Ship License Numbers Recognition Using Deep Neural Networks

To cite this article: Baolong Liu et al 2018 J. Phys.: Conf. Ser. 1060 012064

View the article online for updates and enhancements.

Related content

- A Horizontal Tilt Correction Method for Ship License Numbers Recognition
  Baolong Liu, Sanyuan Zhang, Zhenjie Hong et al.

- Ship License Number Layout Normalization Based on Regional Texts Fine Localization
  Baolong Liu, Sanyuan Zhang, Zhenjie Hong et al.

- Lymphoscintigraphic imaging study for quantitative evaluation of a small field of view (SFOV) gamma camera
  M.S. Alqahtani, J.E. Lees, S.L. Bugby et al.
Ship License Numbers Recognition Using Deep Neural Networks

Baolong Liu¹, Shuangqing Wu²*, Sanyuan Zhang¹, Zhenjie Hong³ and Xiuzi Ye³
¹College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
²Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China
³College of Mathematics, Physics and Electronic Information Engineering, Wenzhou University, Wenzhou 325035, China
*wsqing1999@163.com

Abstract. Ships are important waterway transportation vehicles. Ship license numbers can be used to identify ships due to their uniqueness. However, rare attentions have been paid to the domain of SLNs recognition. In this paper, a SLNs recognition framework is proposed to recognize multi-style SLNs. First of all, the styles of SLNs are normalized. In this phrase, the characters of SLNs that are written in multiple lines are normalized into one nearly horizontal line. Then, a convolutional recurrent neural network is trained to recognize these characters. A spatial transformer network is configured to rectify distorted SLN images. A sufficient and proper training dataset is a basic prerequisite for a deep neural network to play out its advantages. Therefore, a synthetic SLNs dataset which contains more than 6 million synthetic SLN images were generated to train the convolutional recurrent neural network. This synthetic SLNs dataset was generated based on the image backgrounds, character fonts, character types and text-line angels of the collected real-world SLN images. Finally, the phonetic transcriptions are eliminated. In the experimental period, the Levenshtein distance is configured to estimate the recognition efficiency. Experimental results on 1846 collected real-world SLNs proved the good SLNs recognition ability of the proposed framework.

1. Introduction
With the needs of security surveillance continually growing, many image processing, machine learning and deep learning technics have been applied to the domain of car license plates location and recognition [1], and impressive progresses on car license plates recognition have been achieved also.

Waterway traffic serves as a powerful assurance for the development of economic and commerce. Ship is the most important vehicle in waterway transportation. In some country, e.g., China, ships that are hoped to work legally in public waters need to be registered on local official authorities with a unique name in advance, these unique ship names are also referred as ship license numbers (SLNs). Therefore, SLNs can be treated as unique identities of ships. However, rare attentions have been paid to the domain of SLNs recognition.

One related research area of SLNs recognition is car license plates recognition [1]. Many applications which based on car license plates recognition technics, e.g., parking charge, vehicle entrance control and transportation surveillance, are already a part of everyday life. Even though, there exists no universal method that can recognize different car license plates from different countries,
most car license plates recognition algorithms can only recognize car license plates that are of specific styles. However, SLNs from different ships have various styles [2]. It is thus difficult to use existing car license plates recognition algorithms to proceed SLNs recognition.

Figure 1. Some 0-Levenshtein-distance recognition results of the proposed approach.

Left: the input SLN images. Right: the corresponding recognition results.

In [3], a transferred deep CNN model in combination with prior features is used to localize multi-style ship license numbers. An effective coarse-to-fine approach for locating various SLNs in the wild is proposed by Liu et al. [2]. A horizontal tilt correction method for SLNs recognition is presented in [4].

