Hybrid CNN-GRU Model for High Efficient Handwritten Digit Recognition

Vantruong Nguyen
Xidian University
No. 2 South Taibai Road
Xi'an, China
86-17809290857
nguyentruong.xian@gmail.com

Jueping Cai
Xidian University
No. 2 South Taibai Road
Xi'an, China
86-18629392982
jpcai@mail.xidian.edu.cn

Jie Chu
Xidian University
No. 2 South Taibai Road
Xi'an, China
86-18729212698
jiechu@stu.xidian.edu.cn

ABSTRACT
Recognition of handwritten digits is a challenging research topic in Optical Character Recognition (OCR) in recent years. In this paper, a hybrid model combining convolutional neural network (CNN) and gate recurrent units (GRU) is proposed, in which GRU is used to replace the CNN fully connected layer part to achieve high recognition accuracy with lower running time. In this model, the features of the original image are firstly extracted by the CNN, and then they are dynamically classified by the GRU. Experiment performed on MNIST handwritten digit dataset suggests that the recognition accuracy of 99.21% while the training time and testing time is only 57.91s and 3.54s, respectively.

CCS Concepts
• **Computing methodologies→ Neural networks** • **Computing methodologies→ Image processing.**

Keywords
Handwritten digit recognition; convolutional neural network; gate recurrent units; MNIST dataset.

1. INTRODUCTION
Handwritten digit recognition is an important part of optical character recognition, and it has been widely used in many applications such as postal services, bank check and handwritten notes transcription, etc. Many methods with remarkable recognition results have been proposed such as K-Nearest-Neighbors (KNN) [1][2][3], Histogram of Oriented Gradient (HOG) [4][5], Support Vector Machines (SVM) [6][7][8], Artificial Networks [9][10][11], deep Q-learning-based recognition [15]. But it that writing style with the characters of different size, shape and thickness remains an opening and challenging problem. As shown in Figure 1, different handwritten digit example of MNIST dataset shows different style of the writers.

Handwritten digit recognition is generally divided into two main steps: feature extraction and classification. These steps can all be realized by the CNN [11]. However, since the CNN has only an excellent performance in feature extraction step, several hybrid structures such as CNN-SVM [12][13], CNN-HMM (Hidden Markov Model) [14], and CNN-RNN (Recurrent Neural Network) [16] have been proposed to improve the recognition accuracy. In these papers, CNN was only used for the feature extraction task.

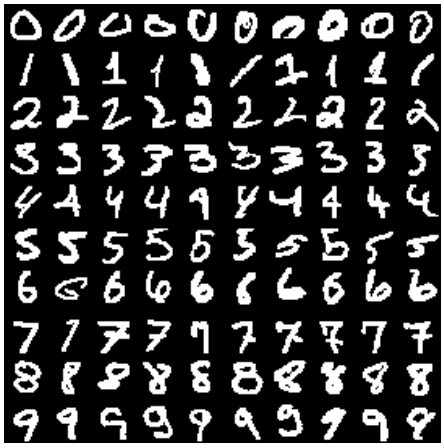


Figure 1. Handwritten digit examples of MNIST dataset.

On the premise of ensuring the recognition accuracy, to reduce the calculation cost is also an important aspect for researchers. Qiao J. et al. [15] combined the Deep Believe Network (DBN) and the Q-learning algorithm to balance the recognition rate and the running time achieving 99.18% recognition accuracy and 21.46s testing time.

In this paper, based on traditional CNN structure, a hybrid structure combining CNN and GRU is proposed to further improve recognition accuracy and reduce running time. GRU is a typical Recurrent Neural Network structure which has been applied widely in sequence problems. The motivation of this paper has two following aspects: Firstly, this structure maintains the CNN superior performance in the feature extraction to extract features from the original image. Secondly, GRU is able to convert handwritten digit recognition into a sequence problem to efficiently perform classification. Experiment result performed on MNIST dataset indicates that the recognition accuracy increases to 99.21%, and the testing time reduces to 3.54s.

The rest of this paper is distributed as follows: Section 2 discusses the structures and principles of CNN and GRU. Section 3 describes the details of the hybrid CNN-GRU model.

AIPR 2019, August 16–18, 2019, Beijing, China

ACM ISBN 978-1-4503-7229-9/19/08…$15.00
DOI: https://doi.org/10.1145/3357254.3357276

Experimental results and the corresponding discussions are presented in Section 4.

