International Journal of Innovative Computing, Information and Control
Volume **11**, Number **6**, December **2015**

pp. 2145–2158

DESIGN OF A CCNN ONLINE HANDWRITTEN DIGIT RECOGNIZER FOR A TEMPERATURE SENSING TERMINAL

Lei Wang[1,*], Jianfeng Wu[2] and Hongsheng Li[1]

[1]School of Automation
Nanjing Institute of Technology
No. 1, Hongjing Road, Jiangning Science Park, Nanjing 211167, P. R. China
[*]Corresponding author: zdhxwl@njit.edu.cn

[2]School of Instrument Science and Engineering
Southeast University
No. 2, Sipailou, Xuanwu District, Nanjing 210096, P. R. China
wjf@seu.edu.cn

Received May 2015; revised September 2015

Abstract. *We described a system which could recognize handwritten digits on a temperature sensing terminal. In this system, a handwritten digit was inputted as a sequence of temperature matrices, subjected to simple preprocessing, and then classified by a trainable cascade-correlation neural network (CCNN). Before classification, hybrid feature extraction was done to improve the recognition rate. The CCNN was trained on a set of 800 digits from approximately 40 different writers handwriting on the temperature sensing terminal, and tested on 200 digits from other writers handwriting on the temperature sensing terminal. Results of comparison experiments with three other classifiers indicated that the CCNN classifier achieved better results: a recognition rate of 99% with 1% rejection. The recognition rate of the online handwritten digit recognizer improved significantly for the temperature sensing terminal.*
Keywords: Temperature sensing terminal, Handwritten digit recognition, Cascade-correlation neural network (CCNN), Hybrid feature extraction

1. **Introduction.** Handwriting recognition has received remarkable attention in the field of character recognition [1-3]. Real-time online handwriting recognition systems [4] are currently under high demand due to the increasing popularity of handheld devices, such as smartphones, tablet PCs. As an important branch of handwriting recognition, handwritten digit recognition is still an open problem to be solved in the future.

The acquisition of handwritten digit information is the foundation of handwritten digit recognition. It requires specific devices which can capture the trajectory of a handwriting pen or a finger in real time. The piezoresistive touch panel [5] and the capacitive touch panel [6] have been widely used as traditional handwriting inputting terminals. The piezoresistive touch panel is sensitive to the resistive changes caused by touch pressure at different position. So a specialized pen is necessary for handwriting on the piezoresistive touch panel. Unlike the piezoresistive touch panel, the capacitive touch panel enables the user to handwrite directly on the panel without any intermediate device. However, the capacitive touch panel cannot be used under wet conditions due to the sensitivity to the distributed capacitance of the capacitive touch panel. Recently, Wu et al. [7] proposed a temperature sensing terminal, which could be used as a new type handwriting inputting terminal, and it was not sensitive to changes of the pressure, the distributed capacitance, and the environment humidity. So it was an attractive complement to traditional HCI interfaces. We addressed an online handwritten digit recognizer for the temperature sensing terminal in this paper. In the temperature sensing terminal, the sensitive panel had an 8×16 temperature sensor array, which was robust against the variation in the environment. However, the high real time requirement of the online handwritten digit recognition limited the spatial resolution of the handwriting inputting information, which brought difficulties to the online handwritten digit recognition.

The automatic handwritten digit recognition normally includes three steps: preprocessing, feature extraction, and classification. The performance of handwritten digit recognition largely depends on the feature extraction approach and the classification scheme. The feature extraction is one of the crucial steps in the recognition system. The common features include static features and dynamic features [8-10]. Static features are insensitive to the writing sequence of the digit, but are susceptible to local noise. Dynamic features are not sensitive to noise and small deformation of handwritten digits, but the different stroke order of digits will cause wrong recognitions. To enhance the judgment of digit features, the hybrid features including static features and dynamic features are extracted. Classification is another crucial step in the recognition system. To improve the performance of handwritten digit recognition systems, many classification methods have been proposed, such as statistical methods [11-17], rule inference methods [18,19], and neural network methods [20-25]. Statistical methods such as Bayesian method [11], Fisher method [12,13], k nearest neighbor (kNN) method [14,15], and support vector machine (SVM) method [16,17], are based on the assumption of the probability density. The disadvantage of the statistical methods is that the small probability event has been ignored [11]. The decision tree method [18] and the association rules method [19] are rule inference methods. The rule inference methods can make full use of the structure characteristics of the handwritten digits. However, they demand a lot of expert knowledge, which requires much manual work. The neural network can store information in distribution, process nonlinearity, and tolerate much fault. So many kinds of neural networks are used for handwritten digit recognition. The multilayer perceptron (MLP) [20,21] is the first neural network classifier tested on MNIST and it is the simplest neural network classifier used to handwritten digit recognition. The back-propagation neural network (BPNN) classifier [22,23] is overall superior in memory usage and classification time but it may provide "false positive" classifications when the input is not a digit. The radial basis function neural network (RBFNN) classifier [24] can provide the similar low error rate with the kNN classifiers. The RBFNN classifier requires more memory and more classification time. The number of hidden units of the neural network has a significant impact on the performance of the neural network classifier, but it is difficult to be determined. The cascade-correlation (CC) algorithm proposed by Fahlman and Lebiere [25] can determine the hidden units number automatically.

