An autonomous elevator button recognition system based on convolutional neural networks

Conference Paper · December 2017

DOI: 10.1109/ROBIO.2017.8324801

3 authors, including:

Delong Zhu
The Chinese University of Hong Kong
14 PUBLICATIONS **50** CITATIONS

Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

An Autonomous Elevator Button Recognition System Based on Convolutional Neural Networks

Zijian Dong, Delong Zhu and Max Q.-H. Meng
Department of Electronic Engineering, The Chinese University of Hong Kong,
Shatin, N.T., Hong Kong SAR, China
Email: {zjdong, dlzhu, max}@ee.cuhk.edu.hk

***Abstract*—The ability to operate an elevator is significant for service robots to move freely inside a building and the elevator button recognition is placed as one of the most critical functions of this process. However, the variety of button styles, the different light conditions and the blurred images caused by the camera motion make this task difficult. To tackle this obstacle to achieve the robust real-time performance, a button recognition system is proposed based on the convolutional neural networks. In consideration of the diverse button shapes, a contour extraction algorithm and the noise filtering are specifically designed to avoid the exhaustive search and reduce the consumed time. Then the fine-tuned CNN model is trained on our established elevator button dataset to achieve a more reliable recognition performance comparing to the template matching methods. Besides, the arrangement pattern of buttons is utilized to deduce the missing buttons and correct mistakes. To verify our algorithm, we run our algorithm on a dataset of 5 distinct elevators. Our algorithm succeeds in localizing and recognizing 98% of the buttons in known elevators and 87.6% in unknown elevators and has an average speed of 3 frames per second.**

***Index Terms*—elevator button recognition, convolutional neural network, contour detection**

I. Introduction

Although the robot navigation system continues to mature, service robots still cannot move freely between floors inside a building equipped with elevators. This ability to operate an elevator will be critically demanded for service robots in popular places such as hotels, hospitals, elderly care centers and office buildings. Current service robots depend more on the human assistance [1] or the elevator renovation to overcome the difficulty of operating an elevator. However, transforming all elevator button patterns or control modes in all buildings is impossible to achieve. Depending on human assistance also makes robots inefficient, since a human being must be stationed in each elevator to help the robots.

A more effective approach is to make robots autonomously operate the elevators through their perception. In this paper, we report our work on the development of algorithms based on deep methods to localize buttons, recognize their numbers and provide guidance for the robot arm to press the button. In our experiment, we limit our study on the interior button panel, because it is more complex and thus a more challenging research for the autonomous operation of elevators.

Fig. 1 displays some common elevator panels. The various button styles, different light conditions, reflective surfaces, and dynamically changing observing angles and distances from the cameras to the button panels form great challenge for the vision algorithm. In addition, because the algorithm is used by robots, the higher accuracy and the near real-time performance of the algorithm are needed, which is different from the traditional algorithm for the object detection. The accuracy of the algorithm needs to be close to 100% and the algorithm requires a fast response speed. Furthermore, due to the movement of the robots, problems such as the tilt, blur and deformation of the captured picture exist, which increases the difficulty of this task.

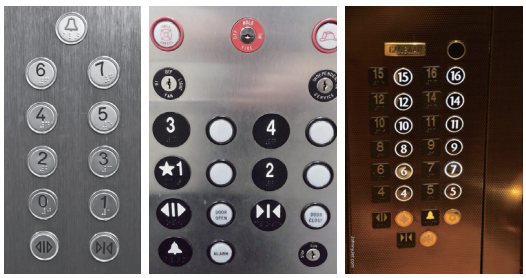


Fig. 1: Samples of elevator panels

To meet the requirement of high accuracy and near real-time control in the changing elevator environment mentioned above, we present a recognition system based on the convolutional neural networks (CNN), which combines the ROI extraction algorithms and the button arrangement rule. Due to the requirement for the algorithm speed and the special contour profile of the elevator buttons, we apply a preprocessing algorithm to obtain the contours of the button through edge detection, thereby providing the region of interest (ROI) for the recognition. Specifically, the size filter, the button feature classifier and the Non-Maximum Suppression (NMS) algorithm are applied to eliminate the false positive regions. In order to further improve the precision of the algorithm, we proposes another postprocessing algorithm to take advantage of the arrangement patterns and sequential orders of elevator buttons to correct mislabeled buttons. A special method based on CNN softmax output is proposed to deal with the jumping button conditions (e.g. the button for 14th floor follows the button for 12th floor and the button for the 13th floor is missing).

