An Implementation of Automatic Container Number Recognition System

Chanathip Roeksukrungrueang,
Theeradet Kusonthammrat,
Napasin Kunapronsujarit,
Thiwat Nunjaruwong
Verily Vision Co., Ltd.
98/91 M. 6, Bang Mueang, Mueang Samutprakan,
Samutprakan 10270, Thailand
verilyvision@gmail.com

Sorawat Chivapreecha
Department of Telecommunication Engineering,
Faculty of Engineering,
King Mongkut's Institute of Technology Ladkrabang,
Bangkok 10520, Thailand.
sorawat@telecom.kmitl.ac.th

***Abstract*— This paper presents a design and implementation of automatic container number recognition (ACNR) system for an oversea logistic purpose. Usually in import/export industry in Thailand, the container number which is used for logistic management and also customs duties is manually inspected and recorded by human at each gate of port. However, the manual operation by human causes of error and slow speed of operation from bottle neck, especially customs clearance verification process which customs officers and terminal operators must deal with containers individually as they enter and leave terminals via a gate. By these reasons, an automatic system is required. The cameras capture containers and by using image processing and optical characteristic recognition (OCR) techniques, a container number will be achieved and then customs clearance verification can be automatically checked via database system including other logistic management. The proposed ACNR system can reduce the bottle neck issue, enhance time utilization, accuracy rate, and data management efficiency. From an experiment, total samples of 218 containers are tested, the proposed system can give an accuracy rate of 95.41% with time utilization < 10 seconds per/container.**

Keywords—Computer Vision; Container Number Recognition; Optical Character Recognition (OCR);

I. INTRODUCTION

At any container port or terminal, the number of lorries and containers entering and leaving is phenomenal during peak hours. Containers are loaded on to lorries and manually inspected and verified by human. However, the manual operation by human causes of error and slow speed of operation from bottle neck, especially customs clearance verification process which customs officers and terminal operators must deal with containers individually as they enter and leave terminals via a gate. In order to better control and improve efficiency, an automatic system is required. This paper presents a method how to detect Container number which appear on the container according to ISO 6346, is a national standard which describes the identification of a shipping container. These containers have various appearance whether size, font and aspect. We apply this application as a real-time system which can tracking region of interest and process simultaneously.

Locating text-line regions on container is a challenge because variety of their positions, colors of characters and backgrounds, font types and sizes. In [1] proposed a method, where image edges were extracted and projected horizontally with a Gaussian smoothing filter. The position of the local maximums and local minimums of the smoothed histogram were found. From each local maximum, the top and bottom position of each text-line region can be obtained. In segmentation-based methods, text-line region is segmented. Then, vertical and horizontal projections, connected component analysis [2], or contour analysis [3] are applied to obtain the position of each character and other binary object measurements such as height, width, and area may be used to eliminate noises. To recognize character images, many methods have been investigated. The template matching technique [4] is one of method that suitable to recognize characters with non-deformable and fixed-size fonts. But container characters do not meet such requirements. Self-organized neural network [5], probabilistic neural networks [6], back-propagation neural network [7], SVM [8] are well adopted methods in recognition.

II. PROPOSED SCHEME

The proposed scheme in this paper were obtained from CCTV IP camera at the ICD. Duration of the experimental data were collect and process offline in the first phase and the second phase we process real-time with an authentic environment. The footage was record with various type of Full-HD IP camera with 4 Million pixels size for one-day long. We set two CCTV camera on the side of container. Because there are 5 spots of the container number on the container: Front side, Back side, Top side and lateral sides. Front side usually concealed by locomotive. On top side also often being soiled. So, there are two spots left back side and lateral side. In this research we choose lateral side because there are two container number on the lateral sides. So, we can cross check the result between two sides. The working flow of the ACNR system (Automatic container number recognition system) is summarized in Fig 1.

This paper divided into three major steps: Real time trigger, Container number Localization, Character Segmentation, and

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

Character Recognition. The first method which is used for real-time container number localization using binarization and finding connected components. The second step is Character Segmentation method which is based on histogram projection method. The last part is character recognition process which is using Convolutional Neural Network.

