) are proposed to fuse predictions of decoder layers and further improve the segmentation accuracy. Chen *et al.* [22], [23] publish a series of papers introducing DeepLab and its improved versions. The last proposed DeepLabv3+ is still one of the state-of-the-art (SOTA) algorithms in semantic segmentation. Zhao *et al.* [24] develop an effective pyramid scene parsing network (PSPNet) that is embedded with a novel pyramid pooling module for semantic segmentation. With the pyramid pooling module, PSPNet can effectively exploit the contextual information of the given input, and achieves superior performance in ImageNet scene parsing challenge 2016, PASCAL VOC 2012 benchmark, and Cityscapes benchmark. Moreover, two computationally-efficient and light-weighted neural networks, i.e., ENet [25] and ICNet [26], are adopted in this work, which may perform semantic segmentation in a real-time manner.

All algorithms mentioned above are implemented. Particularly, for DeepLab series, we adopt the latest version (DeepLabv3+) embedded with multiple feature extraction backbones, i.e., ResNet50 [27], MobileNetV2 [28], MobileNetV3 [29], and GhostNet [30]. The reason why we choose these backbones is due to their compact designs. We randomly split the dataset into train, validation, and test set, following a ratio of 70%, 15%, and 15%. Some common augmentation strategies are adopted, i.e., random horizontal and vertical flipping, clockwise-rotation with an

TABLE IV: The button recognition results on the test set (558 images and 5,309 buttons) based on different segmentation algorithms described in Section V-A. “Button Detected”: the number of buttons correctly detected; “ACC (Detected)” and “ACC (All)” represent the percentage of correctly recognized buttons out of the detected buttons, and the whole test buttons, respectively.

Segmentation Model	Additive OCR Model			Multiplicative OCR Model		
	Button Detected	ACC (Detected)	ACC (All)	Button Detected	ACC (Detected)	ACC (All)
DeeplabV3+ (Res50)	3,296	73.74%	62.08%	3,251	72.73%	61.24%
DeeplabV3+ (Mob2)	1,983	69.58%	37.35%	1,957	68.67%	36.86%
DeeplabV3+ (Mob3)	2,781	72.69%	52.38%	2,705	70.70%	50.95%
DeeplabV3+ (Gh)	3,233	71.78%	60.90%	3,178	70.56%	59.86%
FCN32s	1,746	60.04%	32.89%	1,684	57.91%	31.72%
FCN16s	3,026	63.79%	57.00%	2,915	61.45%	54.91%
FCN8s	3,253	65.27%	61.27%	3,160	63.40%	59.52%
ICNet	2,116	68.66%	39.86%	2,054	66.65%	38.69%
ENet	2,999	70.83%	56.49%	2,941	69.46%	55.40%
U-Net	3,397	67.00%	63.98%	3,301	65.11%	62.18%
PSPNet	3,408	72.50%	64.19%	3,322	70.67%	62.57%

angle $[0^\circ, 90^\circ]$.

During model training, the focus loss [31] is adopted, which emphasizes more on hard and misclassified button examples. The loss function is represented as follows:

$$l(i, j) = -g(i, j)[1 - S(p(i, j))^\gamma] \log[S(p(i, j))], \quad (4)$$

where $p(i, j)$ indicates the predicted probability of pixel (i, j) being categorized as the button pixel; $\gamma \geq 0$ means the focusing parameter; $S(\cdot)$ is the soft-max function; $g(i, j) \in \{1, 0\}$ represents the ground-truth label. 1 means that (i, j) is the button pixel, while 0 means the background pixel.

The results of all semantic segmentation models are summarized in Table III. It can be seen that FCN8s, with the largest parameter size or computation complexity, achieves the highest IoU score. ENet contains almost 400 times smaller model size compared with FCN8s, but its IoU and Precision scores are quite low. PSPNet and DeeplabV3+ with ResNet50 or GhostNet backbone achieve a balance between precision and recall scores, indicating fewer efforts are required to remove false negatives and false positives of their segmentation maps. U-Net achieves a satisfactory button segmentation performance, and meanwhile, the number of trained parameters is small. Generally speaking, FCN8s and FCN16s are good solutions for button segmentation tasks concerning only the segmentation performance in the autonomous system embedded with enough computational resources, especially GPU memory. However, for systems that require real-time segmentation or have limited computation resources, U-Net is a better alternative.

In all segmentation experiments, the initial learning rate is $7e-4$. Each network is trained for 50 epochs, where the model with the highest IoU in evaluation set is selected for inference. The Adam optimizer is adopted with a weight decay of $5e-4$. The networks are implemented based on the PyTorch framework and trained on a workstation with Intel Core(TM) i7-10700K@3.80GHz processors and a NVIDIA GeForce RTX 2080 (8 GB) installed.