978-1-5386-3742-5/17/$31.00 © 2017 IEEE

II. Related Work

A. Elevator Button Recognition

To make robots autonomously operate elevators, some previous research work was reported on the button detection and button recognition.

There are mainly two approaches to solve the problems of button detection. The first approach is by utilizing the special features of buttons, such as the distinct color or texture. Xinguo Yu [2] developed a vision system of a robot based on the Hough transform, structural inference and multi-symbol recognition techniques. To further develop this idea, Ali A. Abdulla [3] proposed a HSL filter to take advantage of the color, shape and size information to find the buttons. However, due to the diversity of the button styles, the changing light conditions and various backgrounds, this method is not robust. The other approach is to apply a sliding window to traverse the entire image [4]. But the exhaustive search makes this method time-consuming and may finally affect the speed and performance of subsequent button recognition.

In the field of button recognition, Heon-Hui Kim [5] applied the template matching combined with a homography-based transform to achieve the recognition of the tilted elevator buttons. This algorithm is efficient, but it is not robust to noise or environmental variability and not particularly scale or rotation invariant [6]. Kang [7] trained an artificial neural network to recognize the buttons. However, because of the low training efficiency and possible local optimal solutions, the simple neural networks cannot ensure the accuracy of the button recognition results.

Using contextual cues can be an efficient way to improve the accuracy of the algorithms. Ellen Klingbeil [4] utilized several machine learning techniques such as Expectation Maximization (EM) and Hidden Markov Model (HMM) to correct the mislabeled buttons. This approach can increase the recognition precision from 70.9 percent to 88.1 percent, but its computation is very time-consuming.

B. Object Detection

Along with the development of deep learning, especially the CNN (Convolutional Neural Network) and the ROI (Region of Interest) extraction algorithm, great breakthroughs have been achieved in the field of object detection. Detectors like R-CNN [8], Fast R-CNN [9] and Faster R-CNN [10] that adopt deep learning methods have greatly improved the performance of the visual object detection results. However, these deep methods cannot be directly applied to the real-time robotic applications, since the computation resources on a service robots is limited and some other programs are competing for the computational resources. To accelerate the detection process, some detectors based on regression such as the YOLO [11] and the SSD [12] are proposed to achieve fast detection results. These algorithms perform very fast, but they have low accuracy on the detection of small objects.

Despite some promising results in object detection and elevator button recognition, none of them can be applied directly in our problem due to the time-inefficiency, inaccuracy and high susceptibility to the environments. Our goal is to make a robot capable to operate all the elevators in a building, including those in different light conditions and backgrounds. In this paper, we mainly focus on the perception algorithm towards the ROI extraction, the button recognition and the post-processing.

III. Approach

A. Elevator Button Dataset

When applying deep models in object recognition and detection, a rich dataset is the most significant prerequisite for effective training. Because few resources of elevator button images are available online, we prepared our own elevator button datasets by recording videos of elevators from different angles and distances. After extracting the elevator images by frame, the dataset can contain images of different sizes, tilt angles and blur degrees. Some examples of the dataset are shown in Fig. 2.

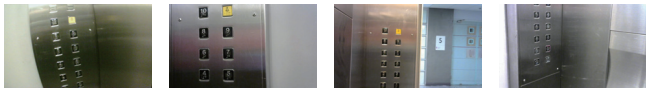


Fig. 2: Samples of the unknown test elevator dataset.

