Container-code recognition system based on computer vision and deep neural networks

Cite as: AIP Conference Proceedings **1955**, 040118 (2018); https://doi.org/10.1063/1.5033782
Published Online: 18 April 2018

Yi Liu, Tianjian Li, Li Jiang, and Xiaoyao Liang

View Online Export Citation

Container-code recognition system based on computer vision and deep neural networks

Yi Liu[a], Tianjian Li[b], Li Jiang[c], Xiaoyao Liang[d]

School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai, China

[a] lewissjtu@sjtu.edu.cn
[b] ltj2013@sjtu.edu.cn
[c] ljiang_cs@sjtu.edu.cn
[d] liang-xy@sjtu.edu.cn

Abstract. Automatic container-code recognition system becomes a crucial requirement for ship transportation industry in recent years. In this paper, an automatic container-code recognition system based on computer vision and deep neural networks is proposed. The system consists of two modules, detection module and recognition module. The detection module applies both algorithms based on computer vision and neural networks, and generates a better detection result through combination to avoid the drawbacks of the two methods. The combined detection results are also collected for online training of the neural networks. The recognition module exploits both character segmentation and end-to-end recognition, and outputs the recognition result which passes the verification. When the recognition module generates false recognition, the result will be corrected and collected for online training of the end-to-end recognition sub-module. By combining several algorithms, the system is able to deal with more situations, and the online training mechanism can improve the performance of the neural networks at runtime. The proposed system is able to achieve 93% of overall recognition accuracy.

Keywords: Text Detection; Text Recognition; Container-code; Deep Neural Networks; Computer Vision.

INTRODUCTION

With the development of global trade, ship transportation industry needs to transport more and more containers. The information of each container should be recorded for management and maintenance. The cost spent on manually recording is increasing in recent years, due to the increasing number of containers. Therefore, an automatic container-code recognition system is required to achieve efficient container-code recognition with high recognition accuracy.

The problem of container-code recognition is basically considered as recognizing text in natural image. Many detection and recognition algorithms have been proposed to solve the problem. Some of them are based on computer vision [1]- [5], such as MSER (Maximally Stable Extremal Regions) [1] and SWT [2], and others are based on deep neural networks [3]- [6]. The detection algorithms based on computer vision can generate precise bounding box for text regions, but they are not robust under disturbance. Algorithms based on neural networks are robust, while their generated bounding box isn't as precise as algorithms-based computer vision. For text recognition, some methods utilize character segmentation and recognize the characters one by one, while some other methods recognize the whole text and output the complete text string [7][8], which is called end-to-end recognition. All these detection and recognition algorithms have their drawbacks, and the proposed container-code recognition system is able to combine several algorithms to achieve better detection and recognition performance. In recent years, many researches on container-code recognition have been published [9]- [11], but they will fail in variant disturbances, such as light reflection and image blur. Therefore, a robust container-code recognition system should be proposed.

Advances in Materials, Machinery, Electronics II
AIP Conf. Proc. 1955, 040118-1–040118-9; https://doi.org/10.1063/1.5033782
Published by AIP Publishing. 978-0-7354-1654-3/$30.00

The container-code is composed of three parts, each of which represents different information. The first four letters represent the company which owns a certain container. The next seven digits represent the identity of containers, and the last digit is the check digit. The last four characters represent the type of containers. In this paper, the proposed recognition system is able to recognize all necessary information of containers.

The proposed container-code recognition system consists of two main modules, detection module and recognition module. The detection module applies both algorithms based on computer vision and neural networks to achieve both high robustness and precision. The recognition module utilizes character segmentation and end-to-end recognition. By using both of the methods, the recognition module is able to achieve better recognition accuracy. The proposed system also provides a mechanism to update the neural networks in the system. Details of the proposed system will be described in the next section.

TECHNICAL DETAILS

Architecture overview of the system

The proposed system is composed of detection module and recognition module as shown in Fig. 1. In the detection module, MSER [1] is chosen as the algorithm-based computer vision, which is both efficient and effective for container-code detection. CTPN (Connectionist Text Proposal Network) [12] is chosen as the neural networks in detection module, which is robust in text detection under variant disturbance. The detection results of MSER and CTPN will be sent to the combination sub-module to generate a better combined detection result. The combined results of two methods are sent to recognition module. The detection module also contains a Sub-module, which is responsible for updating CTPN model through online-training using combined detection results. By updating CTPN, CTPN can achieve higher accuracy. When CTPN achieves the same performance of the whole detection module, MSER can be eliminated from the detection module. The recognition module chooses CRNN (Convolutional Recurrent Neural Network) [8] as its end-to-end recognition method and uses the information provided by the detection module to do character segmentation. The combination sub-module will verify the two recognition results according to the check digit in the container-code, and output the recognition result which passed the verification. Similarly, the recognition module also contains a sub-module, which is responsible for collecting all recognition results and updating CRNN during the runtime.