2. CNN AND GRU BASIC MODELS

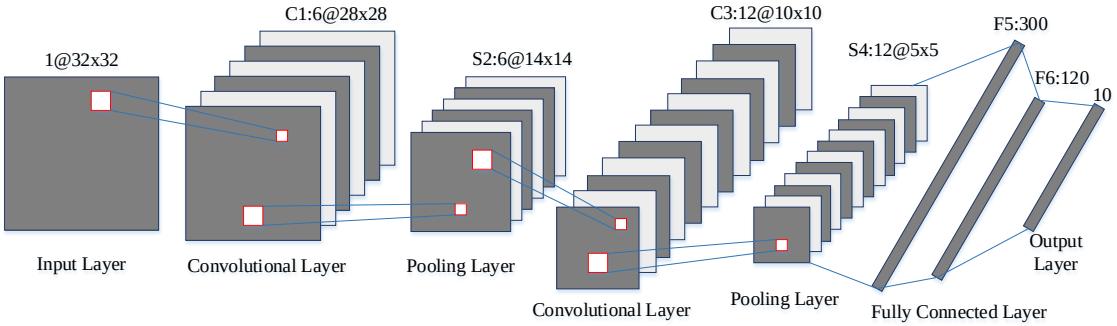
2.1 Convolutional Neural Networks

CNN is a multi-layer neural network structure which is composed of multiple convolutional layers, pooling layers and fully connected layers, in which the convolutional layers and pooling layers are used to extract features of the handwritten digit character. In addition, the fully connected layers are used to classify the handwritten digit from the extracted feature maps. Each layer input is connected to the previous layer output, and its output passes to the next layer. The parameters of the network are shared and trained by the back propagation method.

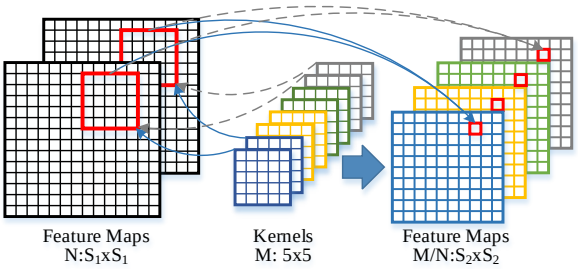
The CNN architecture is illustrated in Figure 2(a) with seven layers including two convolutional layers, two subsampling layers and three fully connected layers. In the convolutional layer (shown in Figure 2(b)), the output feature maps are computed via sliding 5 by 5 kernels [11] on the previous layer feature maps. With N input feature maps with size S_1 by S_1 and the sliding stride of 1, the convolutional layer outputs are M/N feature maps with size S_1-4 by S_1-4, where M is the number of convolutional kernels. In the pooling layer (shown in Figure 2(c)), the number of output maps is unchanged, and each map size reduces from the input feature size S_2 to $S_2/2$ by using a kernel size of 2 by 2 and a polling stride of 2. The number neurons of the first fully connected layer are obtained by converting all features of the last pooling layer into one vector. In the fully connected layer (shown in Figure 2(d)), there is unidirectional full connection from the previous layer to current fully connected layer, and no connection between neurons in the same layer. Each neuron value is computed by Equation 1, where w is the weight of the connection from l-1 layer to l layer, b_t^l is a bias, and f is an activation function.

$$h_t^l = f(\sum w \cdot h_t^{l-1} + b_t^l) \qquad (1)$$

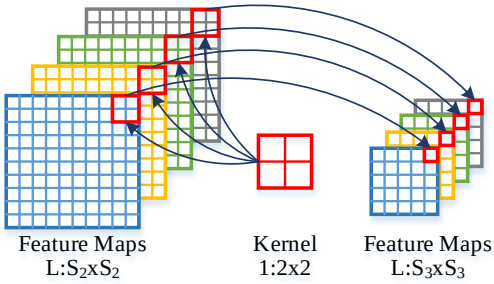
Lenet-5 model proposed by LeCun [11] is the most well-known model to perform handwritten character recognition. In this paper, the CNN model is built based on the architecture of Lenet-5, and the CNN model parameters are chose from the model of Maitra [13]. As shown in Figure 2(a), the proposed CNN architecture as



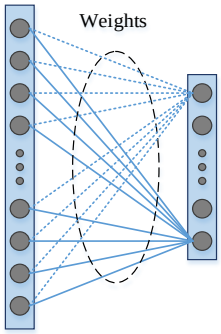
(a) Architecture of convolutional neural network



(b) Operation on convolutional layer



(c) Operation on pooling layer



(d) Operation on fully connected layer

Figure 2. Convolutional Neural Network.