To achieve the robustness for the requirement of different button styles, we choose two different elevators in one building and record the video clips from different time periods. After that, 800 different elevator panel images are extracted from the video clips. Finally, we obtained over 10,000 elevator button images through the pre-processing methods and the contour extraction algorithm mentioned in section III-B. After the manual adjustment, we divided them into 15 classes including 0∼10, open, close, other buttons and the negative class. The examples of the elevator buttons are shown in Fig. 3.

Fig. 3: Samples of training elevator button dataset.

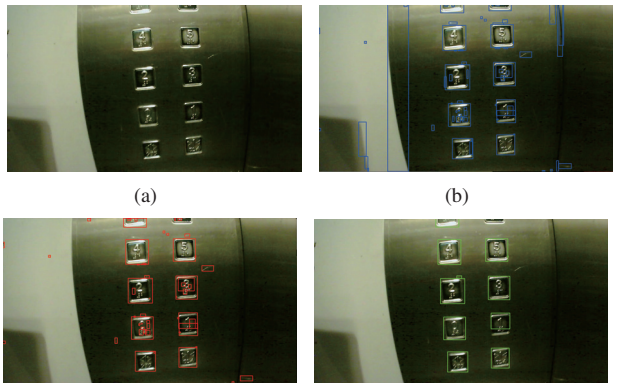
B. ROI extraction

Inspired by the R-CNN algorithm [8], the first part of our algorithm is designed to provide the proper candidate proposals for button recognition to reduce computational load. Since the elevator button always has a square or circular shape with a clear contour, we develop a button detector based on the contour features together with a filter to remove the false positives.

The ROI extractor composes of three parts. We first preprocess the input images into greyscale images and then remove

noise by applying the median filter. After that, a canny detector [15] is applied to extract the edges with a fast response. Specifically, due to the variation of light conditions, we choose the dynamic threshold to remove the redundant edges. The initial candidate proposals are then obtained by extracting the closed contours and their enclosure rectangles.

However, as shown in Fig. 4(b), many false positives are introduced by this detector, since the existence of background objects adds a lot of noises to the detector. Considering that the elevator buttons have the similar sizes and shapes, we use a size filter to remove the initial estimates with abnormal sizes or length-width ratios. The result of the size filter are shown in Fig. 4(c). Furthermore, a simple binary classifier and the NMS (Non-Maximum Suppression) algorithm [16] are applied to remove the false positive noises and the overlapping candidate regions. As shown in Fig. 4(d), all the elevator buttons are precisely identified in the output of the detection algorithm within 0.012 second.



(a) (b) (c) (d)

Fig. 4: The result of the button localization algorithm in the elevator environment. (a) The original image. (b) Blue region proposals from the contour extraction. (c) The effect of size filter. (d) The effect of the simple classifier and NMS algorithm

C. Button Recognition

Different from the traditional button recognition algorithms like the OCR algorithm [3] [4], we adopt the deep methods to precisely classify all the detected buttons and adopt the AlexNet [17] to train on our dataset. To reduce the computational requirement of the deep methods in button recognition, we modify the network structure of the AlexNet and ensure that the accuracy of recognition is sufficiently high.

Considering the similarity between the numbers on the elevator button and handwritten numbers, we initialize our deep models with the trained models in MNIST dataset. Then the AlexNet model is fine-tuned to suit our problems. To save the computational efforts, we reduce the number of convolutional layers to 2 layers and increase the number of kernels in every layer from 6 to 64. Then we change the original pooling layer using the Max-Pooling to ensure the extraction of the detailed information and use the ReLU activation function to replace the sigmoid function. Besides, dropout layers are also added to compensate the over-fitting problem. The improved CNN structure is shown in Fig. 5.

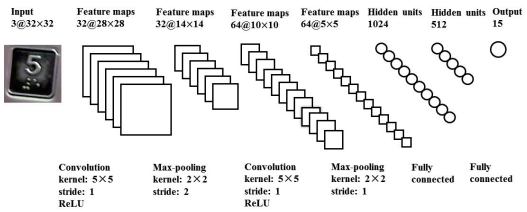


Fig. 5: The structure of the improved CNN

We train our network on the button dataset described in III-A. Some data augmentation approaches, such as horizontal flips, random crops or scales, rotation and translation, are applied to increase the amount of input data. After over 30,000 training iterations, the recognition accuracy of CNN network for the labels reached more than 95%. The training result is shown in Fig. 6.