B. Button Recognition

Before conducting button recognition, the box exactly bounding each elevator button is generated based on the

button segmentation masks derived in Section V-A. Specifically, a powerful clustering method called density-based spatial clustering of applications with noise (DBSCAN) [32] is applied on the segmentation mask first, which aims to group pixels close to each other as an independent button. The maximum and minimum coordinates of the pixels of one button are then derived, which can form one bounding box. Ideally, each bounding box contains one button. Finally, an accurate and efficient OCR framework proposed in our early work [14] is applied to perform button character recognition, which is based on an attention-RNN (see Fig. 6).

Following different segmentation strategies, the button recognition results on test set are shown in Table IV. According to the different designs of attention RNN, we divide the OCR model into two categories, i.e., additive and multiplicative models. Froms Table IV, we observe that the OCR models with additive attention perform slightly better than those with multiplicative attention mechanism. While testing on the whole test set, it can be seen that a highest accuracy of 64.19% and 62.57% is achieved in two OCR models based on PSPNet segmentation. While testing on the correctly detected buttons, Deeplabv3+ with ResNet50 backbone has better recognition results, but the number of detected buttons it leverages is smaller than that of PSPNet. U-Net achieves almost the same performance as PSPNet, and meanwhile requires much less computation resources (see Table III), thus it can be regarded as a great candidate in elevator button recognition.

In all recognition experiments, the initial learning rate is set to $1e-3$ and decays by $0.1x$ every 17 epochs, with a total of 50 training epochs. The RMSprop optimizer is adopted with a weight decay of 0.1. The networks are implemented based on the PyTorch framework and trained on a workstation with Intel Core(TM) i7-5930K@3.50GHz processors and a NVIDIA GTX TITAN X (12 GB) installed.

VI. CONCLUSIONS

In this paper, the first large-scale publicly available elevator panel dataset is released, which contains 3,718 panel images with 35,100 button labels. This dataset aims to benchmark future methods on elevator button segmentation and recognition. Along with the dataset, several measurement

metrics are established for performance evaluation, and many popular network implementations for button segmentation and recognition are evaluated and released, which are believed to further push forward related studies in this research area and facilitate real-world autonomous elevator operation.

REFERENCES

- [1] J. Miura, K. Iwase, and Y. Shirai, "Interactive teaching of a mobile robot," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3378–3383.
- [2] J.-G. Kang, S.-Y. An, and S.-Y. Oh, "Navigation strategy for the service robot in the elevator environment," in *2007 International Conference on Control, Automation and Systems*. IEEE, 2007, pp. 1092–1097.
- [3] E. Klingbeil, B. Carpenter, O. Russakovsky, and A. Y. Ng, "Autonomous operation of novel elevators for robot navigation," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 751–758.
- [4] H.-H. Kim, D.-J. Kim, and K.-H. Park, "Robust elevator button recognition in the presence of partial occlusion and clutter by specular reflections," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1597–1611, 2011.
- [5] W. N. F. W. Zakaria, M. R. Daud, S. Razali, and M. F. Abas, "Elevators external button recognition and detection for vision-based system," *Proceeding of the Electrical Engineering Computer Science and Informatics*, vol. 1, no. 1, pp. 265–269, 2014.
- [6] J. Liu and Y. Tian, "Recognizing elevator buttons and labels for blind navigation," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2017, pp. 1236–1240.
- [7] P.-Y. Yang, T.-H. Chang, Y.-H. Chang, and B.-F. Wu, "Intelligent mobile robot controller design for hotel room service with deep learning arm-based elevator manipulator," in *2018 International Conference on System Science and Engineering (ICSSE)*. IEEE, 2018, pp. 1–6.
- [8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [9] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, June 2010.
- [11] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.
- [12] Z. Dong, D. Zhu, and M. Q.-H. Meng, "An autonomous elevator button recognition system based on convolutional neural networks," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 2533–2539.
- [13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [14] D. Zhu, T. Li, D. Ho, T. Zhou, and M. Q. Meng, "A novel ocr-cnn for elevator button recognition," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3626–3631.
- [15] D. Zhu, Y. Fang, Z. Min, D. Ho, and M. Q. H. Meng, "Ocr-cnn: An accurate and efficient framework for elevator button recognition," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2021.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [17] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.
- [18] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [19] D. Zhu, J. Liu, N. Ma, Z. Min, and M. Q.-H. Meng, "Autonomous removal of perspective distortion for robotic elevator button recognition," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 913–917.
- [20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [23] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.
- [24] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.
- [25] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.
- [26] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 405–420.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [29] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1314–1324.
- [30] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589.
- [31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [32] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.