

Robust Recognition of Chinese Text from Cellphone-acquired Low-quality Identity Card Images Using Convolutional Recurrent Neural Network

Jianmei Wang, Ruize Wu, and Shaoming Zhang*

College of Surveying and Geo-informatics, Tongji University, 1239 Siping Road, Shanghai 200092, China

(Received July 23, 2020; accepted January 6, 2021; online published January 25, 2021)

Keywords: Chinese text recognition, synthetic data, convolutional recurrent neural network, conditional generative adversarial network, DenseNet

An automatic reading of text from an identity (ID) card image has a wide range of social uses. In this paper, we propose a novel method for Chinese text recognition from ID card images taken by cellphone cameras. The paper has two main contributions: (1) A synthetic data engine based on a conditional adversarial generative network is designed to generate million-level synthetic ID card text line images, which can not only retain the inherent template pattern of ID card images but also preserve the diversity of synthetic data. (2) An improved convolutional recurrent neural network (CRNN) is presented to increase Chinese text recognition accuracy, in which DenseNet substitutes VGGNet architecture to extract more sophisticated spatial features. The proposed method is evaluated with more than 7000 real ID card text line images. The experimental results demonstrate that the improved CRNN model trained only on the synthetic dataset can increase the recognition accuracy of Chinese text in cellphone-acquired low-quality images. Specifically, compared with the original CRNN, the average character recognition accuracy (CRA) is increased from 96.87 to 98.57% and the line recognition accuracy (LRA) is increased from 65.92 to 90.10%.

1. Introduction

Identity (ID) cards are a kind of legal certificate to prove the residential ID of the holder in China and are widely used in all aspects of modern social life. It is necessary to input ID card information when doing business involving the government, public security, banking, securities, insurance, taxation, and so forth. Manually inputting ID card information is inefficient and prone to errors, and it is not possible to input unknown words. It will greatly improve the work efficiency and service level if ID card information can be read automatically.

The most commonly used device for reading ID card information is an ID card reader, which is based on the induction principle of magnetic cards and RFID technology. The device can quickly and accurately read the information stored in a second-generation ID card chip, but this technology needs close contact between the ID card and the card reader. In recent years, with the integration of internet technology and traditional industries in China, doing business online

*Corresponding author: e-mail: 08053@tongji.edu.cn
<https://doi.org/10.18494/SAM.2021.2991>

is becoming increasingly popular. To read customer information automatically from uploaded ID card images, optical character recognition (OCR) technology is needed.

OCR refers to using optical equipment to obtain images containing text, and then converting the text in the image into computer-readable and editable character codes through digital image processing and pattern recognition methods. ID card images can be obtained from a scanner or a camera. The OCR technology applied to scanned ID card images is mature, and the recognition accuracy has reached over 99%.⁽¹⁾ However, recognizing text from camera-acquired low-quality ID card images is still a challenging task.

OCR can be divided into two stages: text detection and text recognition. The goal of text detection is to produce segmentations or bounding boxes of texts in the whole image, while text recognition aims at converting a cropped text image to text strings. This paper only focuses on text recognition. There are Chinese characters, English letters, numbers, and punctuations on an ID card, which are printed horizontally in lines. A convolutional recurrent neural network (CRNN) is the most popular model for recognizing regular texts owing to its capability of acquiring competitive results with relatively few parameters.⁽²⁾ The architecture of a CRNN consists of three components: convolutional layers, recurrent layers, and a transcription layer. The convolutional layers automatically extract a sequence of features from each input image, the recurrent layers predict a label distribution for each frame in the feature sequence, and the transcription layer translates the per-frame predictions into the final label sequence.⁽³⁾ However, the CRNN was originally designed for English character recognition. Compared with the 52 English characters (i.e., 26 lower-case and 26 upper-case letters), there are thousands of Chinese characters including more than 6000 commonly used ones. Furthermore, many Chinese characters appear similar, e.g., “日” and “曰”, “土” and “士”, and “治” and “冶”. These differences call for a more complicated model to extract more sophisticated structure features to recognize Chinese characters. The first contribution of this paper is that a novel convolutional neural network (CNN) is introduced into a CRNN model to replace the original convolutional layers for the extraction of more sophisticated structure features in Chinese characters.