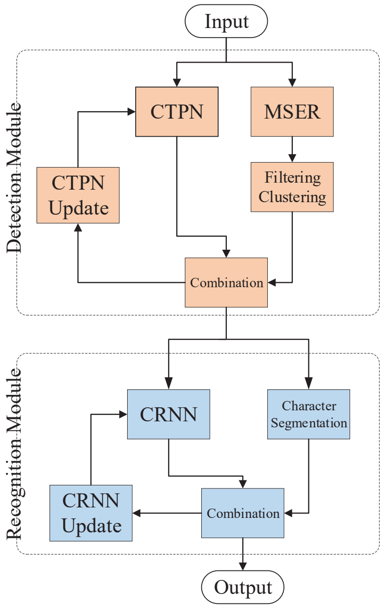

FIGURE 1. Architecture of the proposed container-code recognition system

The proposed container-code recognition system combines several algorithms to complete the same task. Through combination, the system is able to avoid drawbacks of each individual algorithm. Besides, the system also provides mechanisms to update the neural networks used in the system at the runtime.

Detection Module

Detection Sub-Module Based on Computer Vision

The color of the container-code is stable, and it can be obviously distinguished from its background. MSER can be effective to detect text with this kind of feature. Therefore, MSER is applied in this module to detect all the possible text regions. Fig. 2 represents an example of all possible text regions detected by MSER.

According to the geometry features of the text regions, the system may filter out many non-text regions. As shown in Fig. 2 there are many irrelevant regions, which are not text regions. Then, the system need accurately locate the container-code

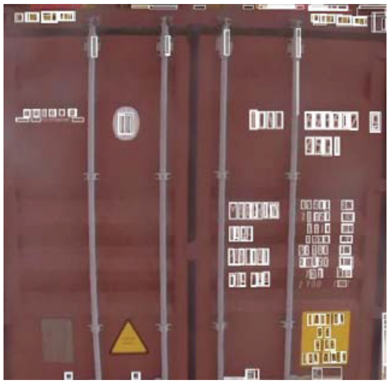

FIGURE 2. All possible text regions generated by MSER

Region. It can be easily observed that characters in container-code share the similar height and locate at approximately the same horizontal line. Based on these facts, the system will do clustering to all the text regions, and decide which cluster contains the container-code. Equations (1) and (2) represent the clustering references. To judge whether a set of regions belong to the same cluster, values calculated from (1) and (2) will be compared with some thresholds. If all the values from (1) and (2) are lower than the thresholds, the set of regions belong to the same cluster.

$$\delta = \frac{1}{n}[\alpha \cdot \sum_{i=1\ldots n}(y_i - \overline{y})^2 + \beta \cdot \sum_{i=1\ldots n}(h_i - \overline{h})^2] \quad (1)$$

$$\delta_i = \alpha\left|y_i - \overline{y}\right| + \beta\left|h_i - \overline{h}\right|, i \in \{1,2,\ldots,n\} \quad (2)$$

In the formulas, n represents the number of regions in the set of text regions. y_i and h_i represent the vertical coordinate and height of a text region in the set. $\overline{y}$ and $\overline{h}$ represent the average vertical coordinate and the average height of all text regions in the set. α and β are two adjustable values, which represent how much the vertical coordinate and the height of text regions affect the clustering.

Before clustering, the system will set the thresholds for δ and δ_i . If δ and δ_i of a set of text regions are lower than the thresholds, the set of text regions belong to the same cluster. When doing clustering, the system will cluster the text regions one by one, and find a proper existing cluster for each text region. If there is no cluster with

δ and δ_i lower than the thresholds after adding a text region, then a new cluster will be created for the text region. And if there is no cluster existing, a new cluster will also be created. The details of the procedure is shown in Fig. 3

After clustering, the proposed system will determine which cluster contains the container-code based on the character number and patterns of container-code. Then, the detected regions will be sent to the combination sub-module. The combination sub-module will synthesize the detection results from MSER and CTPN to generate better detection results.