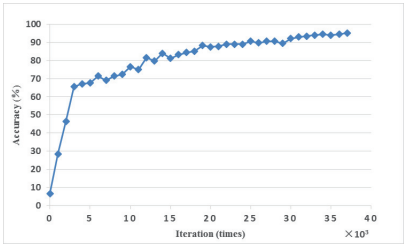


Fig. 6: The changing accuracy rates in the process of training

For every input image, our network can generate the maximum confidence value as the label of each button, which is used together with the coordinate information to feed the control input to a robot arm. Fig. 7 shows the result of our algorithm and the corresponding coordinate information is shown in Table I.

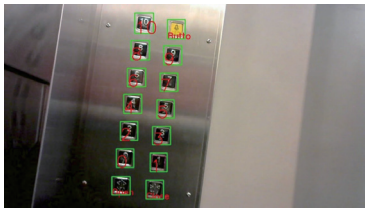


Fig. 7: All the buttons in this image are recognized by our algorithm.

D. Post-processing

Merely based on the button detection and button recognition, we sometimes have errors. For example, it is shown in Fig. 11(a) that floors 6 and 9 are recognized as 8 by mistake and these mistakes may cause the wrong operation of robots.

TABLE I: Coordinates of the button recognition

Button	Coordinate	Button	Coordinate
0	(398,510)	7	(553,243)
1	(518,525)	8	(450,128)
2	(414,412)	9	(566,146)
3	(526,431)	10	(463,31)
4	(424,318)	open	(382,608)
5	(542,337)	close	(503,621)
6	(437,226)	alarm	(579,54)

Inspired by the ability of human to guess the missing button between two corresponding recognized buttons, we develop a post-processing algorithm, which takes advantage of the grid arrangement and sequential order to correct the errors.

As is shown in algorithm 1, the complete process of this algorithm can be divided into three parts: similarity clustering, obtaining arrangement pattern and error correction. In the first part, the horizontal distance between two buttons is defined as the similarity measurement. If the similarity measurement of two buttons is less than the width of a button, these two buttons are placed in one class for a column. After clusttering of the buttons, we can obtain a classified result $\{M_1, M_2, ..., M_{N_r}\}$. Every class M_i includes all the buttons in the same row and N_r is the number of columns.

To correct the error in recognition, the next goal is to get the order of buttons. Based on the observations, the elevator buttons are mainly arranged in four patterns described in Fig. 8, typically from the bottom to the top. To decide on which pattern is the real one, we define the frequency of the effective ordered pair n_{order} as the evaluation index. If the elevator button sequence of one arrangement pattern is $\{b_1, b_2, ..., b_i, ..., b_k\}$, the corresponding frequency n_{order} can be calculated as follows:

$$n_{order} = \sum_{i=1}^{k-1} \sigma_1(b_i, b_{i+1}) \tag{1}$$

$$\sigma_1(b_i, b_{i+1}) = \begin{cases} 1, & \text{if } b_i < b_{i+1} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

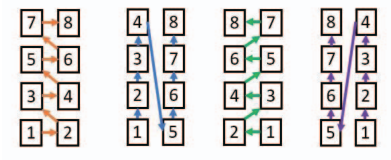
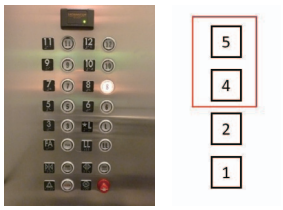


Fig. 8: Four arrangement patterns of elevator panels .

