International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 10, No. 6 (2017), pp. 87-96
http//dx.doi.org/10.14257/ijsip.2017.10.6.08

Character Recognition at Low Resolution Based on Hopfield Network

Liu Yang[1*], Shi Yong-kui[1,2], Xu Yi-ni[1] and Zhang Liang-liang[1]

[1]Department of Industrial Engineering, College of Mining and Safety Engineering, Shandong University of Science and Technology, TsingTao, 266590, China;
[2]State Key Laboratory Breeding Base for Mine Disaster Prevention and Control in Shandong Province, Shandong University of Science and Technology, TsingTao, 266590, China
**Corresponding author E-mail:fanxipingly@163.com(SHI Yong-kui)*

Abstract

In view of the current problems of inefficient character recognition at low resolution, this paper designed one convenient model for character recognition based on discrete hopfield neural network. A critical character matrix was generated in the different noise intensities by random noise. Background subtraction was used to collect the steady states of the characters. Then the recognition model can recognize the character matrix in a given noise intensity. The results showed that when the noise intensity was less than 0.3, the established model had a good recognition performance for the characters at low resolution. With the increase of the noise intensity, the resolution of characters got low, but the accuracy of character recognition model declined slightly. It still maintained a high recognition efficiency. But these characters, including 3, 6, 8 and 9, were not easily identified gradually. When the noise intensity was more than 0.3, the performance of the recognition model started to decrease gradually and lacked stability. The recognition model was getting harder to distinguish the characters as thirty percent of the positional values of the character matrix changed. Take example for 6, it can be recognized in the noise intensity r=0.05, 0.1and 0.2 while it may not be recognized in the noise intensity r=0.15 and 0.25. The recognition result of a given character in the different noise intensities was various. On the whole, the recognition model achieved excellent performance in terms of recognition error rates and recognition efficiency.

***Keywords**: Character Recognition; Low Resolution; noise intensity; recognition performance; Hopfield Network*

1. Introduction

In daily life, we often encounter these problems of inefficient character recognition at low resolution, especially in black and white figures. With the continuous development of new technology, scholars at home and abroad came up with a number of ways and techniques to distinguish the characters, including template matching and optical character recognition. Given this, we researched some relevant literatures and analyzed their advantages and disadvantages.

Tong Jianjun and Zou Mingfu[1] put forward a method of discerning precisely in plate number character recognition which was called as " sub-regional right value template". From the data of test, the accuracy for a given character was effective in plate license recognition, but it was still lower than the accuracy for all characters. CHI Xiaojun[2] presented a simple method for license plate recognition based on SVM. The data of images were translated into text data to avoid the excessive reliance on the extracted image features. However, the different parameters in the kernel function and penalty factors (c and g) had a