The supervised training of a large model such as a CRNN, which contains millions of parameters, requires a very large amount of labeled training data. Owing to the privacy associated with ID cards, it is impossible to build a large-scale training dataset consisting of real ID card images except for public security organizations. Synthetic datasets provide detailed ground-truth annotations, which are cheap and scalable alternatives to annotating images manually. They have been widely used to learn scene text recognition models^(4,5) and scene text detection models.⁽⁶⁾ The second contribution of this paper is that a novel ID card text image generator (G) based on a conditional generative adversarial network (cGAN) named pix2pix⁽⁷⁾ is proposed, which is capable of emulating ID card text images in a natural environment in the case of a small number of real ID card images.

2. Related Works

Text recognition methods can be broadly divided into three categories: character-based, word-based, and sequence-based methods.

Character-based recognition methods generally consist of three steps: character detection, character recognition, and character combination. Wang *et al.* used random ferns and a histogram of oriented gradient (HOG) features to detect characters, then found an optimal configuration of a particular word via a pictorial structure.⁽⁸⁾ Mishra *et al.* detected character candidates using sliding windows and integrated both bottom-up and top-down cues in a unified conditional random field (CRF) model.⁽⁹⁾ Bissacco *et al.* used a neural network classifier acting on the HOG features of the segments as scores to find the best combination of segments using beam search.⁽¹⁰⁾ Jaderberg *et al.* used a combination of a binary text/no-text classifier, a character classifier, and a bigram classifier densely computed across the word image as cues to a Viterbi scoring function in the context of a fixed lexicon.⁽¹¹⁾ Character-based recognition methods require robust and accurate character detection and recognition, otherwise the word alignment will lead to incorrect results due to error accumulation from lower to higher levels.

Word-based recognition methods treat each word image as a whole without requiring character detection and recognition. Goel *et al.* converted the word recognition task into a problem of retrieving the best match from a lexicon image set with a weighted dynamic time warping approach.⁽¹²⁾ Almazán *et al.* embedded word images and word labels into a common Euclidean space, and used embedding vectors to match images and labels.⁽¹³⁾ Jaderberg *et al.* treated text recognition as an image classification problem. Each class corresponded to one English word in a pre-defined large dictionary composed of around 90k words.⁽¹⁴⁾ However, lexicon-driven word recognition methods lack flexibility and cannot recognize a rarely occurring word that is not included in the lexicon.

Sequence-based recognition methods regard text recognition as an image-based sequence recognition problem, where images and texts are separately encoded as patch and character sequences. Su and Lu extracted a sequential image representation, which is a sequence of HOG descriptors, and predicted the corresponding character sequence with a recurrent neural network (RNN).⁽¹⁵⁾ Shi *et al.* proposed an end-to-end neural network architecture that combined CNN and RNN for visual feature representation, then the connectionist temporal classification (CTC) loss⁽¹⁶⁾ was combined with the RNN outputs to calculate the conditional probability between the predicted and target sequences.⁽³⁾ Inspired by the sequence-to-sequence framework for machine translation,⁽¹⁷⁾ Lee and Osindero used a recursive RNN to learn broader contextual information and applied an attention-based decoder for sequence generation.⁽¹⁸⁾ Cheng *et al.* proposed a focus mechanism to eliminate the attention drift to improve the recognition performance of regular text.⁽¹⁹⁾ Bai *et al.* proposed an edit probability metric to handle the misalignment between the ground-truth string and the attention's output sequence of a probability distribution.⁽²⁰⁾ Both CTC and encoder–decoder frameworks were originally designed for 1D sequential input data, and therefore applied to the recognition of straight and horizontal text, which can be encoded into a sequence of feature frames without losing important information. In contrast to CTC, the decoder module of the encoder–decoder framework is an implicit language model, so it can incorporate more linguistic priors. For the same reason, the encoder–decoder framework requires a larger training dataset with a larger vocabulary. Otherwise, the model may degenerate when reading words that are not seen during training. In contrast, CTC is less dependent on language models and has a better character-to-pixel alignment. Therefore, it is potentially better on languages such as Chinese and Japanese that have a large character set.⁽²⁾

3. Synthetic ID Card Text Line Image

There is a standard template for Chinese ID cards, such as the font, size, spacing, and color. We construct a corpus based on the content of Chinese ID cards. So that the corpus is similar to the Chinese ID card text distribution, the text of names is randomly selected from a Chinese name corpus,⁽²¹⁾ the text of addresses comes from a random combination of the different levels of administrative divisions in a China area corpus,⁽²²⁾ and the texts of gender, nationality, date of birth, and ID card number are randomly selected from their value domains. A punctuation mark is inserted between texts of different contents. Some uncommon characters are supplemented to mitigate the problem of imbalanced samples.