I:1@32x32-C1:6@28x28-S2:6@14x14-C3:12@10x10-S4:12@ 5x5-F5:300N-F6:120NO:10N. That means the input layer has one input image of size 32 by 32, and the parameters of following convolutional layers and pooling layers have the same meanings. The fully connected layers have 300 neurons and 120 neurons, respectively. Finally, output layer has 10 neurons correspond ten digit 0~9 classes.

2.2 Gate Recurrent Units

GRU is a concise version of the long short term memory model (LSTM) [18], which was proposed by Cho K. in 2014 [17]. The LSTM has three gates including input gate, forget gate and update gate. Meanwhile, the GRU is simpler than the LSTM since it has only two gates, which are reset gate and update gate. These gates are used to determine whether the information is useful or not. The useful information is reserved while useless information is forgotten. As shown in Figure 3, in GRU structure, W_z, W_r and W represent the update gate, reset gate and candidate information, respectively.

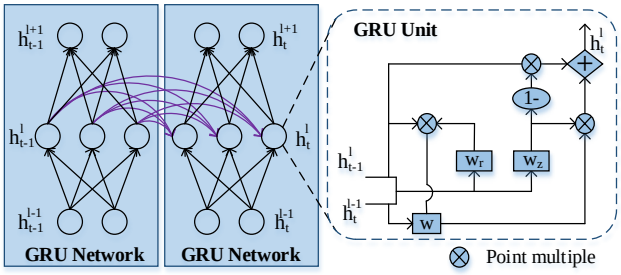


Figure 3. Architecture of gate recurrent unit network.

In addition to fully connect with the neurons in the previous hidden layer at time *t*, each neuron in the GRU hidden layer is fully connected to all neurons in the current hidden layer at time *t-1*. The l^{th} hidden layer output is computed by the following equation:

$$h_t^l = (1 - z_t) * h_{t-1}^l + z_t * \tilde{h}_t^l \quad (2)$$

where z_t is an update gate, and $\tilde{h}_t^l$ is the candidate memory information. Moreover, z_t is given by equation 3 controls how much information of the previous memory and the current memory will be forgotten or to be added.

$$z_t = sigmoid(W_z \cdot [h_{t-1}^l, h_t^{l-1}]) \quad (3)$$

The candidate information value $\tilde{h}_t^l$ is calculated by Equation 4:

$$\tilde{h}_t^l = \tanh(W \cdot [r_t * h_{t-1}^l, h_t^{l-1}]) \quad (4)$$

where r_t defined by Equation 5 is the GRU reset gate which efficiently resets the information in the memory.

$$r_t = sigmoid(W_r \cdot [h_{t-1}^l, h_t^{l-1}]) \quad (5)$$

Sigmoid and tanh are activate functions, expressed as:

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (7)$$

In comparison with the CNN fully connected layers, the calculated information by the GRUs model contains more the historical state, in which the neuron value at time *t* is not only determined by the data in the previous layer at time *t*, but also is determined by the data stored in the GRU cell at time *t-1* (as Equation 2).

3. HYBRID CNN-GRU MODEL

3.1 Model Architecture

Because in the same CNN fully connected layer, neural units are not connected to each other. The GRU networks replace CNN fully connected layers to transform the classification task into a sequence task, in which the classification results of each feature map are added to the next feature map classification calculation in the same hidden layer to improve the CNN recognition performance. The proposed CNN-GRU architecture is shown in Figure 4. Furthermore, the dropout method which can randomly delete some neural units in the hidden layer is used to avoid the possible over-fitting problem in the CNN-GRU model.