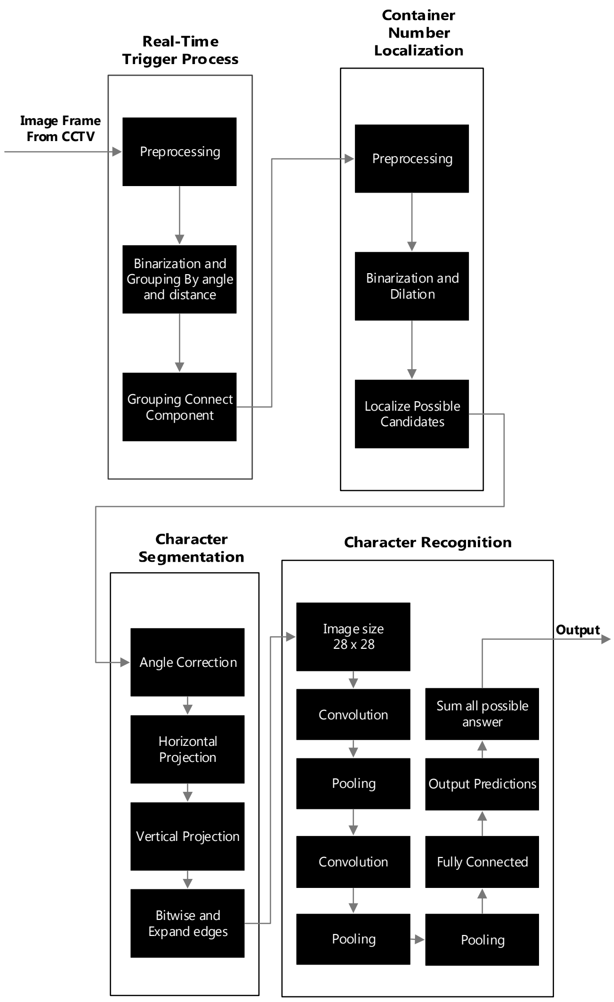

Fig. 1. The working flow of the ACNR system.

A. *Real-time Trigger*

First step is Real-time trigger system, from CCTV IP camera transmitting image frame using Real Time Streaming Protocol, image frame will be processed to grayscale using HSV domain and pull only Value domain. After than we blur image with Gaussian blur buffer filter size 13x13. Then using adaptive threshold, with adaptive block size 55 and adaptive threshold weight 15. After that we find contour of the image in order to find connected region. Every connect regions have to connect to another region with 180 degree (Vertical) or 90 degree (Horizontal). With this algorithm this application can trigger system to start process selected image frame with the next step Container number localization. The results of real-time trigger process are shown in Fig 2.

B. *Container Number Localization*

Second step is localization the container number, by grouping connected component. In this section binarization threshold and adaptive threshold weight will be different from the previous step. First thing to do in this section is to with gray image frame (Value domain in previous section) is to apply Gaussian smooth filter with kernel size 3x3 then binarization image frame with adaptive threshold with adaptive block size 35 and adaptive weight 15. After preprocessing method, we create rectangle kernel which size equal to (1,2) then dilating image frame with iteration equal to 1. After we get a binary image and finding contours of all component in frame, then filtered all candidate with fix aspect ratio consist of area of the contours, width, height and solidity of the component. Then grouping all possible candidate to a group of eleven, group of four or group of seven, according to experimental data all container number have 4 types of possible form: 11 characters horizontal, 11 characters vertical, 2 rows horizontal lines first row consist of 4 characters and second row consist of 7 characters and 3 rows horizontal lines. After grouping all the possible candidates then cut out the region of interest from image frame. The results of Container number localization process are shown in Fig 3.