ISSN: 2005-4254 IJSIP

considerable influence on the recognition efficiency. PING Yuan and LI Huina[3] optimized the skeleton of the standard plate license character and improved the variable length coding of the skeleton and presented a new method for plate license character recognition. The proposed method was applied easily, but the recognition accuracy of Chinese characters was lower than that of the letters and numbers. FU Desheng and ZHANG Xueyong[4] used the improved hopfield neural network to identify numbers. The method improved the memory capability and the correct-identification rate of traditional network, but the recognition performance needed to be improved because of existing pseudo steady states. ZHANG Kunyan[5] et al designed a new identification system based on the BP network. The system had good performance even when the images had low quality and the license plates were located in a complicated natural scene. However, the training process of the model required manual intervention more or less. ZHAO Zhihong, YANG Shaopu and MA Zengqiang[6] proposed the application of convolutional neural network LeNet-5 in license plate character recognition. The accuracy and the speed of convolutional neural network were superior to the three-layer BP neural network, but it may not exceed the multi-layer BP neural network. WANG Wei, MA Yongqian and PENG Qiang[7] studied an algorithm of character recognition based on SVM binary classification. The algorithm had good feasibility and effectiveness, but the training speed was slow and different classifications still may overlap. Jin Can[8] designed a discrete hopfield neural network(DHNN) model with the associative memory function using MATLAB. It can correctly recognize the numerical dot matrixes with the noises. Because there were a lot of pseudo stable points in the DHNN, the model was hard to get the real steady states. Zhu Haodong and Li Hongchan[9] put forward a character recognition algorithm based on the feature weighting and fuzzy template matching. The recognition rate and robustness were effectively improved, but it can't separate completely the character stroke and character tailing. FANG Chengzhi, ZHOU Pin and FU Shiqing[10] combined the features of the characters with character recognition and proposed a method(LS-SVM) to analyze the feasibility of this wavele kernel function. This way improved the recognition rate while the recognition time was reduced, but the best kernel function may not be restricted to wavelet kernel function. LI Shanshan, LI Yimin and GUO Zhenzhen[11] proposed a phased license plate recognition algorithm based on BP neural network and convolution neural network. The accuracy of recognition algorithm was improved and the time of recognition algorithm was shortened, but the generalization ability of the algorithm reached a lower level. Rong Cheng and Yanping Bai[12] presented a character segmentation algorithm based on the shortest distance classification, which took advantage of exactly seven points gained from SOM as class centers. The proposed method can be implemented efficiently, but the recognition accuracy of Chinese characters was lower than the accuracy of the letters and numbers.

Overseas scholars made a lot of researches on their native characters, such as Tamil, Hindi, Persian, Devanagari, ect. Devi, S. Sangeetha and Amitha, T[13] used a hidden Markov model (HMM) classifier for character recognition of offline Tamil handwritten. Wahi, Amitabh, Sundaramurthy, S and Poovizhi, P[14] developed an OCR system for the recognition of basic characters in handwritten Tamil language. Singh, Pratibha, Verma, Ajay and Chaudhari, Narendras[15] presented a method for the recognition of handwritten Hindi numerals. Sajedi, Hedieh[16] presented an important step towards the standardization of research works on optical character recognition in Persian language. Pandya, Mrudang D and Patel Jay, R[17] presented the survey of how efficient an artificial neural network can be utilized for character recognition of Devanagari handwritten.

Despite intense researches, however, these problems of inefficient character recognition especially at low resolution remain, to a wide extent, unresolved. How to extract the complete information from incomplete characters is the key in the character recognition. It is necessary to present a convenient model or method for character recognition in terms of recognition error rates and recognition efficiency.

2. Background Subtraction and Discrete Hopfield Network

2.1. Background Subtraction

For convenience sake, background subtraction was applied to obtain these character features in the steady states. The basic principle of background subtraction was as shown in figure 1.

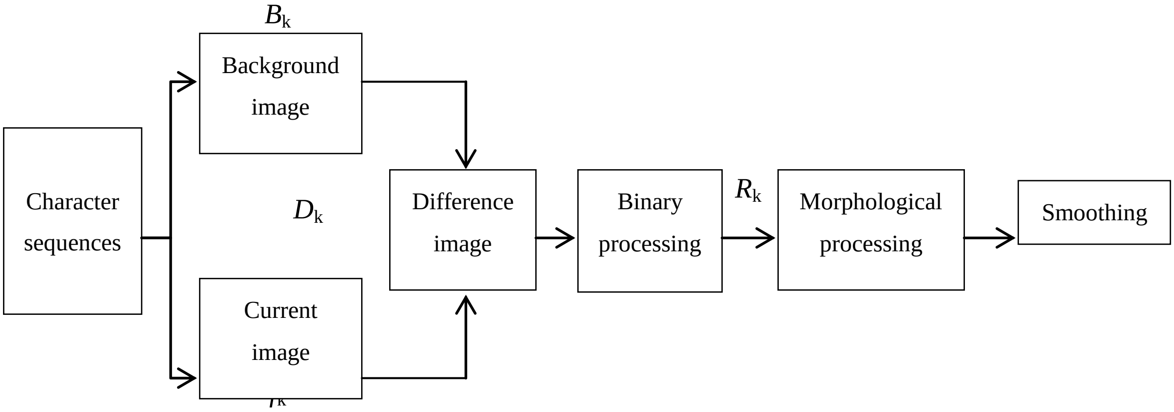

Figure 1. The Principle Diagram of Background Subtraction

As shown in figure 1, we set the current frame image as $f_k(x, y)$ and the background frame image as $B_k(x, y)$. Then the difference image was shown in the formula 1.