By comparing the frequency n_{order} of the four patterns, the actual button arrangement pattern is determined. This algorithm can transform the classified results $\{M_1, M_2, M_3, ..., M_{N_r}\}$ into a one-dimensional vector $\{s_1, ..., s_k\}$. The element s_i is a four-dimensional vector for describing the coordinates of the button center, the width and the height of the button. The corresponding button labels for $\{s_1, ..., s_k\}$ are $\{x_1, ..., x_k\}$.

Some elevators have the jump buttons in their elevator panels. As shown in Fig. 9(a), the button 4 is missing between the buttons 3 and 5. Thus if we only correct the errors with the order, the button 5 will be incorrectly marked as an error. We solve this problem based on the observation that every jump button still has the sequential relationship with the button above or below. As shown in Fig. 9(b), although a jump button exists between the buttons 2 and 4, the button 4 still has the sequential relationship with the button 5. Thus we define the error detection function as follows.

$$\sigma_2(x_i) = \begin{cases} 1, & \text{if } x_i = x_{i-1} + 1 \text{ or } x_i = x_{i+1} - 1 \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$



(a) (b)

Fig. 9: (a) The sample of the elevator panel with some jump buttons. (b) The diagram of the law of jump buttons.

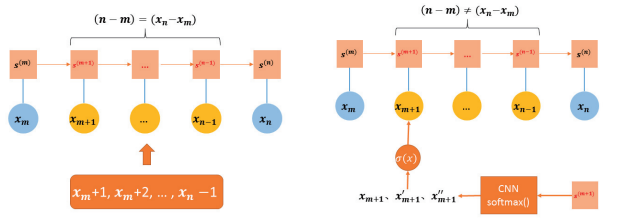
Here, $\sigma_2(x_i) = 1$ means that the button x_i is the correct button, $\sigma_2(x_i) = 0$ means that the button x_i is the wrong button. After detecting the mistakes, we utilize the button arrangement patterns and the softmax function of CNN to correct them. Two strategies for the correction are described as follows.

1) Correct errors with the arrangement pattern: The continuous recognition errors for buttons $\{s^{m+1}, ..., s^{n-1}\}$ happened between the button s^m and s^n. If there exists no jump button (the value of $(n - m)$ is equal to the value of $(x_n - x_m)$), we can change these wrong labels of buttons into $\{x_m + 1, x_m + 2, ..., x_n - 1\}$. Here, x_m and x_n are respectively the correct labels of the button s^m and s^n. The complete process of this strategy is shown in Figure 10(a)

2) Correct errors with the confidence possibility: If there exists a jump button in the sequence, for the button in error $s^i \in \{s^{m+1}, ..., s^{n-1}\}$, we can obtain the first three possible labels x_i, x_i', x_i'' and rank them from the low possibility to high possibility. This can ensure that the higher possible labels for the button can be given higher priority. The ranking is defined by the function $\sigma_3(x_i)$.

$$\sigma_3(x_i) = \begin{cases} 1, & \text{if } x_i = x_m + (i - m) \text{ or} \\ & \quad x_i = x_n - (n - i) \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Here, x_m and x_n are the correct labels for the corresponding buttons. The process of this strategy is shown in Figure 10(b) as follows. The complete process of the post-processing algorithm is described in algorithm 1.



(a) (b)

Fig. 10: Two strategies for error correction. (a) The diagram of correction with the arrangement order. (b) The diagram of correction with the confidence possibility which the softmax function outputs

Algorithm 1 Post-processing algorithm

Input: sequence of detection
Output: corrected button sequence

Initialize: num, mis_button, co_error as an empty list.
$column\_group$ =get_coordinate($button\_detection$)
$example\_model$ =get_fourmodel($column\_group$)
for every $model \in example\_model$ **do**
  num.append(get_frequency($model$))
end for
seq_model =comp_model($num, example\_model$)
for every $button \in seq\_model$ **do**
  if $\sigma_2(button.value) == false$) **then**
    mis_button.append($button$)
  end if
end for
for every $mistake \in mis\_button$ **do**
  if isContinuous($mistake$) **then**
    co_error.append($mistake$)
    continue
  else
    $[bm, bn]$ =find_rightbutton(co_error)
    if existJumpButton(co_error) **then**
      for $b \in co\_error$ **do**
        $[p'', p', p]$ =CNNsoftmax(b)
        $b.value \leftarrow$get_value($[p, p', p'']$)
      end for
    else
      $[b_{m+1}.value, ..., b_{n-1}.value] \leftarrow (x_{m+1}, x_{n-1})$
    end if
  end if
end for
return $correct\_button$

