A Study of RNN Based Online Handwritten Uyghur Word Recognition Using Different Word Transcriptions

Wujiahemaiti Simayi, Mayire Ibrayim, and Askar Hamdulla(✉)

Institute of Information Science and Engineering,
Xinjiang University, Urumqi, China
askar@xju.edu.cn

Abstract. Recurrent neural networks-RNN based online handwriting Uyghur word recognition experiments are conducted applying connectionist temporal classification in this paper. Handwritten trajectory is fed to the network without explicit or implicit character segmentation. The network is trained to transcribe the input word trajectory to a string of characters directly. According to the writing characteristics of Uyghur, experiments are designed using two Unicode word transcriptions respectively based on 32+2 basic character types and 128 specific character forms to represent a word. The training process and recognition results based on same network architecture show that both transcription methods are applicable. The word transcription system using basic 34 character types showed better performance than the one using 128 specific character forms in our experiments. 13.96%, 14.73% character error rates (CER) have been observed respectively for char34 system and char128 system.

Keywords: Online handwriting recognition · Recurrent neural networks · Connectionist temporal classification · Uyghur word transcription

1 Introduction

Handwriting recognition technology based on recorded trajectory with temporal information is called online handwriting recognition, while offline handwriting recognition works on the handwritten shape images which only provide spatial information [1]. Achievements on both online and offline handwriting recognition has been witnessed on well-investigated script kinds [2, 3]. Several competitions were held to improve the handwriting recognition technology on the popular scripts [4, 5]. General pattern recognition systems including recurrent neural networks with connectionist temporal classification-CTC are proving themselves robust for the variety of the script kinds, especially for alphabetic scripts, both in isolated and cursive writing styles [11].

Uyghur is an alphabetic script which is one of the important languages in northwest China and Central Asia. Previous studies on Uyghur handwriting recognition mainly uses classic pattern recognition framework which requires tremendous human observation and expert design to extract features for later classification [6, 7]. A first successful end-to-end unconstrained handwriting recognition system by Li et al. [8]

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 518–527, 2019.
https://doi.org/10.1007/978-3-030-32216-8_50

achieved good results on printed text images. It is fact that recognition of handwritten shapes is more difficult than printed ones.

According to the written characteristic of Uyghur, a word has two kinds of Unicode based representations that either based on character types or specific character forms. In order to compare the effect of the two word transcription methods, this paper conducts comparative handwritten word recognition experiments using recurrent neural networks with connectionist temporal classification-CTC. The experiments are designed in unconstrained recognition manner that the applied model can map handwritten trajectory into sequence of characters directly without prior segmentation and lexicon help.

Research on the application of intelligent systems has been gaining more and more attention in recent years [13, 14]. The handwriting word recognition experiments in this paper will be a reference for later study and development of intelligent systems. The remaining content is arranged in several sections where Sect. 2 introduces Uyghur alphabet and word transcription methods; Sect. 3 details the implemented model structure; Experiment design and results on the collected dataset are described in Sect. 4. At last, Sect. 5 draws a brief conclusion.

2 Alphabet and Word Transcription

Uyghur is one of the typical alphabetic scripts. Like other alphabetic scripts, a word is composed of several characters/letters arranged by language rules. There is an interesting word formation characteristic in Uyghur that a word can be transcribed in two different ways. As given in Table 1, an Uyghur word can be split to two kinds of character sequences, which are respectively based on character forms and character types.

Table 1. Different Unicode representations of a word

Unicodes	Character in the word	Word
by character forms	ق + ـه + ـلا + ـيـ + ـر + ا + ـچـ =	چرايلىق
by character types	ق + ى + ل + ي + ر + ا + چ + ى =	چرايلىق

There are 32 basic Uyghur characters that each of them has several different character shapes according to position within a word. In addition, there are one special component character (char-33) and a compound character (34). The component character is very commonly used in typewriting and the compound character always occurs in handwriting for its ligature shape. According to the alphabet in Table 2, there are 32 +2 basic character types and total 128 character forms.

An Uyghur word can be recorded and represented by two Unicode strings either by using unicodes of specific 128 character forms or by 32+2 character representative

forms. The perfect morphological rules made it possible to arrange corresponding character forms according to the ordered character types of the word. This word coding property is similar to other Arabic based scripts. Although not all character forms are frequently used, this paper takes all 128 character forms and 34 character types into consideration, for the character labels are suppressed with low confidence if they are not present in the word transcription.