In CNN-GRU structure, the original CNN input layer, convolutional layers and pooling layers parameters with sizes as 1@32x32-C1:6@28x28-S2:6@14x14-C3:12@10x10S4:12@5x5 are unchanged to perform the features extraction. Each output feature map of the l^{th} convolutional layer is computed as follows:

$$O\_conv_j^l = sigmoid(\sum_{i=1}^{M} convn(a_i^{l-1} * k_{ij}^l) + b_j^l) \quad (8)$$

where a_i^{l-1} is the ith feature map in the total feature maps M of the $l\text{-}1^{th}$ previous layer, and k_{ij}^l denotes the l^{th} layer kernel. The value at the position of *(m, n)* in the convolution operation *convn* is defined as Equation 9.

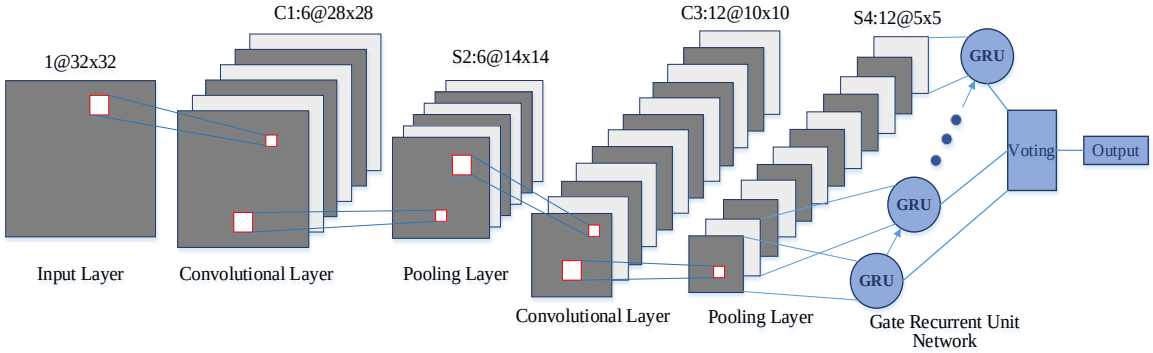


Figure 4. Structure of the hybrid CNN-GRU model.

$$convn(a_i^{l-1}, k_{ij}^l)[m][n] = \sum_{w=0}^{k-1}\sum_{l=0}^{k-1} k_{ij}^l[w][l] * a_i^{l-1}[m+w][n+l] \quad (9)$$

where $k=5$ is the convolutional kernel width. In the pooling layer, each output feature map value at the position of (m, n) is calculated by using average pooling method shown in equation 10, where a_j^{l-1} is the j^{th} feature map of the l-1 convolutional layer, $K=2$ is the pooling kernel width.

$$aver\_pool_j^l[m][n] = \frac{1}{4}\sum_{l,w=0}^{K-1} a_j^{l-1}(m+l, n+w) \quad (10)$$

Then, all features of each feature map in the final CNN pooling layer are connected to a corresponding GRU model. That means there are 12 GRU networks in total, and each GRU network sets three layers, including an input layer of 25 neurons, a hidden layer of 50 neurons, and an output layer of 10 neurons. CNN-GRU model uses Equation 2 instead of Equation 1 to estimate each neuron value in the hidden layer. Finally, the GRU outputs are activated by softmax function defined as Equation 11 to classify the handwritten digit. On the testing phases, all the GRU outputs have been voted to find out the final recognition result.

$$y_k = \frac{\exp(w_k h)}{\sum_{k'}\exp(w_{k'} h)} \quad (11)$$

3.2 Dropout for GRU

When the neural network system implements large-scale recognition task, the neural network system is prone to over-fitting due to the reasons of small database size, large sample interference, and so on. In this case, the testing recognition accuracy keeps going up with the training recognition accuracy, but it decreases after a certain number of iteration epochs. To solve this problem and promote recognition rate, the dropout method is used to delete some neural units in the hidden layer [19][20]. In CNN-GRU model, dropout method can be applied to feed-forward connection and recurrent connection of the GRU hidden layer with different dropout probability.

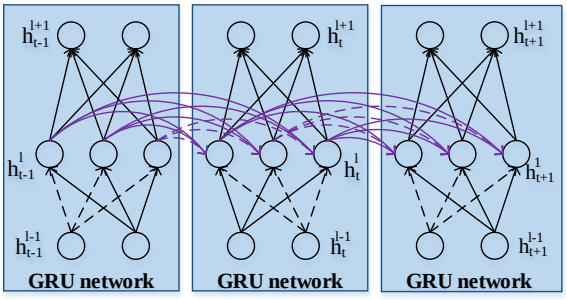


Figure 5. Dropout is applied to feed-forward connection and recurrent connection in GRU.