The result of our post-processing algorithm is shown in Fig. 11(b). Comparing with Fig. 11(a), the labels for buttons 6 and 9 are corrected to the correct numbers through the post-processing algorithm. In order to describe the effect of the post-processing algorithm more accurately, some quantitative evaluations are presented in the next section.

(a) (b)

Fig. 11: The effect of post-processing algorithm. (a) Some mistakes of button recognition. (b) The corrected results

IV. Experiments and Discussions

This section presents the property and robustness of the algorithm through experiments. To evaluate the real-time and robust performance of the algorithm, we first test each step in our pipeline separately. Then, an experiment to recognize unknown elevator panels is conducted to test the generality of our algorithm. Finally, the robust performance of our algorithm is further tested under the variation in light condition, ambiguity, the depth and the incline angle.

To carry out the test, we remake the test dataset of 180 images of known elevators and 280 images of four unknown elevators. The images in the dataset are collected under different light conditions, ambiguities and incline angles to ensure the validity of the test. The images of the untrained elevators in the test dataset are shown in Fig. 12.

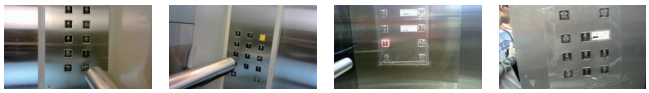


Fig. 12: Samples of the unknown test elevator dataset.

A. Evaluation by Steps

To validate the effectiveness of the region proposal extraction, the recognition accuracy of the CNN model, and the error correction efficiency of the post-processing, the evaluation of each step is carried out separately on the test dataset of known elevators. The result of the test is shown in Table II.

TABLE II: The robust and real-time performance of three steps in the algorithm

Method	Time(s)	Precision	Recall	F1-measure
ROI extraction	0.012	0.983	0.971	0.978
Button Recognition	0.406	0.938	0.987	0.962
Post-processing	0.0005	0.980	0.982	0.981

As shown in Table II, the accuracy of ROI extraction is 98.3%, which ensures the efficiency of the extraction of region proposals. Our trained CNN network performs well in recognizing the elevator buttons with a precision of 93.8%. After that, the post-processing algorithm increased the precision to 98.0%, which ensures its suitability for a robotic application. During the process of the recognition, the buttons 6, 8, 9 are easily mistakenly recognized, since they have similar structures. Thus, in the future, our CNN network should be designed to handle these specific confusing buttons.

Besides the robust performance, our algorithm can process each image within 0.4s (in a regular personal computer with a GeForce GTX 770 graphic card and 32GB memory), which is faster than fast-RCNN algorithm [9], which requires nearly 3s to process each image. Thus our algorithm can achieve a near real-time performance for application in robot perception.

B. Evaluation of Untrained Elevators

To evaluate the generality of our algorithm, we test the recognition performance on the dataset of a known elevator and four unknown elevators shown in Fig. 12. The result is shown in Table III.

TABLE III: The robust performance of the algorithm to recognize different elevators

Object	Precision	Recall	F1-measure
Known Elevators	0.980	0.982	0.981
Unknown Elevators	0.876	0.900	0.888

As can be seen in Table III, the recognition system maintains the robust performance at about 98% accuracy for the familiar elevator, which means our algorithm can be trained to have a robust performance for autonomous operation of robots in any known elevator with only a small dataset of the elevator.