Recently, deep neural networks-based methods have achieved great progresses on many tasks, e.g. object detection [5] and character recognition [6]. An end-to-end trainable neural network, which named as Convolutional Recurrent Neural Network (CRNN), is presented in [6], it integrates feature extraction, sequence modelling and transcription into a unified framework, and can accomplish character sequence recognition task. However, on the one hand, CRNN can only recognize English alphabet so far. On the other hand, for CRNN, the characters need to be recognized should be nearly listed in one line, hence, CRNN is unable to read characters that are written in multiple lines. It is hard to directly use CRNN to recognize SLNs because multilingual characters may be found in SLNs. For example, SLNs of Chinese cargo ships contain not only English alphabet but also Chinese characters. Moreover, characters of SLNs may be also written in more than one lines. To this end, CRNN is improved in this paper to recognize multi-style SLNs. First of all, SLNs that have different styles are normalized, characters that are written in different lines are normalized into one nearly horizontal line. Then, CRNN is improved to recognize these normalized characters. Specifically, for an input SLN image, the CNN features are computed by the CNN module, a spatial transformer network [7] is inserted after the CNN to rectify distorted SLN images, finally a RNN with long short-term memory (LSTM) mechanism is used to recognize the character sequences contained by the input SLN image. A sufficient and proper dataset is a basic prerequisite for deep neural network to play out its advantages. A synthetic SLNs dataset is thus produced. The dataset contains more than 6 million SLNs images, in which about 5.46 million are used as the training data and the remaining are used as validation data. Several influence factors including SLN image backgrounds, character fonts, character types and text-line angels are taken into consideration when making these synthetic SLN images. Meanwhile, a real-world SLNs dataset is also collected. The networks is firstly trained on the synthetic SLNs dataset, then fine turned on the collected real-world SLNs dataset. Testing experiments on 1846 real-world SLN images is conducted to validate the recognition efficiency of the proposed approach. Figure 1 shows some recognition results of the proposed approach.

2. **The proposed SLNs recognition approach**

Figure 2 shows the SLNs recognition pipeline of the proposed approach. It consists six main parts.
Figure 2. The SLNs recognition pipeline of the proposed approach.

The technical details are described as follows:

(1) SLN layout normalization. Characters that are written in two or more lines are normalized into one line before they are fed to CRNN to recognize. For one-line SLNs, some phonetic transcriptions which comprised by English alphabet are removed, then our horizontal tilt SLN correction method presented in [4] is used to correct the input SLN image; For multiple-lines SLNs, the text region detection method proposed in [8] is fist used to fine localize the texts contained in the SLN image, multiple-line SLNs then are normalized into one line by chaining the detected bounding boxes one by one from top to bottom.

(2) Convolutional feature sequence extraction. The input SLN is resized into 32*256 (height*width) grayscale image. Then, the grayscale image is fed to the convolutional feature sequence extraction module to extract sequential features of the input SLN image.

(3) Spatial transformer network. The spatial transformer networks presented in [7] is inserted after the CNN layers to rectify distorted SLN images. STN is kind of powerful neural network architecture that can achieve spatial invariance by automatically rectifying the input images before they are fed to a normal neural network. STN is end-to-end differential and can be applied to the existing network architectures without extra supervision.

(4) Bidirectional LSTM-based sequence labelling. The extracted convolutional feature sequences are fed to bidirectional LSTM layers to predict a label distribution. Long-Short-Term-Memory (LSTM) [9] is a kind of RNN unit that can alleviate the vanishing gradient problem of RNN [6].

(5) Predictions transcription. The predictions resulted by LSTM are converted into a label sequence in this phrase.

(6) Phonetic transcription elimination. The phonetic transcriptions (also referred as Pinyin), appear in SLNs frequently. Phonetic transcriptions are composed by English alphabet and need to be discarded since they are just some pronunciations.

The network architectures of step 2, 4 and 5 are kept same with that in CRNN, please refer [6] for more details.

3. Experimental results

3.1 Dataset

(1) Synthetic SLNs dataset. No large SLNs training dataset is published so far, a synthetic dataset is generated in this paper to handle the problem of SLNs training data insufficient. These synthetic SLNs data were generated based on the image backgrounds, character fonts, character types and text-line angles of the collected real-world SLN images. Fifty images that have different RGB value are used as the basic background images. Twenty one kinds of Chinese fonts that may appear in real-world SLN images are used as the basic character fonts. Characters that appeared in real-world SLNs are used to generate corpuses, the character appearance frequencies were also taken into consideration. Random noises were added. More than 6 million synthetic SLN images were generated, in which 5,468,671 synthetic SLN images were used as the training data, the remaining were used as the validation data.
Each synthetic SLN image has a ground truth label. Figure 3 demonstrates some generated synthetic SLNs.

(2) Real-world SLNs dataset. 10,975 SLNs images with their reverse images (totally 21,950) are collected as the training dataset. Another 1846 SLNs images are used as testing instances. These SLN images were collected in the practical application environment of the proposed framework — Beijing Hangzhou the Grand Canal (Hangzhou).

![Some samples of the generated synthetic SLNs.](image)

3.2 Implement details
Experiments were carried out on a workstation with a 3.50 GHz Intel(R) Core(R) i7-5930 CPU, 64GB RAM and an NVIDIA(R) GeForce GTX 1080 GPU. All images were scaled to 256*32 (width*height) in order to accelerate the training process. Networks were implemented using PyTorch, and trained with RMSProp. The networks were firstly trained on synthetic dataset for 160k iterations (learning rate was set to $10^{-2}$), then fine-tuned on the real-world SLNs dataset for 10k iterations (learning rate of $10^{-7}$).