Figure 5 illustrates the dropout architecture in GRU where dropout is both applied to feed-forward connection and recurrent connection. When dropout with probabilities p_1 and p_2 is applied in feed-forward connection and recurrent connection respectively. Instead of using Equation 2-3, the candidate value and the output value of the hidden layer will be decided by following equations where m_i is a binary mask vector generated from the Bernoulli distribution with the p_i probability value at 0.

$$\tilde{h}_t^l = \tanh(W \cdot [r_t * h_{t-1}^l, h_t^{l-1}]) * m_1 \quad (12)$$

$$h_t^l = (1-z_t) * (h_{t-1}^l * m_2) + z_t * \tilde{h}_t^l \quad (13)$$

During the testing phases, all neural units of the hidden layer are hold on, and the candidate values and the output values of them are reduced by 1-p_i time as the following equations:

$$\tilde{h}_t^l = \tanh(W \cdot [r_t * h_{t-1}^l, h_t^{l-1}]) * (1-p_1) \quad (14)$$

$$h_t^l = (1-z_t) * (h_{t-1}^l * (1-p_2)) + z_t * \tilde{h}_t^l \quad (15)$$

Table 1 is the details of the proposed CNN-GRU model, where *ks* denotes kernel size; *ms* denote the output feature map size; *kn*, *mn*, *nn*, and *ng* denotes the number of kernels, feature maps, neurons and GRU units, respectively.

Table 1. Settings of the proposed CNN-GRU model

Layers	Kernels	Outputs
Conv-C1	ks=5x5, kn=6	ms=28x28, mn=6
Pool-S2	ks=2x2, kn=1	ms=14x14, mn=6
Conv-C3	ks=5x5, kn=72	ms=10x10, mn=12
Pool-S4	ks=2x2, kn=1	ms=5x5, mn=12
GRU_input	-	nn=25, ng=12
GRU_hidden	-	nn=50, ng=12, dropout
GRU_output	-	nn=10, ng=12

4. EXPERIMENTS

4.1 Database

CNN-GRU model was verified on the MNIST handwritten digit dataset [21]. It totally contains 70,000 images, in which 60,000 samples were used for training and 10,000 samples for testing. All the experiments were executed on MATLAB version R2017b, and CPU is Intel(R) Core(TM) i5-6500 with 3.2 GHz and 16 GB RAM.

4.2 Experimental Results

In experiments, the CNN-GRU was trained with different learning rate. Figure 6 shows the CNN-GRU training accuracy of all twelve

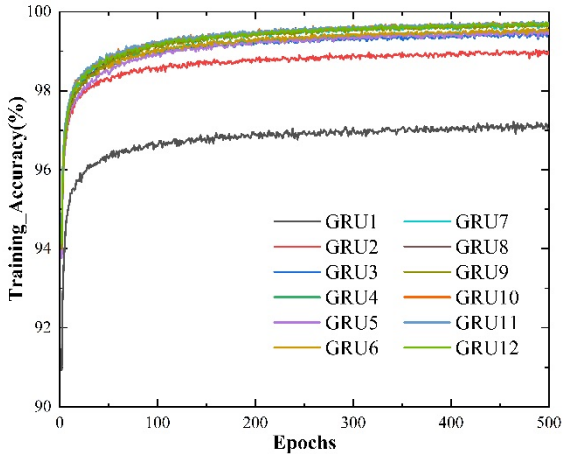
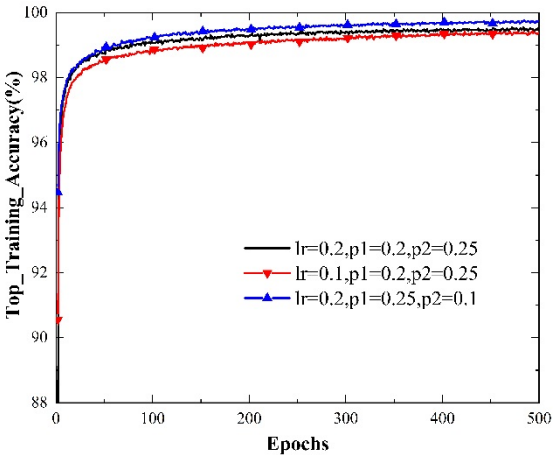