Table 2. Uyghur alphabet

End	Mid	Begin	Single	Rep	No.	End	Mid	Begin	Single	Rep	No.
ﺋﺎ	ﺋﺎ	ﺋﺎ	ﺋﺎ	ﺋﺎ	20	ﺎ			ﺍ	ﺍ	1
ﻞ	ﻠ	ﻟ	ﻝ	ﻝ	21	ﺎ			ﺍ		
ﻢ	ﻤ	ﻣ	ﻡ	ﻡ	22	ﻪ			ﻩ	ﻩ	2
ﻦ	ﻨ	ﻧ	ﻥ	ﻥ	23	ﻪ			ﻩ		
ﻪ	ﻪ	ﻫ	ﻫ	ﻫ	24	ﺐ	ﺒ	ﺑ	ﺏ	ﺏ	3
ﺆ			ﺅ	ﻭ	25	ﭗ	ﭙ	ﭘ	ﭖ	ﭖ	4
ﻮ			ﻭ			ﺖ	ﺘ	ﺗ	ﺕ	ﺕ	5
ﯘ			ﯘ	ﯘ	26	ﺞ	ﺠ	ﺟ	ﺝ	ﺝ	6
ﯗ			ﯗ			ﭺ	ﭽ	ﭼ	ﭺ	ﭺ	7
ﯙ			ﯙ	ﯙ	27	ﺦ	ﺨ	ﺧ	ﺥ	ﺥ	8
ﯚ			ﯚ			ﺪ			ﺩ	ﺩ	9
ﯛ			ﯛ	ﯛ	28	ﺮ			ﺭ	ﺭ	10
ﯜ			ﯜ			ﺰ			ﺯ	ﺯ	11
ﯞ			ﯞ	ﯞ	29	ﮋ			ﮊ	ﮊ	12
ﯟ	ﯟ	ﯟ	ﯟ	ﭒ	30	ﺲ	ﺴ	ﺳ	ﺱ	ﺱ	13
ﻲ	ﻴ	ﻳ	ﻱ			ﺶ	ﺸ	ﺷ	ﺵ	ﺵ	14
ﻰ	ﻨ	ﻧ	ﻯ	ﻯ	31	ﻎ	ﻐ	ﻏ	ﻍ	ﻍ	15
ﻯ	ﻤ	ﻣ	ﻯ			ﻒ	ﻔ	ﻓ	ﻑ	ﻑ	16
ﻲ	ﻴ	ﻳ	ﻲ	ﻲ	32	ﻖ	ﻘ	ﻗ	ﻕ	ﻕ	17
	ﯩ	ﯨ		ﯨ	33	ﻚ	ﻜ	ﻛ	ﻙ	ﻙ	18
ﻼ			ﻻ	ﻻ	34	ﮓ	ﮕ	ﮔ	ﮒ	ﮒ	19

Rep, Single, Begin, Mid and End means the representative form, isolated form, beginning form, intermediate form and ending form of a character respectively.

3 End-to-End Handwriting Recognition System

3.1 Input

Raw handwritten trajectory is processed to make short and informative trajectory. The implemented preprocessing techniques include duplication removing and critic point selection. In order to enrich the informative content of the raw input, two dimensional direction vector (Δx, Δy) and another two dimensional pen-state vector are added to the point coordinates of each point [9]. Thus, each point in input sequence is in shape of [x, y, Δx, Δy, PS[0], PS[1]] where PS is for pen-state. Pen-state is confirmed conveniently

by order of neighbor strokes as in that [0, 1] means pen-up state while [1, 0] is for pen-down state. The temporal direction factor is simply calculated using Eq. (1).

$$\Delta x = x_i - x_{i-1} \tag{1}$$

$$\Delta y = y_i - y_{i-1} \tag{2}$$

3.2 Model Architecture

A deep neural network including two bidirectional recurrent layers and two fully connected layers are applied to build online handwriting word recognition system as shown in Fig. 1(a).