$$D_k(x,y) = |f_k(x,y) - B_k(x,y)| \qquad \text{Formula 1}$$

We make the difference image binarization processing, as shown in the formula 2.

$$R_k(x,y) = \begin{cases} 1, & D_k(x,y) \geq T \\ 0, & D_k(x,y) < T \end{cases} \qquad \text{Formula 2}$$

If a pixel value in the difference image is greater than or equal to a given threshold value, the pixel is a foreground valve. If not, it is considered the pixel is a background valve. We transform the difference image R_k to binaryzation and obtain D_k, then make morphological processing and connectivity analysis.

2.2. Hopfield Neural Network

Discrete hopfield neural network (DHNN) is a single-layer network with taking “1” and “-1” as the input. The structure of discrete hopfield neural network including *n* neurons was shown in figure 2.

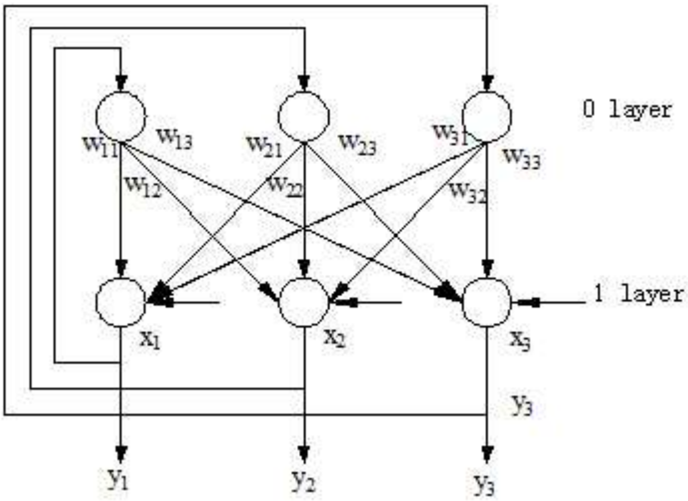

Figure 2. The Structure of Discrete Hopfield Neural Network

Neurons in the first layer sum up the information of the input and the weight coefficient. The information of the output is generated by the nonlinear threshold function *f*. For an output layer network with *n* neurons, its state at time (*t*) is one n-dimensional vector, as shown in the formula 3.

$$Y(t) = [y_1(t), y_2(t), ..., y_n(t)]^T \quad \text{Formula 3}$$

There are 2^n network states because $y_i(t)(i=1,2,\ldots,n)$ can be valued to "1" or "-1". As the state of node *j* of DHNN at time (*t*) changed, the node at time (*t*+1) can be obtained as shown in the formula 4 and formula 5.

$$y_i(t+1) = f[u_j(t)] = \begin{cases} 1, u_j(t) \geq 0 \\ -1, u_j(t) < 0 \end{cases} \quad \text{Formula 4}$$

$$u_j(t) = \sum_{i=1}^{n} \omega_{ij} y_i(t) + x_j - \theta_j \quad \text{Formula 5}$$

DHNN adopts the orthogonal method to adjust and modify its weights and the overall steps are as follows:

Step 1: To input *N* modes $t = \{t, t^2, ..., t^{N-1}, t^N\}$ and parameters τ , *h*;

Step 2: To calculate $A = \{t - t^N, t^2 - t^N, ..., t^{N-1} - t^N\}$;

Step 3: To check *A* into singular value decomposition $A=USV^T$ and calculate the rank *K*=rank (*A*) of *A*;

Step 4: To calculate $T^p = \sum_{i=1}^{K} u^i (u^i)^T$ and $T^m = \sum_{i=K+1}^{N} u^i (u^i)^T$ from $U^p = \{U^1, U^2, ..., U^k\}$ and $u^m = \{u^{K+1}, u^{K+2}, ..., u^N\}$;