Furthermore, only trained by one elevator, our system achieves a robust performance of 87.6% accuracy for the four unknown elevators in our dataset. This exhibits the great generality of our algorithm, which further can be utilized to recognize more types of elevators after increasing the number and the type of our training dataset

By analyzing the errors, 15% of them are occurred when the button is blocked in the elevators. In the actual operation, the robots can adjust their positions to avoid the occlusion to reduce the errors. Therefore, the accuracy of our algorithm can be further improved in the actual situations.

C. Robust Performance

1) Recognition Performance with Variation of Depth and Inclined Angle: To validate the effective region of the recognition system, we test our algorithm with different depths and angles and study the accuracies of each possibility. Considering the range of the robot motion inside an elevator, the distance between the camera and the button panel is defined from 30cm to 80cm at every 10cm increment and the inclined angles are defined from -75° to 75° with an angle interval of 25°.

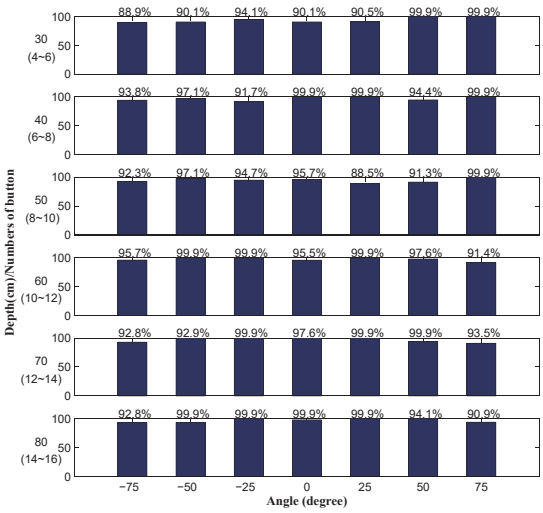


Fig. 13: Recognition rates with various distances and rotation angles

Fig. 13 shows the recognition accuracy of our algorithm at each test position. The overall performance keeps a high recognition level (the accuracy is over 97.0% in most cases). Comparing to the methods reported in [5], our system expands the effective distance from 45cm to 80cm and increases the range of the rotation angle from 45° to 75°. As a result, this expanded coverage enables the robot to operate the elevators more easily in more positions without moving close to the elevator panels.

2) Robustness to Ambiguity: Because of the movement of robot and the shake of the elevator, the blurring and ambiguity always occur in images. To obtain the robust performance in recognizing blurred button images quantitatively, we blur the images in our test dataset using the Gaussian Blur. The size of the Gaussian filter is 5×5 and the ambiguity is modified by changing the standard derivation of the Gaussian distribution (StDev) from 0 to 12. The recognition accuracy is shown in Fig. 14 as follows.

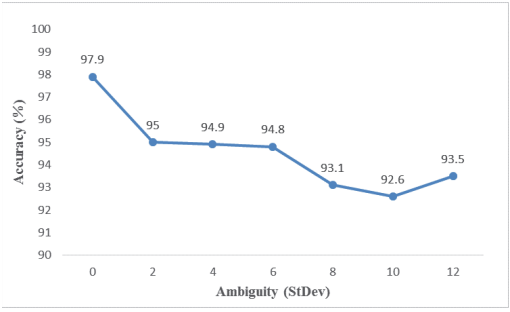


Fig. 14: Changing recognition rates at various ambiguity degrees

As shown in Fig. 14, our algorithm maintains a recognition accuracy of over 92%, even if the image is completely blurred. Comparing to the methods proposed in [4][5], our algorithm overcomes the difficulty of these blurs due to camera motions, which ensures the robustness of recognition when the robot adjust its position and moves to operate the elevator.

3) Robustness to Changing Light Conditions : The dim light condition and the reflective surface of the elevator panels make it difficult to recognize the elevator buttons. To test the recognition accuracy in various light conditions, we modify the brightness of the images in test dataset by adding or subtracting a portion of the average brightness of original images. The recognition result is shown in Fig. 15 as follows.