![Some recognition results of (a) 1-Levenshtein-distance and (b) 2-Levenshtein-distance. Characters with red color were missed or mistakenly recognized.](image)

3.3 Experimental results
For an input SLN image, the corresponding recognition output is a sequence of characters. The Levenshtein distance (Edit distance) between the recognized characters and the ground truth labels are computed to evaluate the recognition accuracy. The smaller the Levenshtein distance is, the more approximate the recognized outputs and the ground truth label are. The Levenshtein distance between a recognized sequence and a ground truth label is equal to 0 represents that the output sequence is totally the same with the corresponding ground truth.

In our experiments, recognition accuracies are computed under the conditions of different Levenshtein distance thresholds. For example, the Levenshtein distance threshold is set to 1 means
that, those character sequences whose Levenshtein distances are less than or equal to 1 are regarded as corrections. 1460 in Table 1 means that there are 1460 resulted sequences whose Levenshtein distances are 0.

**Table 1. Recognition results of SLNs**

| Levenshtein distances | 0   | 1       | 2       | Greater than 2 |
|-----------------------|-----|---------|---------|----------------|
| Number of recognized instances | 1460 | 204     | 118     | 64             |
| Overall accuracy      | 79.09% | 90.14% | 96.53% | 100%           |

Table 1 demonstrates the overall recognition accuracies of the proposed approach on 1846 real-world SLNs images under the Levenshtein distance threshold of 0, 1 and 2. Table 1 suggests that the proposed approach achieved a SLNs recognition accuracy of 79.09% under the condition of 0 Levenshtein distance, a recognition accuracy of 90.14% under the condition of 1 Levenshtein distance and a recognition accuracy of 96.53% under the condition of 2 Levenshtein distance. Table 1 proves the effectiveness of the proposed SLNs recognition framework. Figure 1 shows some 0-Levenshtein-distance recognition results. From Figure 1 one can recognizes the good recognition ability of the proposed approach. Figure 4 shows some 1-Levenshtein-distance and 2-Levenshtein-distance recognition results, from which one can see some characters of SLNs were missed or mistakenly recognized (characters with red color), the reason is that these characters are in lower resolutions or are interfered.

4. Conclusions

In this paper, a SLNs recognition framework have been proposed to recognize multi-style SLNs. CRNN is improved to recognize SLNs. A synthetic SLNs dataset which contains more than 6 million synthetic SLN images were generated. Experimental results on 1846 collected real-world SLNs proves the good SLNs recognition efficiency of the proposed framework.

Acknowledgments

This work was supported partly by National Natural Science Foundation of China (61272304 and 61772374), National Key Research Project “Cloud Computing & Big Data” of China (2017YFB1002600), Ningbo Natural Science Foundation (2017A610108), and Zhejiang Provincial Natural Science Foundation (LY16F020023, LQ18F030010 and LQ17F030002).

References

[1] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic license plate recognition (alpr): A state-of-the-art review,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 2, pp. 311–325, 2013.

[2] B. Liu, J. Sheng, J. Dun, S. Zhang, Z. Hong, and X. Ye, “Locating various ship license numbers in the wild: An effective approach,” IEEE Intelligent Transportation Systems Magazine, vol. 9, no. 4, pp.102–117, 2017.

[3] B. Liu, X. Lyu, C. Li, S. Zhang, Z. Hong, and X. Ye, “Using transferred deep model in combination with prior features to localize multi-style ship license numbers in nature scenes,” in IEEE International Conference on TOOLS with Artificial Intelligence, 2018, pp. 506–510.

[4] B. Liu, S. Zhang, Z. Hong, and X. Ye, “A Horizontal Tilt Correction Method for Ship License Numbers Recognition,” in Journal of Physics: Conf. Series, vol. 976, pp. 012013, 2018.

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., & Fu, C. Y., et al. “SSD: Single Shot MultiBox Detector,” in European Conference on Computer Vision, 2016, pp. 21-37.

[6] 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," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298- 2304, Nov. 1 2017.
[7] Jaderberg M, Simonyan K, Zisserman A, “Spatial transformer networks,” in *Advances in Neural Information Processing Systems*. 2015: 2017-2025.

[8] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, “Single Shot Text Detector with Regional Attention,” in *IEEE International Conference on Computer Vision*, 2017, pp. 3066-3074.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 3