The process of generating a synthetic ID card text line image is shown in Fig. 1. Ten consecutive characters are extracted from any position in the corpus to generate a binary text line image with size 32×280 . Next, the binary text line image is distorted with a random, full perspective transformation, simulating the 3D world. Because the input image size of G is fixed at 256×256 , the binary text line image and its seven duplicates are mosaicked into one image and resized to 256×256 . The synthetic ID card text image output from G is split into eight identical sub-images from top to bottom, one of which is selected as the synthetic ID card text line image and resized to 32×280 . Finally, Gaussian noise, out-of-focus blur, and so forth, are added to the synthetic ID card text line image with random intensity.

We use a cGAN named pix2pix⁽⁷⁾ to train G to learn mapping from binary text images to ID card text images. The process of training G is shown in Fig. 2. G learns to translate binary text images x to synthetic ID card text images $G(x)$ that cannot be distinguished from the corresponding real ID card text images y by an adversarially trained discriminator (D), while simultaneously D learns to classify between fake $\{G(x), x\}$ and real $\{y, x\}$.

The objective of the pix2pix network can be expressed as

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G), \quad (1)$$

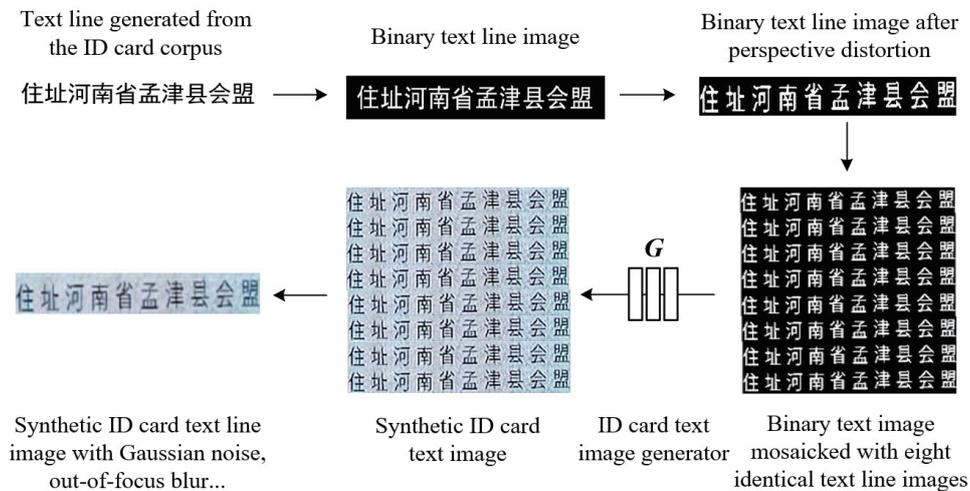


Fig. 1. (Color online) Proposed synthetic ID card text line image generation process.

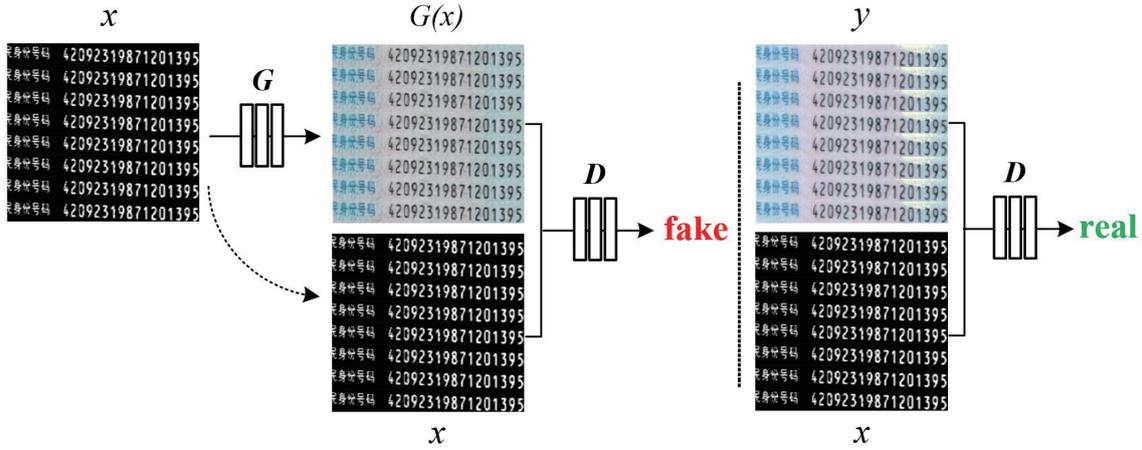


Fig. 2. (Color online) Process of training G based on pix2pix⁽⁷⁾. G learns to translate binary text images x to synthetic ID card text images $G(x)$ that cannot be distinguished from the corresponding real ID card text images y by D , while simultaneously D learns to classify between fake $\{G(x), x\}$ and real $\{y, x\}$.