Step 5: To calculate $W^t = T^p - \tau \times T^m$ and $b^t = t^N - W^t \times t^N$;

Step 6: To calculate $W = \exp(h \times W^t)$;

Step 7: To calculate $b = U \times \begin{bmatrix} C_1 \times I(K) & 0(K, N-K) \\ 0(N-K, K) C_2 \times I(N-K) \end{bmatrix} \times U^T \times b^\tau$, among them, C_1=exp(*h*)-1,C_2=-[exp(-τ×*h*)-1]/ τ.

DHNN gains the associative memory function by training and learning to realize the local minimum of energy function until it reaches a steady state.

The dynamic rule of DHNN is that if the network node is in the initial state of *S* (0), after the *t* steps it will achieve the state of *S* (*t*+1) by the following formula 6.

$$S_t(t+1)=\operatorname{sgn}\left[\sum_{j=1}^{n}W_{ij}S_j(t)+I_i\right] \qquad \text{Formula 6}$$

Among them, *sgn* is the sign function. The learning steps are as follows, among them *I*=0.

Step 1: To set the weights according to the hebb rule.

Step 2: To initialize the unknown samples:

$$S_i(0)=x_i,\ i=1,2,...,n \qquad \text{Formula 7}$$

Among them, $S_i\,(t)$ is the output of node *i*; x_i is the *i*th element of the unknown samples.

Step 3: To iterative calculate.

$$S_i(t+1)=\operatorname{sgn}\left[\sum_{j=1}^{n}W_{ij}S_j(t)\right] \qquad \text{Formula 8}$$

The process of iteration will not stop until the states of output of the node don’t change.

Step 4: To go back to step 2 and calculate again.

The state of DHNN tends steady gradually, with the whole neurons converging at a particular point, or in a cyclical iteration, or in a chaotic state, after the above four steps.

3. Recognition Model Establishment

We took advantage of the associative memory function of DHNN and designed a convenient model for character recognition in the light of the dynamics learning rule.

3.1. The Standard Steady State of Simulated Characters

We described each simulated character with a matrix of 10×10 intuitively. The parts of the characters were in terms of “1” and the blank parts were in terms of “-1”. Take example for“0”, the standard matrix can be obtained, like array_zero=[-1 1 1 1 1 1 1 1 1 -1; -1 1 1 1 1 1 1 1 1 -1; -1 1 1 -1 -1 -1 -1 1 1 -1; -1 1 1 -1 -1 -1 -1 1 1 -1; -1 1 1 -1 -1 -1 -1 1 1 -1; -1 1 1 -1 -1 -1 -1 1 1 -1; -1 1 1 -1 -1 -1 -1 1 1 -1; -1 1 1 -1 -1 -1 -1 1 1 -1; -1 1 1 1 1 1 1 1 1 -1; -1 1 1 1 1 1 1 1 1 -1].

3.2. A Model Building

We formed the matrixes of the characters (0~9) to constitute the training samples *T* = [zero, one, two, three, four, five, six, seven, eight, nine]’. Then we applied the function of new hop () to create a model based on DHNN.

3.3. The Identification of the Steady Characters

We applied the method of random noise to simulate the steady states of the characters at low resolution. The matrixes of the characters (0~9) were generated in the different noise intensities. If a pixel value was greater than or equal to the given threshold, it argued that the pixel was a foreground pixel as “1”. Otherwise, it argued that the pixel was a background pixel as “-1”. We aimed to require the model to identify these ten characters in a certain noise.

4. Recognition Model Test

The characters identified by background subtraction in the steady states were input into the model. Meanwhile we adjusted the range of the noise intensity to render the model to recognize the matrixes of each character in a certain noise.

4.1. Recognition Test in the different Noise Intensities

4.1.1. The Noise Intensity r=0.05: Figure 3 was the result of recognition test for all characters in the noise intensity r=0.05.