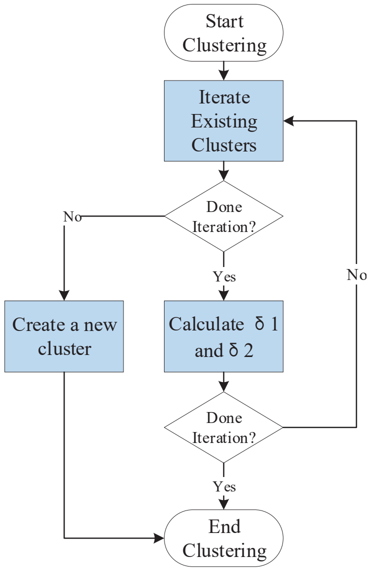

FIGURE 3. Text regions clustering procedure

Detection Sub-Module Based on Neural Networks

Text detection is much more complex compared to object detection. For object detection, the bounding box is acceptable if it covers over 80% of the detected object, while the text detection requires higher coverage ratio. Therefore, the proposed system applies CTPN for container-code detection, which can generate a bounding box for container-coder with a high coverage ratio. But the generated bounding box is often larger than the actual container-code region so that the detection of CTPN is not as precise as MSER. Fig. 4 shows an example of detection result of CTPN.

As shown in Fig. 4, there are some text regions which are not container-code, but can be filtered through the combination with the results from MSER. The detected result will be sent to combination sub-module for synthesis.

Combination Sub-module

Generally, CTPN can distinguish the text region and the non-text region better, but its generated bounding boxes are not as precise as the bounding boxes of MSER. MSER might miss some of the characters during detection. Therefore, the combination sub-module is responsible for optimizing the bounding box of CTPN and finding the missing character in the detection of MSER.

To optimize the bounding boxes of CTPN, the system will retrieve the information of the detection results from MSER. As stated before, the characters in the container-code share the similar height, and locate at approximately the same horizontal line. By measuring the text height and the horizontal line of the detected text regions of MSER, the system is able to determine the text line position and also the text height of the container-code. With the necessary information, the system can optimize the bounding boxes of CTPN, which can generate more precise bounding boxes, as shown in Fig. 5

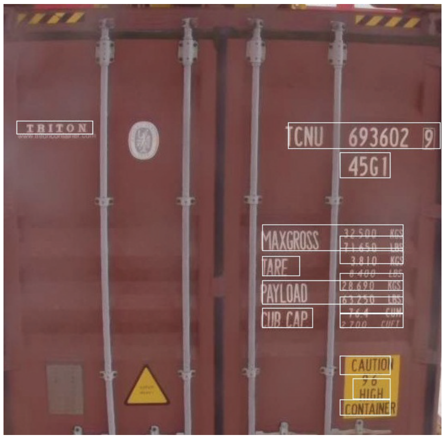

FIGURE 4. Detected regions by CTPN

To find the missing characters among the detected text regions of MSER, the system will utilize the optimized bounding boxes of CTPN, and pinpoint each detected text region in the container-code according to the printing pattern. And this position information can be used to predict the missing characters. After finding the missing characters, they will be added to the detected text regions of MSER, to complete the container-code characters. This method of finding missing characters works well in most of the case.

After combination, the optimized bounding box of CTPN and the refined detected regions of MSER will be sent to recognition module. The recognition module will utilize the detection result to generate a final container-code string. Besides passing results to recognition module, this sub-module will also pass the results to CTPN updating module, which will use the optimized bounding box to update the weights of CTPN, therefore CTPN can improve its detection performance at the runtime.

Recognition Module

Character segmentation recognition module

In this module, character segmentation is applied first. After segmentation, characters will be recognized one by one using neural networks based on ResNet[13]. Most of the algorithms perform the character segmentation based on the RGB information. However, these segmentation algorithms will fail in the complex environment. In the detection module, the method based on computer vision generates bounding boxes with each box covers a single character. If the position information of each character is provided, the boundary between characters can be determined. Therefore, the information provided by detection module will be used as the reference of character segmentation. After Segmentation, this module will use pre-trained neural networks to recognize each character. In this module, ResNet is used as the recognition neural networks.

When doing recognition, letter I is often confused with digit 1. But this recognition error can be avoided according to the features of the container-code.

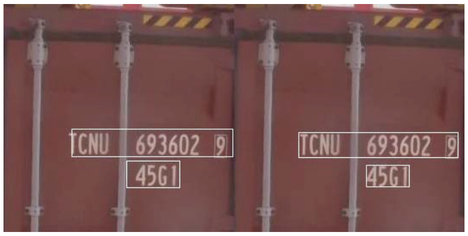

FIGURE 5. Optimized bounding boxes of CTPN

End-to-end recognition module

Generally, the accuracy of character segmentation has a significant impact on the recognition accuracy. When faults exist in character segmentation, the recognition will totally fail. The end-to-end recognition module doesn't require character segmentation. In this recognition module, the detected container-code region is regarded as a whole and recognized by CRNN which will generate a text string as its recognition result.