where $\mathcal{L}_{cGAN}(G, D)$ representing the adversarial loss of the cGAN and $\mathcal{L}_{L1}(G)$ representing the L1 distance loss are expressed as

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x, y} [\log D(x, y)] + \mathbb{E}_x \log [1 - D(x, G(x))], \quad (2)$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x, y} [\|y - G(x)\|_1]. \quad (3)$$

λ in Eq. (1) controls the relative importance of the two objectives. We set λ to 100 to encourage the output of G to be less blurring. Figure 3 illustrates some samples generated by G , which is trained with 613 real ID card text images and their corresponding binary text images.

4. Improved CRNN

4.1 CRNN architecture⁽³⁾

A CRNN is an end-to-end training neural network for image-based sequence recognition, whose architecture consists of three components: convolutional layers, recurrent layers, and a transcription layer from bottom to top, as shown in Fig. 4. The convolutional layers automatically extract a feature sequence $\mathbf{x} = x_1, \dots, x_T$ from each input image, where T is the sequence length. The recurrent layers predict a label distribution y_t for each frame x_t . The transcription layer converts the per-frame predictions $\mathbf{y} = y_1, \dots, y_T$ into a label sequence \mathbf{l} . Mathematically, transcription is finding the label sequence \mathbf{l} that maximizes $P(\mathbf{l}|\mathbf{y})$, where $P(\mathbf{l}|\mathbf{y})$ is defined in the CTC layer proposed by Graves *et al.*⁽¹⁶⁾

We denote the training dataset by $\mathcal{X} = \{I_i, \mathbf{l}_i\}_i$, where I_i is the training image and \mathbf{l}_i is the ground-truth label sequence. The objective is to minimize the negative log-likelihood of the conditional probability of ground truth:

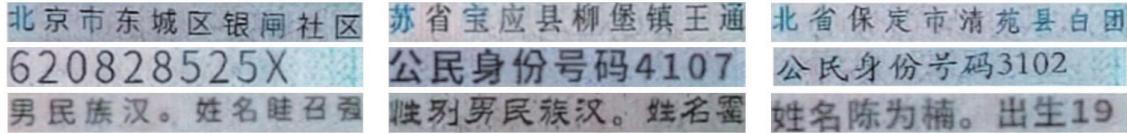


Fig. 3. (Color online) Samples of synthetic ID card text line images.

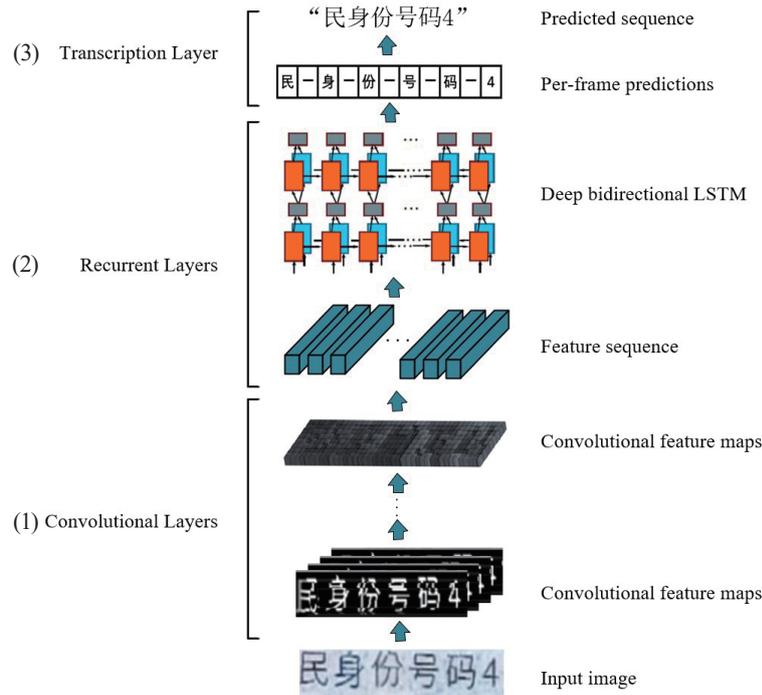


Fig. 4. (Color online) CRNN architecture. The architecture consists of three parts: (1) convolutional layers, which extract a feature sequence from the input image; (2) recurrent layers, which predict a label distribution for each frame; (3) a transcription layer, which translates the per-frame predictions into the final label sequence.

$$\mathcal{O} = - \sum_{I_i, I_i \in \mathcal{X}} \log p(I_i | y_i), \quad (4)$$

where y_i is the sequence produced from I_i by the recurrent and convolutional layers. The objective function calculates a cost value directly from an image and its ground-truth label sequence, so the network can be end-to-end trained on pairs of images and sequences.