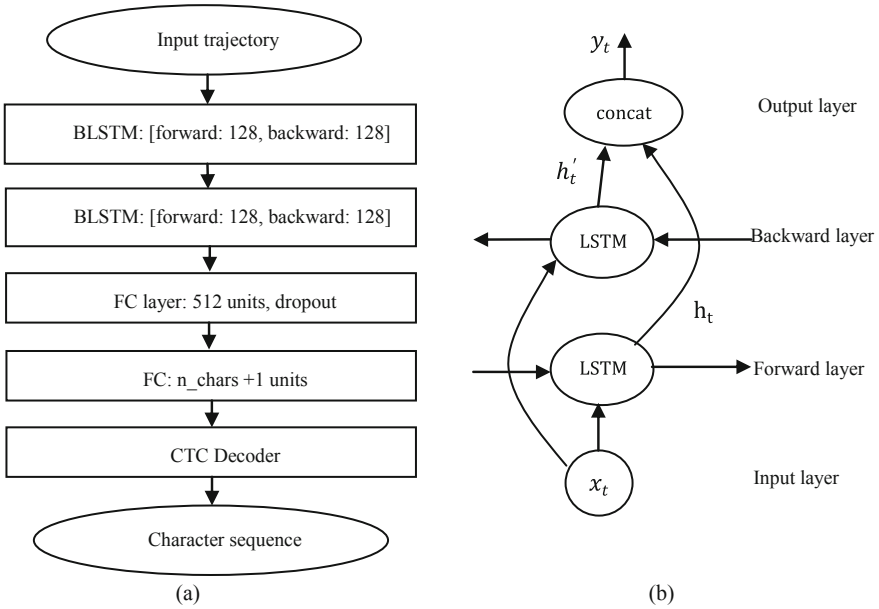


Fig. 1. End-to-End handwritten word recognitions system (a) Model architecture (b) Bidirectional recurrent layer

Considering the subtlety to gradient vanishing of recurrent networks, Long Short Term Memory-LSTM is applied for each cell or unit of the recurrent layers in this paper. The output of the first recurrent layers is directly sent to the second recurrent layer to obtain more generalized sequential feature. The fully connected layers are assumed to further generalize the learned features from the recurrent layers. One of the most effective regularization methods for neural networks, Dropout, is applied on the fully connected layers to avoid overfitting, because dense connectivity in fully connection layers makes large number of variables to the network. The number of neurons

in the last fully connected layer is set by the number of characters to make word transcription and a Blank label used for CTC decoding. CTC decoder provides the last output string (character sequence) by calculating the most possible character sequence.

3.3 Bidirectional LSTM Layer

Handwriting is usually written in either from right to left or from left to right direction. However, many disorders happen in actual handwriting even in small handwriting case such as words. Observing handwritten trajectory from both right-left and left-right directions is more helpful and fit for the nature of online handwritten trajectory [10]. A bidirectional recurrent layer consists two sub recurrent layers. Input sequence is fed to one recurrent layer in original order while another one receives the input sequence in reverse order. Each LSTM cell in a recurrent layer controls input, output and state values to the next state with gate mechanism, as given in Eqs. (2)–(6).

$$i_t = sigm(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$

$$f_t = sigm\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{3}$$

$$o_t = sigm(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + tanh(W_o x_t + U_o h_{t-1} + b_o) \tag{5}$$

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

Where W_i, W_f, W_o are the input-hidden weight matrix, U_i, U_f, U_o are the state-state weight matrix and b_i, b_f, b_o are bias vectors, respectively. I_t, f_t, o_t are the activation values at the input, forget and output gates, while c_t and h_t are the state and output values of the cell.

The output of the two sub-recurrent layers are concatenated into longer sequence, see Fig. 1(b) and Eqs. (7)–(9).

$$Y_{forward} = h_t = [y_{r1}, y_{r2}, \ldots . y_{rN}] \tag{7}$$

$$Y_{backward} = h'_t = [y_{r1}, y_{r2}, \ldots . y_{rN}] \tag{8}$$

$$Y = concat\left(Y_{forward}, Y_{backward}\right) \tag{9}$$

Where y_{rN} and y_{lN} represents the output of the N^{th} node of in right-left and left-right sub-recurrent layers. $Y_{forward}$ and $Y_{backward}$ are the outputs of the two inverse sub-layers and Y is last output of a bi-directional recurrent layer.