The main contributions of this paper can be summarized as follows. Firstly, the hybrid features exaction method is proposed for the online handwritten digit recognition. Both static features of the temperature matrix from the temperature sensing terminal and dynamic features of handwritten digits are considered, which contribute to the improvement of the handwritten digit recognition rate for the temperature sensing terminal. Secondly, we propose a CCNN online handwritten digit recognizer for the temperature sensing terminal. We examine the performance of the recognizer through three experiments on the handwriting inputting terminal. The experiment results indicate that CCNN recognizer based on hybrid features yields better recognition accuracy than three other recognizers.

The rest of this paper is organized as follows. Section 2 describes the handwritten digit recognition system for the temperature sensing terminal. Section 3 explains the CCNN digit recognition approach. Experiments and results are reported and discussed in Section 4. Section 5 derives conclusions.

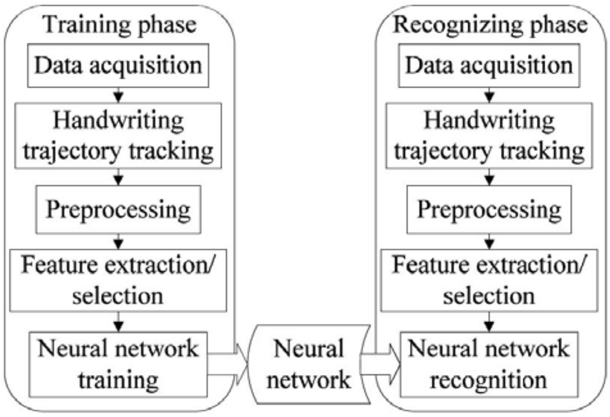


Figure 1. Overview of the handwritten digit recognition system

2. **Overview of the Handwritten Digit Recognition System.** Figure 1 shows an overview of the handwritten digit recognition system for the temperature sensing terminal.

The system consisted of a training phase and a recognizing phase. These phases involved all or some of the following steps: data acquisition, handwriting trajectory tracking, data preprocessing, feature extraction, neural network training and recognition. Firstly, a series of temperature data frames were acquired by the serial port of the computer from the temperature sensing terminal. Then the static frames before handwriting were eliminated. Next, the fingertip moving trajectory was tracked according to the point with the maximum value in each frame. After that, the typical features were extracted including static features and dynamic features. In order to improve the performance of recognition, these features were selected by the method of PCA. Finally, the features were trained or recognized by the CCNN.

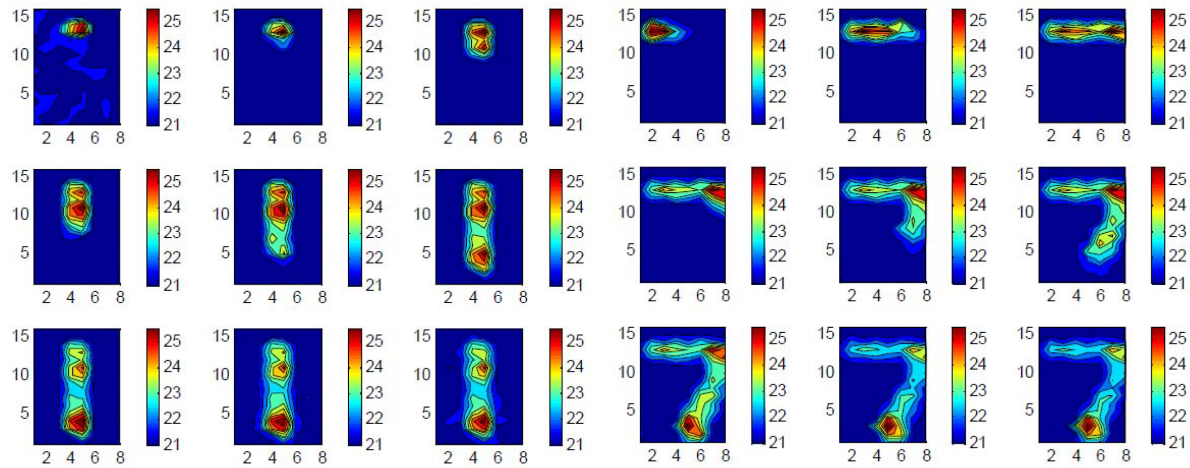
2.1. **Data acquisition.** We first captured the temperature data from the temperature sensing terminal. When the finger touched the temperature sensitive panel during the handwriting, the terminal transformed temperature data into a sequence of digital frames. Examples of a sequence of frames from the temperature sensitive panel were shown in Figure 2. Frames of digit character 1 captured at each sampling time were shown in Figure 2(a), and frames of digit character 7 captured at each sampling time were shown in Figure 2(b). The sampling interval was 0.25 second.