4.2 Improved feature sequence extraction module

The CRNN was originally designed for English character recognition, and its architecture of convolutional layers is based on the VGG-VeryDeep architecture,⁽²³⁾ which is prone to losing fine spatial features. Compared with English characters, Chinese characters have a more sophisticated spatial structure and a more similar appearance. To improve Chinese

text recognition accuracy, a novel feature sequence extraction module based on a dense convolutional network (DenseNet) architecture⁽²⁴⁾ is proposed in this paper as shown in Fig. 5. Figure 6 illustrates a five-layer dense block with a growth rate of $k = 4$. We assume that the network comprises L layers, each of which implements a nonlinear transformation $H_\ell(\cdot)$, where ℓ indexes the layer and $H_\ell(\cdot)$ is a composite function of three consecutive operations: batch normalization (BN),⁽²⁵⁾ a rectified linear unit (ReLU),⁽²⁶⁾ and a 3×3 convolution (Conv). We denote the input feature map as x_0 and the output of the ℓ th layer as x_ℓ . Then,

$$x_\ell = H_\ell([x_0, x_1, \dots, x_{\ell-1}]), \quad (5)$$

where $[x_0, x_1, \dots, x_{\ell-1}]$ refers to the concatenation of the feature maps produced in layers 0, 1, ..., $\ell - 1$. If each function $H_\ell(\cdot)$ produces k feature maps, it follows that the ℓ th layer has $\kappa_0 + k \times (\ell - 1)$ input feature maps, where κ_0 is the number of channels in the input layer. The transition layers consist of a BN layer and a 1×1 convolutional layer followed by a 2×2 average pooling layer.

The proposed feature sequence extraction module has three dense blocks, with each block having eight layers. Before entering the first dense block, a convolution with 64 output channels is performed on input images. For convolutional layers with kernel size 3×3 , each side of the inputs is zero-padded by one pixel to keep the feature map size fixed. A 1×1 convolution followed by 2×2 average pooling is used as the transition layer between two contiguous dense

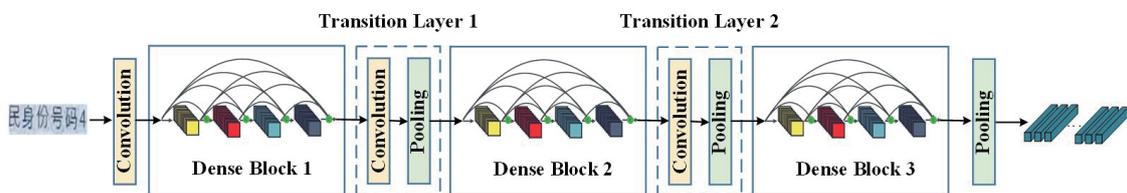


Fig. 5. (Color online) Schematic diagram of proposed feature sequence extraction module.

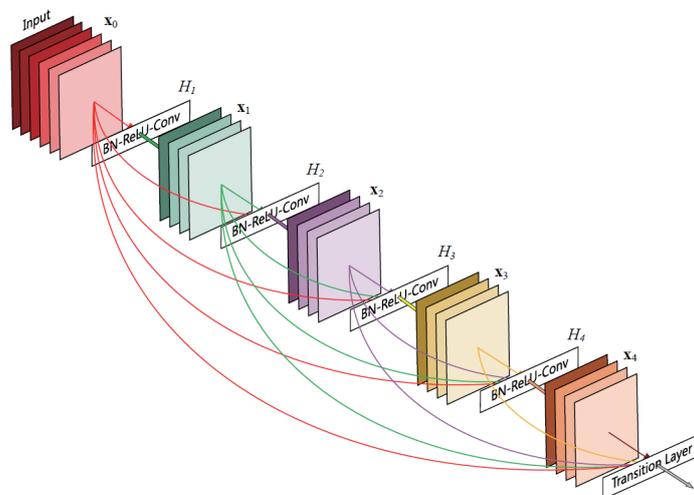


Fig. 6. (Color online) Five-layer dense block with growth rate of $k = 4$.⁽²⁴⁾

blocks. At the end of the last dense block, 4×1 average pooling is performed to extract the feature sequence. We set the text line image size to 32×280 and the growth rate to $k = 8$. The exact network configuration is shown in Table 1.