The accuracy of end-to-end recognition module depends on the size of training data set. But the collection of a large training data set requires too much manual efforts. Therefore, the system provides a mechanism to update CRNN at the runtime, which alleviates the efforts paid on CRNN training.

Combination recognition module

As stated before, the last digit in the container-code is a check digit that can be used to verify whether the container-code is correctly recognized. The combination module checks the results of above two recognition modules as shown in Fig. 6. The recognition result which passes the verification is regarded as the correct container-code, and it will be sent to the output.

Generally, the size of training data set determines the accuracy of end-to-end recognition module, and it is time-consuming to manually collect the training data set. In order to alleviate the efforts, put on the collection of training data set, recognition results that pass the check will be collected into the training data set. If both recognition methods fail to pass the verification, the combination module will remind the administrator, ask for a correction of the recognition result. After correction, the result will be collected for online-training of CRNN. CRNN can continue to improve its recognition accuracy at the runtime, so that it can greatly save the cost spent on the collecting training data set.

EXPERIMENTAL RESULTS

Detection Experimental Result

For experimental purpose, 200 manually labeled images are used as test data, which contain the position information of the container-code. To analyze the performance of MSER, CTPN, and the combined results, recall rate and coverage ratio are used as evaluation standards. Recall rate represents how many characters of container-code are detected. Coverage ratio is evaluated by r_1 and r_2, which are calculated through the following equations:

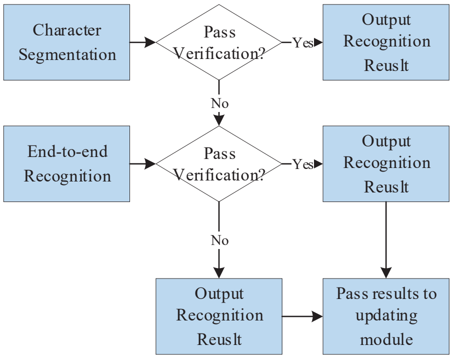
Character Segmentation
Pass Verification?
Yes
Output Recognition Reuslt
No
End-to-end Recognition
Pass Verification?
Yes
Output Recognition Reuslt
No
Output Recognition Reuslt
Pass results to updating module

FIGURE 6. Procedures of combination in recognition module

$$r_1 = area_{overlap} / area_{det} \tag{3}$$

$$r_2 = area_{overlap} / area_{GT} \tag{4}$$

$area_{det}$ represents the area of the detected text regions, $area_{GT}$ represents the area of the ground truth container-code region, and $area_{overlap}$ represents the area of the overlapped regions of $area_{det}$ and $area_{GT}$. By evaluating these values, the comparisons of CTPN, MSER, and the combined detection results are shown in Table 1.

As shown in Table 1, MSER has a relatively low recall rate, which means there are more missing characters in the detection results of MSER. Through combination, the overall recall rate is increase by 9.7% compared to recall rate of MSER. CTPN has the lower value of r_1 , which means the detected regions of MSER is more precise. The combined results increase the value of r_1 by 12.9% compared to CTPN, which means the optimized bounding boxes of CTPN is much more precise than the original results of CTPN. MSER is much lower than CTPN on the value of r_2 , which means its detected regions have less overlapped area than the detected regions of CTPN. The reason is that MSER has a lower recall rate and its detection is character-wise, therefore the space between characters will not be detected, which results in much lower value of r_2 than CTPN. Through Combination, the value of r_2 is increased.

By analyzing the experimental data, A conclusion can be made that the combination sub-module effectively combines both the detection results of CTPN and MSER, and yields better detection results by avoiding the drawbacks of the two methods.

TABLE 1. Detection Experimental Results

	Recall rate	**r_1**	**r_2**
MSER	83.1%	89.2%	59.2%
CTPN	92.6%	78.4%	88.2%
Combined Results	93.4%	91.3%	93.4%

TABLE 2. Recognition Experimental Results

	Failed number	**Recognized number**	**accuracy**
Character segmentation	167	33	83.5%
CRNN	133	67	66.5%
Combined recognition	186	14	93.0%

Recognition Experimental Results

For experimental purpose, the 200 labeled images in the detection experiment are also labeled with correct container-code. The experiment will evaluate the recognition accuracy of character segmentation recognition, end-to-end recognition, and the combined recognition. The experimental data is shown in Table 2.