3.4 Output

The output is the sequence of alphabet characters that are assumed to be in the handwritten trajectory input [11]. The number of nodes in the last fully connection layer which its output is decoded into character string is set by the number of the

alphabetic characters that the word transcription based on, either by 32+2 overall character types or by 128 specific character shapes, with adding the Blank label especially designed for CTC decoding. Therefore, this paper proposes two systems based on 32+2 basic character types or 128 specific character forms to compare their performances. In this way, an input trajectory is transcribed into two different sequences of Unicode characters by the two systems.

4 Experiments

4.1 Dataset

A dataset has been established by collecting online handwritten word samples from 26 different writers. The dataset contains 900 word classes and each word is recorded in two different character unicode strings, respectively using character type unicodes and character form unicodes. Each writer is asked to write all word classes continuously. The recorded handwritten word trajectories of each writer are saved in separate binary files, with POTEX extension. Each handwritten word sample contains sequentially recorded pen-tip (x, y) coordinates. A stroke is separated from its neighbor by a special stroke-end mark and complete word trajectory is ended by another word-end mark. A handwritten word sample in binary files is put together with its two word transcriptions mentioned above and overall trajectory information including trajectory length, number of strokes etc. The collected 23400 handwritten word samples are divided into training and test sets with respect to the writers to conduct writer independent word recognition experiments. 19800 samples from 22 writers are put in training set while the remained 3600 samples from other 4 writer are used as test set. Statistics on the collected datasets found that words which have 4–10 characters are the most common ones. The longest word is recorded to have 22 characters in the dataset used in this paper. The calculated average numbers of characters is 7.8. The longest and average handwritten word trajectory lengths are found to hold 1023 and 221 points, respectively.

4.2 Design and Configuration

In preprocessing, a point is removed if its distance to previous neighbor is less than half of the average neighbor distance in the stroke. For critical point selection, threshold of $\Pi/6$ is found appropriate in our case. By preprocessing, the average trajectory length is shorted to 67.

According to unicode representations of a word, either by character types or character forms, two unconstrained handwriting word recognition systems are compared in this paper. The two systems are differed only in the width of the last fully connection (FC)-output layer. The system which transcribes input trajectory to a sequence of basic character types is set with 32+2+1 units in the last FC-output layer and noted char34 system in this paper. The another system uses 128+1 units at the last FC layer to generate output sequence of specific character shapes and named char128

system in this context. The transcribed model output is used as word recognition result directly without help of any lexicon search and external language models.

The model performance is evaluated using character error rate-CER and character accurate rate- CAR metric [12] and calculated using Eqs. (10) and (11).

$$CER = \frac{De + Se + Ie}{Nt} \tag{10}$$

$$CAR = 1 - \frac{De + Se + Ie}{Nt} \tag{11}$$

where (Nt) is the total characters in the reference text. (Se), (De), (Ie) denote substitution errors, deletion errors and insertion errors, respectively. Sum of these three errors are just the minimum edit distance to align the output sequence to ground truth and calculated by dynamic programming.

The experiments are conducted using one GTCx980 GPU with 4G RAM for acceleration of training. One of most favored self adaptive optimizers-Adam is implemented in all experiments. Samples from one writer in training set are temporarily used for performance validation during training and remained samples from 21 writers are used to update network parameters. Train samples are rearranged randomly in each epoch and put 64 samples in a minibatch. Global learning rate is lowered by decreasing factor of 0.5 when no improvement seen in successive 3 epochs on validation set.

Training is performed for two sessions in succession. In the first session, initial learning rate and drop-rate is set as 0.001 and 0.5, while the values are set as 0.00001 and 0.75 in the second session of training. Both training sessions use the same early stopping mechanism that training is stopped when 10 successive epochs cannot see any progress on validation set. The generalization ability of the trained model is evaluated on the test set which contains 3600 samples from new 4 writers.

4.3 Results and Discussion

To compare the performances of the two systems, the training procedure is recorded using evaluation results on 10 batches of train and validation set against per epoch of training, as in Fig. 2. Thanks to the short and rich informative input representation obtained by preprocessing, the applied model has got very fast error decline both on train and validation sets. Word transcription using 34 character types has shown better performance than using 128 character forms. Using character type based transcription had steadier decline in training error than using character forms based transcription method.