5. Experimental Results

5.1 Implementation details

Approximately 2000 real ID card images taken by cellphone camera are provided by a construction company under a privacy agreement that prohibited us from revealing the full information of any individual. These images are taken from diverse angles and distances under various lighting conditions by different cellphone brands and models. We cut them into text line images, 613 of which are used to train the pix2pix network and 7824 are used to evaluate the performance of the improved CRNN. Experiments are carried out on a workstation with a 3.4 GHz Intel i7-330 CPU, 16 GB RAM, and an 8 GB NVIDIA GTX 1080 GPU.

The pix2pix network is implemented in TensorFlow 1.2.0. The optimization method is Adam with a learning rate of 0.0002 and momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The batch size is set to 1. The maximum number of iterations is set to 100k. After 90k iterations, $\mathcal{L}_{L1}(G)$ becomes less than 0.1. When the training is done, the method proposed in Sect. 2 is used to generate synthetic ID card text line images. The synthetic dataset contains 6.6 million images covering 7265 types of characters in total.

The improved CRNN is implemented in Caffe with CUDA 8.0 and cuDNN 5.6. The optimization method is Nesterov with a learning rate of 0.0001, a momentum of 0.9, and γ of 0.5. The batch size is set to 64. To verify that the synthetic data is sufficiently realistic to substitute for real data, we only use the synthetic data for training and real data for testing. The training process takes about 20k iterations to reach convergence. Test images are scaled to a height of 32, and the image width is proportionally scaled with height. The image width is at least 280 pixels, and we apply zero-padding for short images.

Table 1
Configuration of proposed feature sequence extraction module. The growth rate is $k = 8$. Note that each ‘‘Conv’’ layer shown in the table corresponds to the sequence BN-ReLU-Conv.

Layer	Output size	Configuration
Convolution	$16 \times 140 \times 64$	5×5 Conv, stride 2
Dense block 1	$16 \times 140 \times 128$	$\begin{bmatrix} 1 \times 1 \text{ Conv} \\ 3 \times 3 \text{ Conv} \end{bmatrix} \times 8$
Transition layer	$16 \times 140 \times 128$	1×1 Conv
	$8 \times 70 \times 128$	2×2 average pooling, stride 2
Dense block 2	$8 \times 70 \times 192$	$\begin{bmatrix} 1 \times 1 \text{ Conv} \\ 3 \times 3 \text{ Conv} \end{bmatrix} \times 8$
Transition layer	$8 \times 70 \times 192$	1×1 Conv
	$4 \times 35 \times 192$	2×2 average pooling, stride 2
Dense block 3	$4 \times 35 \times 256$	$\begin{bmatrix} 1 \times 1 \text{ Conv} \\ 3 \times 3 \text{ Conv} \end{bmatrix} \times 8$
Pooling	$1 \times 35 \times 256$	4×1 average pooling

5.2 Results

Figure 7 shows samples correctly recognized in the test dataset. It can be seen from Fig. 7 that even if the text line images are affected by noise, blur, uneven illumination, perspective distortion, a complex background, and so forth, the improved CRNN can still accurately recognize the text in the images and maintain good robustness.

To further verify the effectiveness of the improved CRNN, we compare it with the original CRNN in quantitative and qualitative analyses. We use two metrics to quantitatively evaluate the recognition performance: (1) the average character recognition accuracy (CRA) based on the longest common subsequence (LCS), defined as

$$CRA = \text{length}(LCS(l, \hat{l})) / \text{length}(l), \quad (6)$$

where \hat{l} represents the predicted label sequence and l represents the ground-truth label sequence; (2) line recognition accuracy (LRA), i.e., the percentage of text line images correctly recognized, where the text line image is correctly recognized if no character is misidentified. Table 2 shows the text recognition accuracies of the improved and original CRNNs. Compared with the original CRNN, CRA is increased from 96.87 to 98.57% and LRA is increased from 65.92 to 90.10% for the improved CRNN. Table 3 lists some images with different recognition results. From the qualitative perspective, the improved CRNN can correctly recognize easily confused Chinese characters in the case of a complex background (a and b in Table 3) and a slanting text line (c and d) owing to higher feature extraction capabilities than the original CRNN. To analyze the shortcomings of the improved CRNN, we list some incorrectly recognized samples



Fig. 7. (Color online) Correctly recognized samples in test dataset with improved CRNN. The text below each image is the recognition result.