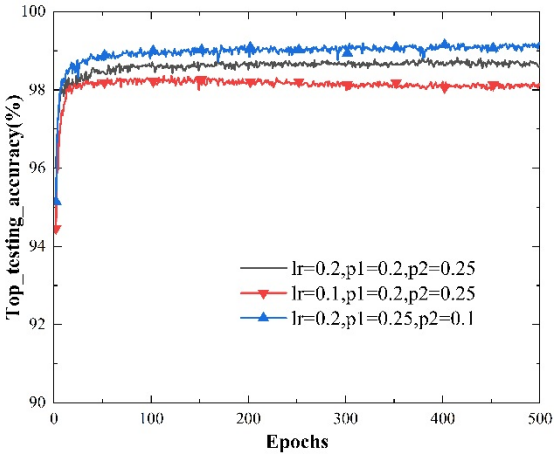
Figure 6. Training accuracy rate of CNN_GRU model.

GRU networks when setting 50 neurons in the GRU hidden layer. From Figure 6, by taking the advantage of the GRU sequence performance, the training recognition rates from the second GRU network are greatly improved from 97.18% to more 99%.

Experiments are also implemented with different learning rate lr and dropout probabilities p_1, p_2 to verify the performance of the proposed network. Figure 7 shows the changes of top training accuracy and top testing accuracy when the learning rate and dropout probabilities are changed. After 500 epochs, the highest training and testing recognition rates reaching up to 99.76% and 99.21% when the learning rate *lr* is 0.2 and the dropout probabilities p_1 and p_2 are 0.25 and 0.1, respectively.



(a) Top accuracy rate on training phases



(b) Top accuracy rate on testing phases

Figure 7. Comparison of CNN-GRU model with different parameter setting.

Table 2 shows the CNN-GRU model testing accuracy rates and running time compared with some different published methods. The best CNN-GRU model achieves 99.21% testing accuracy rate, 57.91 second training time, and 3.54 second testing time.

The CNN-GRU misrecognized digits are shown in Table 3 and Figure 8, in which the upper left label is the truth value, and the upper right label is the prediction value. The misrecognized images can be contributed to the following two main reasons. Firstly, the non-normative writing, superfluous or missing strokes, joined-up writing of the writer, and so on, lead to the several digits, which have similar shapes such as "3-5", "5-6"and "4-9" are easy to confuse when classifying them. Secondly, due to the images properties in themselves such as rough shape, fuzzy outline and intruder noise could also result in the misrecognition for these handwriting digits.

Table 2. Recognition accuracy rate on the MNIST dataset

Method	Reference	Accuracy rate (%)	Running time(s)	
			Train-ing	Test-ing
CNN	Calculated by authors	98.99	64.62	4.58
CNN	Lecun [11]	99.05	-	-
CNN_SVM	Maitra [13]	99.10	-	-
HOG_SVM	Ebrahimzadeh [5]	97.25	-	-
Q-ADBN	J.Qiaol [15]	99.18	58.67	21.46
CNN_GRU	This paper	99.21	57.91	3.54

Table 3. Confusion matrix of misrecognition

Truth	Recognition result									
	0	1	2	3	4	5	6	7	8	9
0							1		1	
1			1					1		
2	2							2		
3	1		2			9		2	2	1
4		1						1		9
5	1			4			8			
6	2	1			3	1				
7		1	2		2					1
8					2	1	1	2		1
9	3	1			6					

Moreover, sixty people were participated to recognize these above misrecognized digits. The correct recognition percentage is showed in the down label of Figure 8. The result indicates that even the human eyes are difficult to be recognized correctly.

5. CONCLUSION

In this paper, a hybrid CNN-GRU model for handwritten digit recognition is proposed. High performance of image feature is firstly extracted by the CNN, and then these extracted features are recognized by the GRU. In comparison with the other published methods, this model has two key advantages. The first, CNN ensures that the best features of the original image can be extracted using multiple convolutional and pooling layers. The second, GRU is a fast recognition method, which established a sequential relationship between features in the hidden layer.

Experiments on the MNIST database use two aspects, including the accuracy rate and running time to evaluate the proposed method performance. The result shows that the proposed method achieves 99.21% accuracy rate, 57.91 second training time, and 3.54 second testing time.