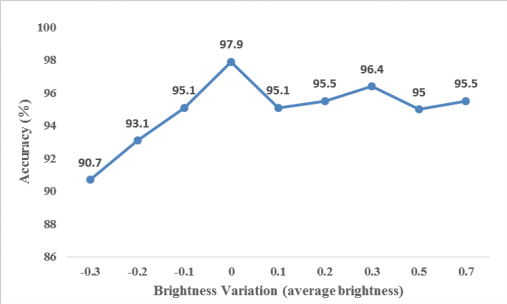


Fig. 15: Changing recognition rates with brightness variation

As shown in Fig. 15, the recognition system achieves an accuracy of over 90% when the brightness variation of images varies from the dim condition subtracted by 0.3 average brightness to the luminous condition added by 0.7 average brightness. This ensures the robust performance of our algorithm in changing light condition caused by the opening and closing of the elevator doors and varying elevator environments.

V. Conclusions

In this paper, we mainly focus on constructing an elevator button recognition system which can achieve a near real-time and robust performance for the robotic operation of elevators. The system covers three significant parts, including ROI extraction, button recognition and post-processing. Deep CNN model is trained on our elevator dataset to recognize elevator buttons. To provide the proper region proposals for this model, contour-based detection, size filter and simple classifier are proposed. Even in a blurred image caused by camera motion, we can compensate by using the arrangement pattern of buttons and confidence probability level of CNN output to correct the errors. By validating our algorithm on the test dataset, the achieved precision is 98.0% for known elevators and 87.6 percent for unknown elevators. Furthermore, the system shows a promising result even in a large depth/angle level, blurred images caused by the camera motion and changing light conditions. Besides, the algorithm can run at a speed of 3 frames per second on a regular computer.

In the future, we need to improve the precision for recognizing untrained elevators. Our dataset will be extended by including more styles of elevators and our CNN model will be modified to adapt the dataset. In addition, we will also work on improving the contour extraction by using the deep methods, to ensure the robustness with respect to the noise and light conditions. More experiments integrating with the robot platform will be conducted as well.

Acknowledgment

Research presented in this paper was supported by the Shenzhen Science and Technology Program No. JCYJ20170413161616163 awarded to Professor Max Q.-H. Meng.

References

[1] J. Miura, K. Iwase, and Y. Shirai, "Interactive teaching of a mobile robot," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 3378–3383.

[2] X. Yu, L. Dong, L. Li, and K. E. Hoe, "Lift-button detection and recognition for service robot in buildings," in *IEEE International Conference on Image Processing*, 2010, pp. 313–316.

[3] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A robust method for elevator operation in semi-outdoor environment for mobile robot transportation system in life science laboratories," in *IEEE Jubilee International Conference on Intelligent Engineering Systems*, 2016.

[4] E. Klingbeil, B. Carpenter, O. Russakovsky, and A. Y. Ng, "Autonomous operation of novel elevators for robot navigation," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 751–758.

[5] 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.

[6] D. Troniak, J. Sattar, A. Gupta, J. J. Little, W. Chan, E. Calisgan, E. Croft, and M. V. D. Loos, "Charlie rides the elevator – integrating vision, navigation and manipulation towards multi-floor robot locomotion," in *International Conference on Computer and Robot Vision*, 2013, pp. 1–8.

[7] J. G. Kang, S. Y. An, and S. Y. Oh, "Navigation strategy for the service robot in the elevator environment," in *International Conference on Control, Automation and Systems*, 2007, pp. 305–326.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[9] R. Girshick, "Fast r-cnn," in *IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[10] R. G. J. S. Shaoqing Ren, Kaiming He, "Faster R-CNN: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," pp. 779–788, 2015.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*, 2016, pp. 21–37.

[13] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[14] C. L. Zitnick and P. Dollr, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*, 2014, pp. 391–405.

[15] Y. Lecun, P. Haffner, L. Bottou, on, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Springer Berlin Heidelberg, 1999.

[16] R. Rothe, M. Guillaumin, and L. V. Gool, *Non-maximum Suppression for Object Detection by Passing Messages Between Windows*. Springer International Publishing, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1097–1105.