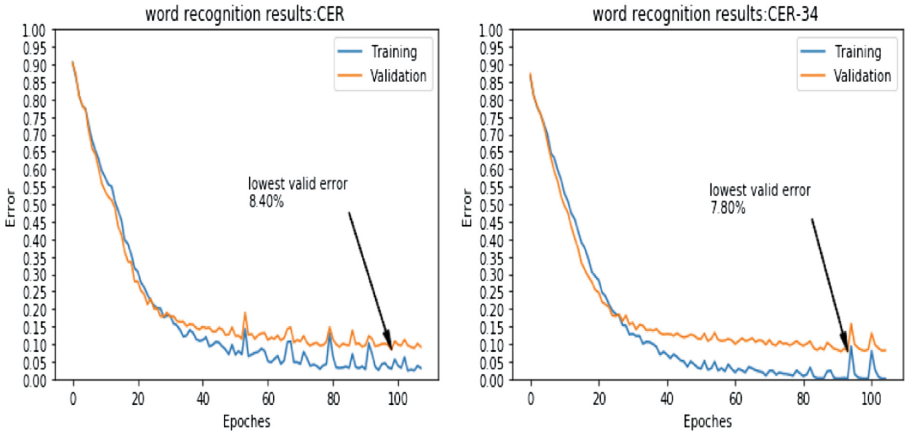
Table 3 gives word recognition results and some other details from the experiments using 34 character type based and 128 character forms based word transcription methods. Since both models are same or very similar in architecture, they are observed to have similar number of variables that each one has almost 1.9M variables and comparable model sizes, see Table 3. Comparing with char34 system, char128 system takes little bit longer time to complete an epoch of training. The char34 system completes an epoch of training for about 4.2 min, while the char128 system uses an

average of 5.7 min. In order to save training time, only 10 batches train and validation subsets are used to navigate the model performance during training. It is also found that char34 system is faster than char128 system on recognition performance, too. The average recognition time per sample for char34 is 0.019 s while char128 system takes about two times longer time to recognize a sample, 0.039 s.

Table 3. Comparison of char34 and char128 systems

Model	No. vars	Model size	No. ep	T/ep	Av-recT	Tr_CER	Te_CER	Te_CAR
Char128	1994117	7.79M	112	~5.7 min	0.039 s	1.78%	14.73%	85.27%
Char34	1849451	7.22M	105	~4.2 min	0.019 s	0.93%	**13.96%**	**86.04%**

Tr_CER and Te_CER: CER on train and test sets, No. ep number of epochs the training stopped, T/ep: average training time per epoch, Av-recT average recognition time per sample, Te_CAR: average character accurate rate on test set.



(a) Training process of char128 system (b) Training process of char34 system

Fig. 2. Training process of char34 and char128 system (results are based on 10 epochs)

Both char34 and char128 systems reached substantial low CER on train set, which are 0.93% and 1.78%, respectively. Also, it can be seen that char34 got better training than char128 system. Evaluation on test set which contains 3600 samples for 900 word classes also showed encouraging results for both systems. 14.73% and 13.96% CER, or 85.27% and 86.04% CAR, results are given for char128 and char34 systems, respectively. The recognition results indicate the superiority of char34 system than char128 system.

According to the training procedure and word recognition results, the experiments in this paper provided good results both systems and showed that char34 system had better performance than char128 system in almost all criteria listed n Table 3. This can be analyzed that char128 system wants to find each specific character form in

handwritten trajectory. However, a handwritten word, especially in cursive natured scripts, always misses some character forms because joining with neighbor characters or casual continues handwriting. The handwritten word sample in Fig. 3(a) has missed some character shapes, and Fig. 3(b) shows a handwritten word trajectory with false written character forms.