Table 2
Comparison of text recognition accuracy between original and improved CRNNs.

Method	CRA	LRA
Original CRNN	96.87%	65.92%
Improved CRNN	98.57%	90.10%

Table 3

(Color online) Different text recognition results obtained with original and improved CRNNs. The underlined characters are incorrectly recognized.

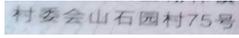
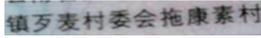
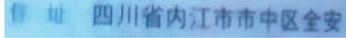
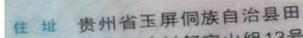
No.	Test image	Original CRNN	Improved CRNN
a		性别男民族汉	性别男民族汉
b		住址海南省儋州市木棠镇薛宅	住址海南省儋州市木棠镇薛宅
c		村委会山石园村 75 号	村委会山石园村 75 号
d		镇歹麦村委会拖康素村	镇歹麦村委会拖康素村

Table 4

(Color online) Incorrectly recognized samples in test dataset with improved CRNN. The underlined characters are incorrectly recognized.

No.	Test image	Ground truth	Improved CRNN
a		姓名胡昌云	<input type="checkbox"/> <input type="checkbox"/> 胡昌云
b		住址四川省内江市市中区全	<input type="checkbox"/> <input type="checkbox"/> 四川省内江市市中区全
c		姓名莫方森	<input type="checkbox"/> <input type="checkbox"/> 莫方春
d		住址贵州省玉屏侗族自治县田	住址勒现融自 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

in Table 4. It is easy to see from the images in Table 4 that recognition errors are mainly caused by low resolution (a and b), out-of-focus blur (c), and character interference (d). Thus, we still need to design a much finer network structure that can extract fine-grained features.

6. Conclusions

In this paper, we propose a novel CRNN for Chinese character recognition from ID card images taken by cellphone cameras that integrates the advantages of both the CRNN architecture and the DenseNet architecture. The CRNN is capable of taking input images of various dimensions and produce predictions with different lengths. It directly runs on coarse level labels, requiring no detailed annotations for each individual element in the training phase. DenseNet allows feature reuse throughout the networks and can consequently learn more compact and accurate internal representations. We have also designed a synthetic data engine based on a conditional adversarial generative network to generate million-level synthetic ID card text line images, which can not only retain the inherent template pattern of ID card images, but also preserve the diversity of synthetic data. We evaluate the performance of the proposed method with more than 7000 real ID card text line images, and the experimental results demonstrate that the improved CRNN model trained only on the synthetic dataset can increase the recognition accuracy of Chinese text in cellphone-acquired low-quality images. Specifically, compared with the original CRNN, the average CRA is increased from 96.87 to 98.57% and the LRA is increased from 65.92 to 90.10%. The proposed Chinese text recognition method has been used to read personal information from cellphone-acquired ID card images in an employee management system of a construction company that adopts manual interaction to ensure the

accuracy of the input information. For the ID card images whose quality does not meet the requirements, the administrator will return them to the users for resubmission.

Acknowledgments

This work is sponsored by the National Key Research and Development Program of China (2018YFB0503005, 2018YFB0505400). We thank the two anonymous reviewers for their comments that improved this paper.