C. *Character Segmentation*

In third section we apply histogram projection method to segment the candidate image frame. In the preprocessing image is to tilt image to get the perfect angle image. In order to use histogram projection image frame, need to be straight horizontally and vertically. Then we binarized image frame with Otsu's method. Then perform a horizontal projection to select process case whether one row horizontal, two rows horizontal or one row vertical case. After getting segmented characters, we perform bitwise and expand the edges of all character. The results of character segmentation process are shown in Fig 4.

D. *Character Recognition*

In this section we use Convolutional Neural Network as a main optical character recognition engine. This paper apply the two LeNet architectures for optical character recognition process which first introduced in [6]. One LeNet architecture for recognize alphabet (A-Z) and the other one for recognize ten digits (0-9). The LeNet architecture consists of the following layers: Convolutional layer (CONV) => Activation Layer (ACT) => Max Pooling Layer (POOL) => Convolutional layer (CONV) => Activation Layer (ACT) => Max Pooling Layer POOL) => Fully-connected layer (FC) => Activation Layer (ACT) => Fully-connected layer (FC) => Activation Layer (ACT).

TABLE I. SUMMARY OF THE LENET ARCHITECTURE

Layer Type	Output Size	Filter Size
Input Image	28 x 28 x 1	
CONV	28 x 28 x 20	5 x 5, K = 20
ACT	28 x 28 x 20	
POOL	14 x 14 x 20	2 x 2
CONV	14 x 14 x 50	5 x 5, K = 50
ACT	14 x 14 x 50	
POOL	7 x 7 x 50	2 x 2
FC	500	
ACT	500	
FC	26	
SOFTMAX	26	

Table I. summarizes the parameters for the LeNet architecture. Our input layer takes an input image with 28 rows, 28 column, and a single channel (grayscale) for depth. Then learn 20 filters, each of which are 5 x 5. The convolutional layer (CONV) is followed by a ReLU activation (ACT) followed by max pooling with a 2 x 2 size and 2 x 2 stride. The next block of the architecture follows the same pattern, but learning 50 5 x 5 filters. We have two fully-connected layers (FC). The first FC contain 5000 hidden nodes followed by a ReLU activation. The final FC layer controls the number of output class labels (A-Z; one for each of the possible alphabet) in this case is 26. Finally, we apply a SoftMax activation to obtain the class probabilities.

The other architecture, same as the first, only change in the last FC layer from 26 to 10 (for the possible ten digits; 0-9).

Fig. 2. Results of real-time trigger process.

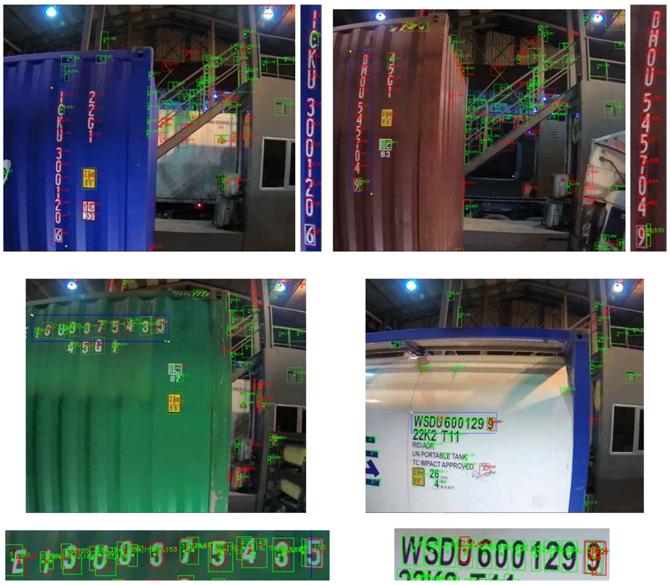

Fig. 3. Results of container number localization process.