As shown in Table 2, the recognition accuracy is improved by combining both character segmentation recognition and end-to-end recognition, which means the utilization of the two recognition methods can achieve better performance.

The Updates of Neural Networks

In the proposed system, CTPN and CRNN will be updated at the runtime, therefore their performance can be improved. To evaluate how this update mechanism improves the performance of CTPN and CRNN. 2500 images are used as test data. And updates are done for every 500 images. Fig. 7 shows the changes of recall rate and coverage ratio of CTPN after each update. And it shows CTPN can generate better detection after each update. Fig. 8 shows the changes of the recognition accuracy of CRNN along with each update. And improvements can be seen in Fig. 8.

DISCUSSION AND CONCLUSION

In this paper, a container-code recognition system based on synthesized methods is proposed. The system utilizes both algorithms based on computer vision and neural networks in its detection module. Through combination the detection module is able achieve better performance than both of the two methods, and avoid drawbacks of each. The recognition module applies both character segmentation recognition and end-to-end recognition. By combining the recognition results, higher recognition accuracy is achieved. The system also provides a mechanism to update CTPN and CRNN, which can improve the performance of CTPN and CRNN at the runtime.

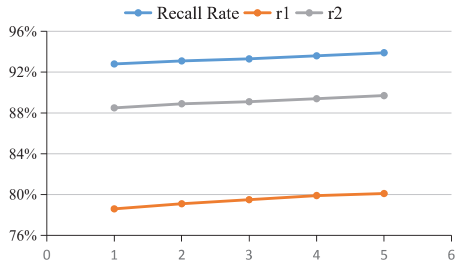

FIGURE 7. Changes of Recall rate and coverage ratio after each update

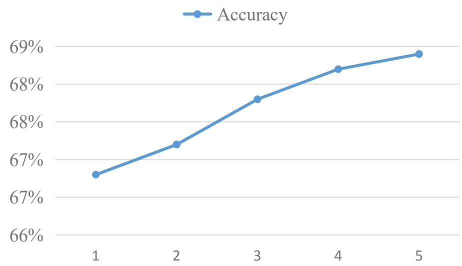

FIGURE 8. Changes of recognition accuracy after each update

REFERENCES

1. H. Chen, S.S. Tsai, G. Schroth, D.M. Chen, R. Grzeszczuk, and B. Girod. "Robust text detection in natural images with edge-enhanced maximally stable extremal regions." In Image Processing (ICIP), 2011 18th IEEE International Conference on, pp. 2609-2612. IEEE, 2011.
2. B. Epshtein, E. Ofek, and Y. Wexler. "Detecting text in natural scenes with stroke width transform." In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2963-2970. IEEE, 2010.
3. M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. "Reading text in the wild with convolutional neural networks." International Journal of Computer Vision 116, no. 1, pp.1-20, 2016.
4. M. Jaderberg, A. Vedaldi, and A. Zisserman. "Deep features for text spotting." In European conference on computer vision, pp. 512-528. Springer, Cham, 2014.
5. T. Wang, D.J. Wu, A. Coates, and A.Y. Ng. "End-to-end text recognition with convolutional neural networks." In Pattern Recognition (ICPR), 2012 21st International Conference on, pp. 3304-3308. IEEE, 2012.
6. T. He, W. Huang, Y. Qiao, and J. Yao. "Text-attentional convolutional neural network for scene text detection." IEEE transactions on image processing 25, no. 6, pp. 2529-2541, 2016.
7. P. He, W. Huang, Y. Qiao, C.C. Loy, and X. Tang. "Reading Scene Text in Deep Convolutional Sequences." In AAAI, pp. 3501-3508. 2016.
8. B. Shi, X. Bai, and C. Yao. "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition." IEEE transactions on pattern analysis and machine intelligence, 2016.
9. W. Wu, Z. Liu, M. Chen, X. Yang, and X. He. "An automated vision system for container-code recognition." Expert Systems with Applications 39, no. 3, pp. 2842-2855, 2012.
10. K.M. Tsai, and P.J. Wang. "Predictions on surface finish in electrical discharge machining based upon neural network models." International Journal of Machine Tools and Manufacture 41, no. 10, pp.1385-1403,2001.

11. S. Xu, Z.F. Ma, and W. Wu. "Container Number Recognition System Based on Computer Vision." Video Engineering 5 pp. 035, 2010.
12. Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. "Detecting text in natural image with connectionist text proposal network." In European Conference on Computer Vision, pp. 56-72. Springer International Publishing, 2016.
13. 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, pp. 770-778. 2016.