References

- 1 Q. Huo: *Sci. Modernization* **66** (2016) 14. <http://www.modernization.ac.cn/upload/filedoc/20180504/FD201805040957052262.pdf>
- 2 S. Long, X. He, and C. Yao: *Int. J. Comput. Vision* **22** (2020) 143. <https://doi.org/10.1007/s11263-020-01369-0>
- 3 B.G. Shi, X. Bai, and C. Yao: *IEEE Trans. Pattern Anal. Mach. Intell.* **39** (2017) 2298. <https://doi.org/10.1109/TPAMI.2016.2646371>
- 4 T. Wang, D. J. Wu, A. Coates, and A. Y. Ng: *Proc. 21st Int. Conf. Pattern Recognition (IEEE, 2012)* 3304–3308. <https://ieeexplore.ieee.org/document/6460871>
- 5 M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman: <https://arxiv.org/abs/1406.2227> (accessed March 2020).
- 6 A. Gupta, A. Vedaldi, and A. Zisserman: *2016 IEEE Conf. Comput. Vision and Pattern Recognit. (IEEE, 2016)* 2315–2324. <https://doi.org/10.1109/CVPR.2016.254>
- 7 P. Isola, J. Zhu, T. Zhou, and A. Efros: <https://arxiv.org/abs/1611.07004> (accessed March 2020).
- 8 K. Wang, B. Babenko, and S. Belongie: *2011 Int. Conf. Comput. Vision (IEEE, 2011)* 1457–1464. <https://doi.org/10.1109/ICCV.2011.6126402>
- 9 A. Mishra, K. Alahari, and C. Jawahar: *2012 IEEE Conf. Comput. Vision and Pattern Recognit. (IEEE, 2012)* 2687–2694. <https://doi.org/10.1109/CVPR.2012.6247990>
- 10 A. Bissacco, M. Cummins, Y. Netzer, and H. Neven: *2013 IEEE Int. Conf. Comput. Vision (IEEE, 2013)* 785–792. <https://doi.org/10.1109/ICCV.2013.102>
- 11 M. Jaderberg, A. Vedaldi, and A. Zisserman: *Lect. Notes Comput. Sci.* **8692** (2014) 512. https://doi.org/10.1007/978-3-319-10593-2_34
- 12 V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar: *2013 12th Int. Conf. Doc. Anal. and Recognit. (IEEE, 2013)* 398–402. <https://doi.org/10.1109/ICDAR.2013.87>
- 13 J. Almazán, A. Gordo, A. Fornés, and E. Valveny: *IEEE Trans. Pattern Anal. Mach. Intell.* **36** (2014) 2552. <https://doi.org/10.1109/TPAMI.2014.2339814>
- 14 M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman: *Int. J. Comput. Vision* **116** (2016) 1. <https://arxiv.org/abs/1412.1842>
- 15 B. Su and S. Lu: *Lect. Notes Comput. Sci.* **9003** (2014) 35. https://doi.org/10.1007/978-3-319-16865-4_3
- 16 A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber: *Proc. 23rd Int. Conf. Mach. Learn. (ACM, 2006)* 369–376. <https://doi.org/10.1145/1143844.1143891>
- 17 D. Bahdanau, K. Cho, and Y. Bengio: <https://arxiv.org/abs/1409.0473> (accessed March 2020).
- 18 C. Y. Lee and S. Osindero: *2016 IEEE Conf. Comput. Vision and Pattern Recognit. (IEEE, 2016)* 2231–2239. <https://doi.org/10.1109/CVPR.2016.245>
- 19 Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou: *2017 IEEE Int. Conf. Comput. Vision (IEEE, 2017)* 5076–5084. <https://doi.org/10.1109/ICCV.2017.543>
- 20 F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou: *2018 IEEE Conf. Comput. Vision and Pattern Recognit. (IEEE, 2018)* 1508–1516. <https://doi.org/10.1109/CVPR.2018.00163>
- 21 <https://github.com/wainshine/Chinese-Names-Corpus> (accessed June 2020).
- 22 https://github.com/adyliu/china_area (accessed June 2020).
- 23 K. Simonyan and A. Zisserman: <https://arxiv.org/abs/1409.1556> (accessed March 2020).
- 24 G. Huang, Z. Liu, L. Maaten, and K. Q. Weinberger: *IEEE Conf. Comput. Vision and Pattern Recognit. (IEEE, 2017)* 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
- 25 S. Ioffe and C. Szegedy: <https://arxiv.org/abs/1502.03167> (accessed March 2020).
- 26 X. Glorot, A. Bordes, and Y. Bengio: *J. Mach. Learn. Res.* **15** (2011) 315. https://www.researchgate.net/profile/Antoine_Bordes/publication/215616967

About the Authors

Jianmei Wang received her B.S. degree in surveying and mapping engineering and her M.S. degree in cartography and geographic information engineering from Tongji University, Shanghai, China, in 1994 and 1997, respectively, and her Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2007. Since 1999, she has been a lecturer at Tongji University. Her research interests are in computer vision, spatial data mining, and remote sensing image analysis. (97031@tongji.edu.cn)

Ruize Wu received his B.S. and M.S. degrees in surveying and mapping engineering from Tongji University, Shanghai, China, in 2017 and 2020, respectively. He is now an algorithm engineer for Alibaba. His research interests are in recommendation systems and computational advertising. (kezhui.wrz@alibaba-inc.com)

Shaoming Zhang received his B.S. degree in electronics engineering from Tianjin University, Tianjin, China, in 2002, his M.S. degree in communication engineering from the 14th Electronics Institute of the Ministry of Information Industry in 2005, and his Ph.D. degree in photogrammetry and remote sensing from Tongji University, China, in 2008. From 2008 to 2014, he was a lecturer at Tongji University. Since 2014, he has been an associate professor at Tongji University. His research interests are in deep learning, computer vision, and SLAM. (08053@tongji.edu.cn)