(a)

(b)

(c)

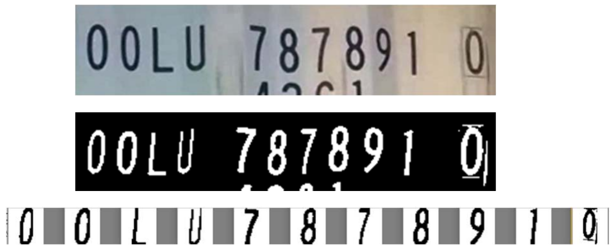

Fig. 4. Result of character segmentation process – (a) Region of Interest of container number; (b) binary image; (c) segmented characters.

III. EXPERIMENTAL RESULTS

In real-time process, the system will process four major methods which are real-time trigger, container number localization, character segmentation, and then character recognition. The containers that unsuccessful in any process will not go on the next process. Each container for several times before showing the result, minimum 3 times before the result and check sum were verify at least 2 times. From an experiment, total samples of 218 containers are tested. All of the defect container such as character were fade or ravel, unable to read were removed.

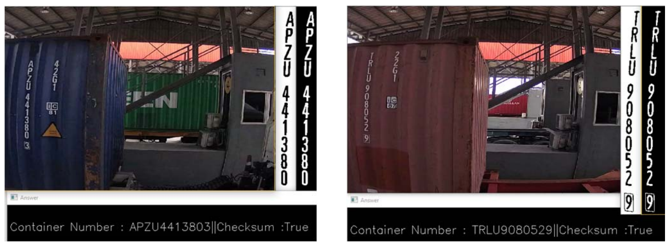

Fig. 5. Results of the ACNR system.

TABLE II. EXPERIMENTAL RESULTS

Process	Correct	Incorrect	Accuracy
Real-Time Trigger process	217	1	99.54%
Container Number Localization	213	4	98.16%
Character Segmentation process	210	3	98.59%
Character Recognition process	209	1	99.52%
Total Process	208	10	95.41%

IV. CONCLUSION

In order to better control and improve efficiency in container number verification process, several image processing techniques have been applied to the proposed automatic container number recognition (ACNR) system which based on computer vision. There are four major processes from the proposed ACNR system. The first one is real-time trigger which can active the rest process to start process the image frame with this method ACNR system can be process real-time with utilization time less than 10 second per one container. The second one is container number localization which combines the candidate character contours and the spatial relationship between successive characters. To segment the characters in the third process, the text-line regions are firstly segmented. Then vertical and horizontal projection is applied to get the position of each character. The last one is character recognition process which adopting LeNet architectures of Convolutional Neural Network.

The proposed algorithm gives accuracy rate of 95.41% with time utilization < 10 seconds per/container and suitable to locate and recognize container numbers of different colors, sizes, and alignment modes.

REFERENCES

[1] V. Abolghasemi, and A. Ahmadyfard, “An edge-based color-aided method for license plate detection,” Image and Vision Computing, vol. 27, pp. 1134–1142, 2009.

[2] G. Li, R. Zeng, and L. Lin, “Research on vehicle license plate location based on neural networks,” The first international conference on innovative computing, information and control, vol. 3, pp. 174–177, 2006.

[3] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, “A license plate-recognition algorithm for intelligent transportation system application,” IEEE Transaction on Intelligent Transportation Systems, vol. 7, pp. 377–392, 2006.

[4] R. Brunelli, “Template Matching Techniques in Computer Vision: Theory and Practice,” Wiley, 2009.

[5] S. Chang, L. Chen, Y. Chung, and S. Chen, “Automatic license plate recognition,” IEEE Transaction on Intelligent Transportation Systems, vol. 5, pp. 42–53, 2004.

[6] C. Coetzee, C. Botha, and D. Weber, “Pc based number plate recognition system,” IEEE international symposium on industrial electronics, pp. 605–610, 1998.

[7] C. Cortes, and V. Vapnik,. “Support-vector networks,” Machine Learning, vol. 20, pp. 273–297, 1995.

[8] D. Deb, S.J. Kang, and K.H. Jo, “Statistical characteristics in HSI color model and position histogram based vehicle license plate detection,” Intelligent Service Robotics, vol. 2, pp. 173–186, 2009.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, pp. 2278-2324, November 1998.