Figure 8. Misrecognized images of CNN-GRU model.

6. ACKNOWLEDGMENTS

This work was supported by the innovation wisdom base for wide bandgap semiconductor and micro-nano electronics of China (B12026), the National Natural Science Foundation of China (61076031), the Natural Science Foundation of Shaanxi Province, China (2016JM6067), Strategic international scientific and technological innovation cooperation key projects (2016YFE0207000).

7. REFERENCES

[1] Vajda, S., Santosh, K. C. 2016. A fast k-nearest neighbor classifier using unsupervised clustering. *Handbook of Recent Trends in Image Processing and Pattern Recognition (RTIP2R).* 709 (Dec. 2016), 185-193.

[2] Babu, U. R., Venkateswarlu, Y., and Chintha, A. K. 2014. Handwritten digit recognition using K-nearest neighbour classifier. In *World Congress on Computing and Communication Technologies*, Trichirappalli, India, April 2014, 60-65.

[3] Makkar, T., Kumar, Y., Dubey, A. K., Rocha Á. 2017. Analogizing time complexity of KNN and CNN in recognizing handwritten digits. *In International Conference on Image Information Processing (ICIIP)*, Shimla, India (Dec. 2017), 1-6.

[4] Lu, W. 2017. Handwritten digits recognition using PCA of histogram of oriented gradient. *In IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Victoria, BC, Canada, (Aug. 2017).

[5] Ebrahimzadeh, R., Jampour, M. 2014. Efficient handwritten digit recognition based on histogram of oriented gradients and SVM. *International Journal of Computer Applications*. 104, 9 (October 2014), 10-13.

[6] Drewnik, M., Winiarski, Z. P. 2017. SVM Kernel Configuration and Optimization for the Handwritten Digit Recognition. *Computer Information Systems and Industrial Management (CISIM)*. 10244 (June 2017), 87-98.

[7] Khan, H. A. 2017. MCS HOG features and SVM based handwritten digit recognition system. *Journal of Intelligent Learning Systems and Applications*. 9, 2 (2017), 21-33.

[8] Lauer, F., Suen, C. Y., and Bloch, G. 2007. A trainable feature extractor for handwritten digit recognition. *Pattern Recognition*. 40, 6 (June 2007), 1816 – 1824.

[9] Al-Omari, S., Sumari, P., Al-Taweel, S. 2009. Digital recognition using neural network. *Journal of Computer Science*. 5, 6 (June 2009), 427-434.

[10] Patil, V., and Shimpi, S. 2011. Handwritten english character recognition using neural network. *Computer Science and Engineering*. 41 (Nov. 2011), 5587-5591.

[11] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*. 86, 11 (Nov. 1998), 2278-2324.

[12] Niu, X. X., Suen, C. Y. 2012. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognition*. 45, 4 (2012), 1318-1325.

[13] Maitra, D. S., Bhattacharya, U., Parui, S. K. 2015. CNN based common approach to handwritten character recognition of multiple scripts. In *International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia (Aug. 2015), 1021-1025.

[14] Guo, Q., Wang, F. L., Lei, J., Tu, D., Li, G. H. 2016. Convolutional feature learning and Hybrid CNN-HMM for scene number recognition. *Neurocomputing*. 184, 5 (April 2016), 78-90.

[15] Qiao, J. F., Wang, G. M., Li, W. J., Chen, M. 2018. An adaptive deep Q-learning strategy for handwritten digit recognition. *Neural Networks*. 107 (Nov. 2018), 61-71.

[16] Hori, T., Watanabe, S., Zhang, Y., Chan, W. 2017. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM. *American Physical Society*, arxiv.org/abs/1706.02737 (June 2017).

[17] Cho, K., Merrienboer, B. V., Gulcehre, C. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Computer Science*. (Sep 2014).

[18] Hochreiter, S., Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*. 9, 8 (Nov. 1997), 1735-1780.

[19] Srivastava, N., Hinton, G., Krizhevsky, A., et al. 2014. Dropout: A simple way to prevent neural networks from over fitting. *The Journal of Machine Learning Research*. 15, 1 (2014), 1929-1958.

[20] Gal, Y., Ghahramani, Z. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*, Barcelona, Spain (2016).

[21] http://yann.lecun.com/exdb/mnist/, The MNIST database of handwritten digits.