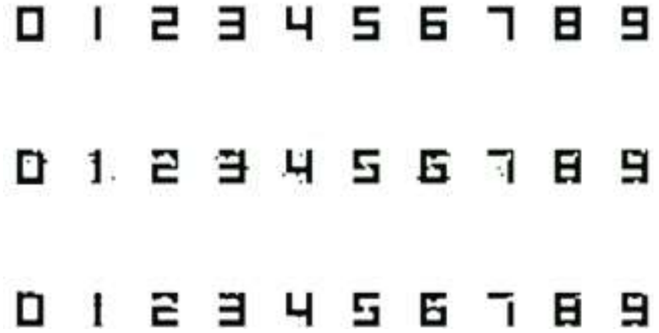

Figure 3. The Result of Recognition Test with r=0.05

Figure 3 showed that the result of recognition test for the characters (0~9) was remarkable. The accuracy of recognition test was 100% and no error.

4.1.2. The Noise Intensity r=0.1: Figure 4 was the result of recognition test for all characters in the noise intensity r=0.1.

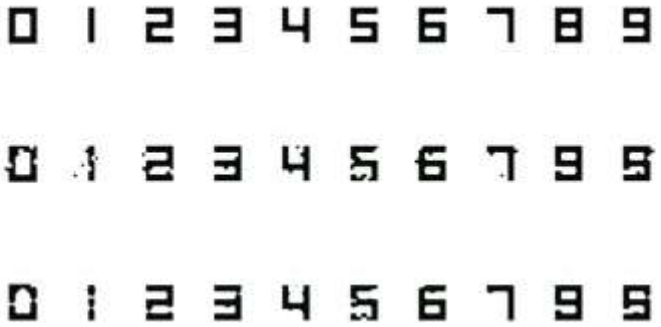

Figure 4. The Result of Recognition Test with r=0.1

Figure 4 showed that the result of recognition test for the characters (0~8) was fine. Only one character (9) was easily identified as 8 and the accuracy of recognition test was 90%.

4.1.3. The Noise Intensity r=0.15: Figure 5 was the result of recognition test for all characters in the noise intensity r=0.15.

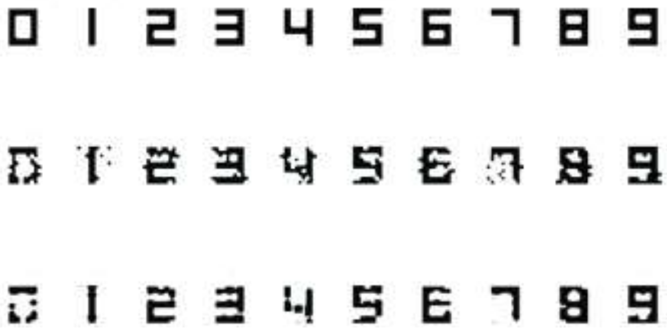

Figure 5. The Result of Recognition Test with r=0.15

Figure 5 showed that the result of recognition test for the characters except 6 was also fine. Only one character (6) was easily identified as 8 and the accuracy of recognition test was 90%.

4.1.4. The Noise Intensity r=0.2: Figure 6 was the result of recognition test for all characters in the noise intensity r=0.2.

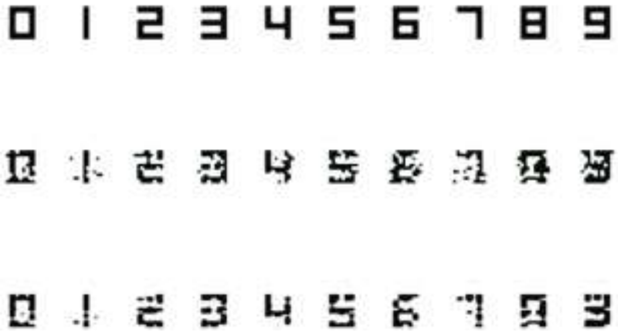

Figure 6. The Result of Recognition Test with r=0.2

Figure 6 showed that the result of recognition test for the characters except 8 and 9 was good. One character (8) was easily identified as 0 and another one character (9) was easily identified as 3. The accuracy of recognition test was 80%.