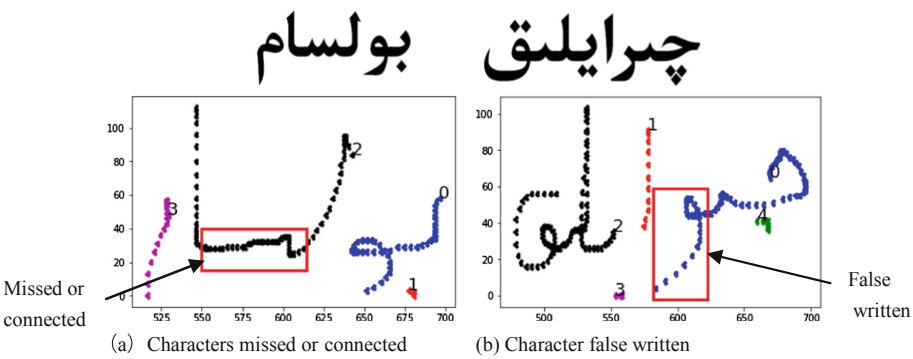


Fig. 3. Some handwritten word samples with printed shapes

Nevertheless, both samples in Fig. 3 are readable and the character forms can be identified within the word context. RNN's capability of using long context information make the labeling by 34 character types more applicable to the casual nature of handwritten word samples. By comparison, labels by 34 character types are more general to detect characters from handwritten word trajectory. Perhaps, using 128 char labels are more suitable for recognizing printed texts instead of handwritten ones.

5 Conclusion

This paper conducts unconstrained online handwriting word recognition experiments using recurrent neural networks on online Uyghur handwritten words. The connectionist temporal classification maps the input handwritten trajectory to a sequence of characters directly and without any lexicon help. According to the writing characteristics of Uyghur, two word transcription methods based on 34 character types and 128 character forms are used as ground-truth labels respectively. Experiment results demonstrate that both word transcription methods are applicable and effective. In experiments, char34-character type based system has better performance in training and evaluation process than char128-character form based system. Char34 and char128 systems obtained 13.96% and 14.73% character error rates on the test set respectively. Different model architectures are to be investigated to further improve the recognition results in later study.

Acknowledgment. This work is supported by National Science Foundation of China (NSFC) under grant number 61462081 and 61263038. The first author is grateful to the National

Laboratory of Pattern Recognition of CASIA (Institute of Automation, Chinese Academy of Sciences) for providing excellent study and experiment environment.

References

1. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Online and offline handwritten Chinese character recognition: benchmarking on new databases. Pattern Recogn. **46**(1), 155–162 (2013)
2. Graves, A., Liwicki, M., Bunke, H., Schmidhuber, J., Fernández, S.: Unconstrained on-line handwriting recognition with recurrent neural networks. In: Conference on Neural Information Processing Systems, pp. 458–464. DBLP, Vancouver (2007)
3. Wu, Y.C., Yin, F., Liu, C.L.: Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recogn. **65**(C), 251–264 (2016)
4. Yin, F., Wang, Q.-F., Zhang, X.-Y., Liu, C.-L.: ICDAR 2013 Chinese handwriting recognition competition. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1464–1470. IEEE, Washington, DC (2013)
5. El Abed, H., Märgner, V.: ICDAR 2009-Arabic handwriting recognition competition. Int. J. Doc. Anal. Recogn. (IJDAR) **14**(1), 3–13 (2011)
6. Ibrahim, M.: Key technologies for recognition of online handwritten Uyghur characters and word. Ph.D. thesis, Wuhan University, China (2013)
7. Simayi, W., Ibrayim, M., Tursun, D., Hamdulla, A.: Survey on the features for recognition of on-line handwritten Uyghur characters. Int. J. Sig. Process. Image Process. Pattern Recogn. **9**(3), 45–58 (2015)
8. Li, P., Zhu, J., Peng, L., Guo, Y.: RNN based Uyghur text line recognition and its training strategy. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS), pp. 19–24. IEEE, Santorini (2016)
9. Zhang, X.Y., Yin, F., Zhang, Y.M., Liu, C.L., Bengio, Y.: Drawing and recognizing Chinese characters with recurrent neural network. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 849–862 (2018)
10. Liwicki, M., Graves, A., Fernàndez, S., Bunke, H., Schmidhuber, J.: A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In: Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR (2007)
11. Graves, A.: Connectionist temporal classification. In: Graves, A. (ed.) Supervised Sequence Labelling with Recurrent Neural Networks. SCI, vol. 385, pp. 61–93. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24797-2_7
12. Su, T.H., Zhang, T.W., Guan, D.J., Huang, H.J.: Off-line recognition of realistic Chinese handwriting using segmentation-free strategy. Pattern Recogn. **42**(1), 167–182 (2009)
13. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS One **13**(5), 1–23 (2018)
14. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)