4.1.5. The Noise Intensity r=0.25: Figure 7 was the result of recognition test for all characters in the noise intensity r=0.25.

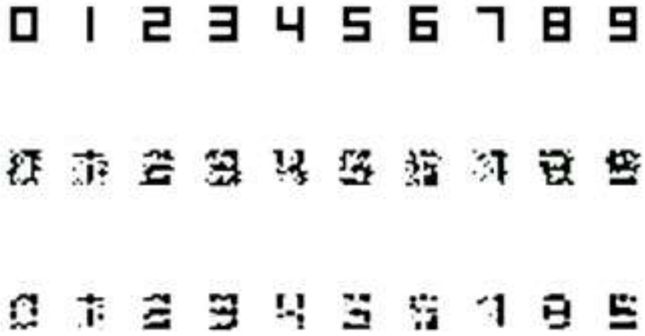

Figure 7. The Result of Recognition Test with r=0.25

Figure 7 showed that the result of recognition test for the characters except 3 and 9 was also good. One character (3) was easily identified as 9 and another one character (6) was easily identified as 5. The accuracy of recognition test was 80%.

4.1.6. The other Noise Intensities: In order to further study on the influence of the bigger noise intensities on the recognition performance for the characters, we tested the characters in the noise intensities r=0.3, 0.35, 0.4 and 0.45. Figure 8 was the result of recognition test for all characters in the noise intensity r=0.3, 0.35, 0.4 and 0.45.

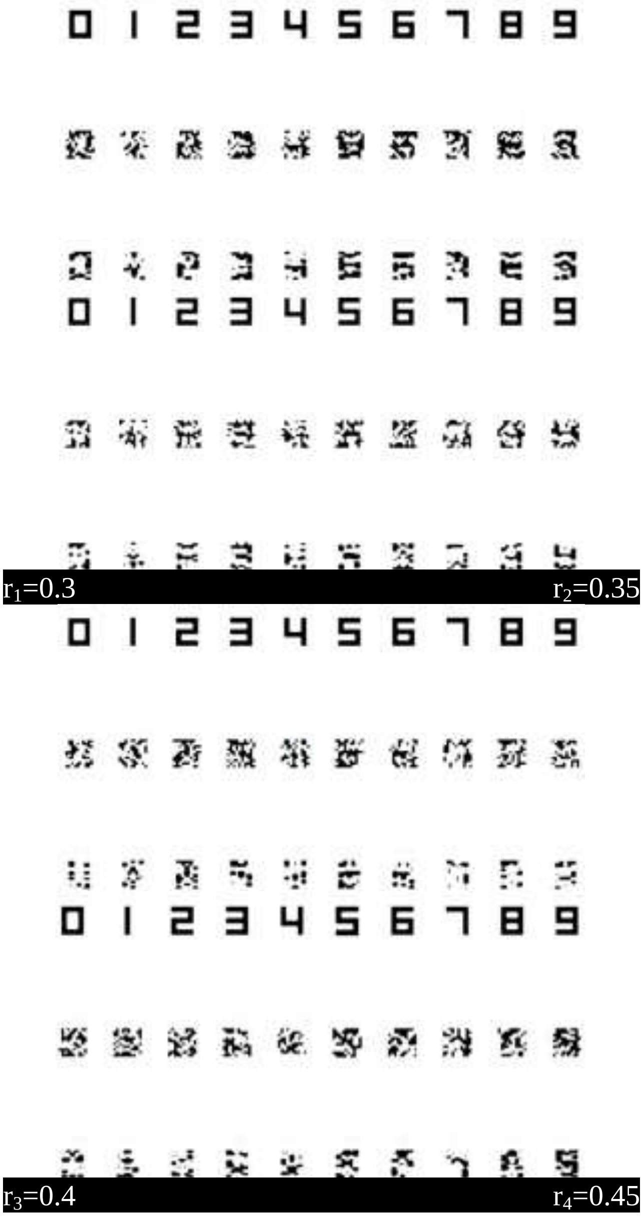

Figure 8. The Result of Recognition Test with r=0.25, 0.35, 0.4 and 0.45

4.2. Result Analysis

4.2.1. Recognition Accuracy in the Certain Noise Intensity: When the noise intensity was less than 0.3, the established model had a good recognition performance at low resolution, as shown in table 1.

Table 1. The Results of Recognition Test with the Noise Intensity r<0.3

Noise intensity	Recognition results										Rrecognition accuracy
	0	1	2	3	4	5	6	7	8	9	
0.05	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	100%
0.1	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*N*	90%
0.15	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*N*	*Y*	*Y*	*Y*	90%
0.2	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*N*	*N*	80%

0.25	*Y*	*Y*	*Y*	*N*	*Y*	*Y*	*N*	*Y*	*Y*	*Y*	80%

The characters including 3, 6, 8 and 9 were not easily identified because the parts of the characters were more than the blank parts. With the increase of the noise intensity, the resolution of characters got low, but the accuracy of character recognition declined slightly, as shown in the figure 8. It still maintained a high recognition efficiency.

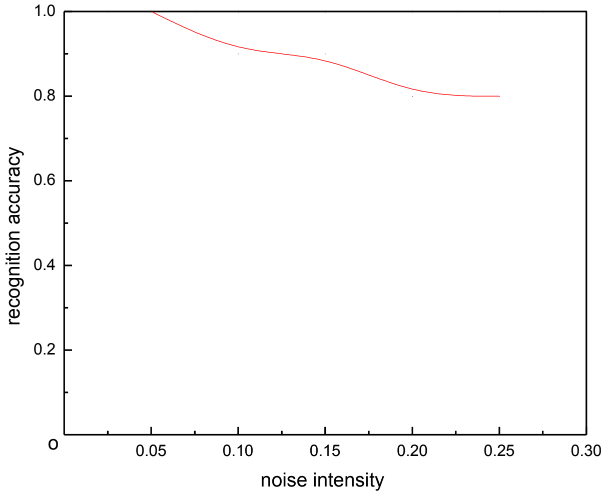

Figure 8. The Accuracy of Recognition Test with r=0.05, 0.1, 0.15, 0.2 and 0.25

When the noise intensity was more than 0.3, the performance of the recognition model started to decrease gradually and lacked stability. The recognition model was getting harder to distinguish as thirty percent of the positional values of the character matrix changed.

4.2.2. Recognition Accuracy in the Different Noise Intensities: Table 1 showed that a certain character can be identified in the different noise intensities. Take example for 6, it can be recognized in the noise intensity r=0.05, 0.1and 0.2 while it may not be recognized in the noise intensity r=0.15 and 0.25. The recognition result of a given character in the different noise intensities was various. This was mainly due to the random character matrixes.

5. Summary

We aimed to propose a convenient model for character recognition at low resolution. A critical character matrix was generated in the different noise intensities by random noise. Background subtraction was used to collect the steady states of the characters. The results showed that when the noise intensity was less than 0.3, the established model had a good recognition performance for the characters at low resolution. And these characters, including 3, 6, 8 and 9, were not easily identifiable. When the noise intensity was more than 0.3, the performance of the recognition model started to decrease gradually and lacked stability. On the whole, the recognition model achieved excellent performance in terms of recognition error rates and recognition efficiency. We, however, found that when the noise intensity increased, it was likely to cause an error in the process of recognition. How to further reduce the pseudo steady states existing in the network, how to learn from the network weights in the sample pretreatment are the key to further improve the accuracy of recognition model in the higher noise intensity.

Acknowledgment

This work is partially supported by SHI Yong-kui1and REN DAWEI. The authors also acknowledge gratefully the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] Tong Jianjun and Zou Mingfu, "A Method of Discerning Precisely in Plate License Character Recognition[J]", Computer Engineering and Applications, no.2, **(2005)** , pp.223-226.

[2] CHI Xiao-jun, "A Method for License Plate Recognition Based on Support Vector Machine[J]", Information Technology and Informatization, no.6, **(2007)**, pp. 103-106.

[3] PING Yuan and LI Huina, "Method of discerning precisely in plate license character recognition[J]", Computer Engineering and Design, vol. 29, no. 9, **(2008)**, pp. 2410-2412.

[4] FU De-sheng and ZHANG Xue-yong, "Noise-Figure Identification Based on Hopfield Neural Network[J]", Communications Technology, vol. 43, no. 1, **(2010)**, pp. 52-65.

[5] ZHANG Kunyan, ZHONG Yiya, MIAO Songchi and WANG Guijuan, "A Plate-Character Identification System Based on Global-Valve Binarization and the BP Neural Network[J]", Computer Engineering and Science, vol. 32, no. 2, **(2010)**, pp. 88-90.

[6] ZHAO Zhi-hong, YANG Shao-pu and MA Zeng-qiang, "License Plate Character Recognition Based on Convolutional Neural Network LeNet-5[J]", Journal of System Simulation, vol. 22, no. 3, **(2010)** , pp. 638-641.

[7] WANG Wei, MA Yongqiang and PENG Qiang, "Application of SVM multi-classifier on license plate character recognition[J]", Computer Engineering and Design, vol. 32, no. 9, **(2011)**, pp.3166-3169.

[8] Jin Can, "On numerical recognition using discrete Hopfield neural network[J]", Computer Era, vol.19, no. 3, **(2012)** , pp 1-3.

[9] Zhu Haodong and Li Hongchan, "Character recognition based on feature weight fuzzy template matching[J]", Journal of Lanzhou University of Technology, vol. 39, no.1, **(2013)**, pp. 68-71.

[10] FANG Cheng-zhi, ZHOU Pin and FU Shi-qing, "Research on Plate Character Recognition Based on Wavelet Kernel LS –SVM[J]", Computer Technology and Development, vol.25,no.3, **(2015)**, pp. 86-90.

[11] LI Shanshan, LI Yimin and GUO Zhenzhen, "Research on a phased license plate character recognition algorithm based on neural network[J]", Industrial Instrumentation and Automation, no.2,**(2016)**, pp. 7-10.

[12] Rong Cheng and Yanping Bai, "A Novel Approach for License Plate Slant Correction, Character Segmentation and Chinese Character Recognition[J]", International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 7, no.1,**(2014)**, pp. 353-364.

[13] Devi, S. Sangeetha and Amitha, T, "Offline handwritten writer independent Tamil character recognition[J]", 2014 International Conference on Information Communication and Embedded Systems, **(2014)**, ICICES.

[14] Wahi, Amitabh, Sundaramurthy, S and Poovizhi, P, "Handwritten Tamil character recognition[J]", 5th International Conference on Advanced Computing, ICoAC 2013, **(2014)**, pp. 389-394, October 12.

[15] Singh, Pratibha, Verma, Ajay and Chaudhari, Narendras, "Feature selection based classifier combination approach for handwritten Devanagari numeral recognition[J]", Sadhana - Academy Proceedings in Engineering Sciences, vol. 40, no. 6,**(2015)**, pp. 1701-1714.

[16] Sajedi, Hedieh, "Handwriting recognition of digits, signs, and numerical strings in Persian[J]", Computers and Electrical Engineering, vol. 49, **(2016)**, pp. 52-65.

[17] Pandya, Mrudang D and Patel Jay, R, "A survey: Artificial neural network for character recognition[J]", Advances in Intelligent Systems and Computing, vol. 436,**(2016)**, pp. 403-410.

Authors

Liu Yang, he was born in March, 1991 studies in College of Mining and Safety Engineering, Shandong university of science and technology and his major is industrial engineering; He is mainly engaged in the researches about industrial engineering, human factors engineering, ergonomics, man-machine interface and other aspects in pipeline engineering, especially in coal mines. He has participated in a number of scientific researches with Dr. Ren Dawei and Professor Shi Yongkui in recent years. Shi Yongkui is the Corresponding author for this academic paper.