2.2. **Handwriting trajectory tracking.** The system repeatedly detected the position of the fingertip in each frame to acquire the stroke of handwritten digits. The position of the fingertip consisted of two-dimensional time-series data:

$$\text{Trackpot}_k = (x_k, y_k) \tag{1}$$

where (x_k, y_k) were coordinates of the fingertip on the image space detected at time t_k. The coordinates were determined by finding the maximum in each frame data. In Figure 3, the numbers of each digit tracking plot represented the frame number. For example, the position of number 1 was the position with the highest temperature value among the first frame. That was also the position of the fingertip touching the temperature sensitive panel. The frame numbers of every digit were different for the different handwriting speeds and the different trajectories of the digit character.

In the digital characters 0-9, character 4 and character 5 were written with two strokes while the other characters were written with only one stroke. Thus, the special trajectories were detected in the experiments of character 4 and character 5. In the process of handwriting trajectory tracking, there were some repeated points such as point 1 and



(a) Digit character 1 (b) Digit character 7

Figure 2. The sequences of digit character frame

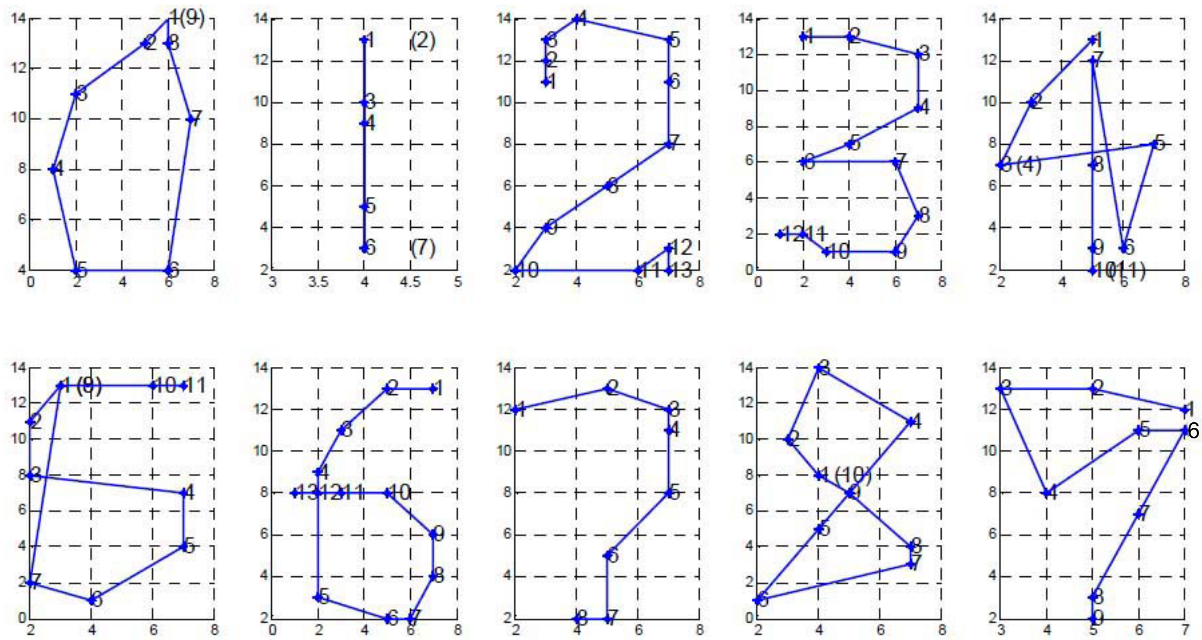


Figure 3. Stroke tracking of handwritten digits 0-9

point 9 in the figure of digit 0 due to the slower handwriting speed. The same things happened in handwritten digit characters such as 1, 4, 5, and 8.

From Figure 3, we knew that the handwriting trajectories of the digital characters 0-9 could be detected correctly for the temperature sensing terminal.

2.3. **Data preprocessing.** Before or during the handwriting trajectory tracking, data preprocessing was necessary, which included eliminating the static frames, deleting the repeated frames, and abandoning the invalid frames. The static frames were eliminated before the handwriting trajectory tracking, while the repeated frames and the invalid frames were deleted during the trajectory tracking.

2.4. **Feature extraction.** The purpose of feature extraction was to extract essential features for different handwritten digits. Both in the recognition process and the training process, features were measured and represented as feature vectors. Feature extraction was crucial for the recognition process. If features were well extracted, it would be easy to design a high performance classifier.

According to the mean temperature matrix of each handwritten digit, we extracted static features. According to the moving trajectory of the fingertip in the process of handwriting, we extracted local dynamic features and global dynamic features. To improve the validity of feature, we selected the essential features using PCA method.

2.4.1. *Static feature extraction.* The coarse grid feature was extracted for its good recognition rate and its insensitivity to noise. This feature was not sensitive to the deformation of digital partial stroke or isolated noise points. The mean temperature matrix of each handwritten digit was divided into several small regions; the dot density of each region was taken as the value of the coarse grid feature. The coarse grid feature extracted was expressed as a matrix with the size of 5×5, and then transformed into a feature vector with the size of 25×1.

2.4.2. *Dynamic feature extraction.* According to the moving trajectory of the fingertip in the process of handwriting, the local dynamic features and the global dynamic features were extracted. According to the results of many recognition experiments based on various dynamic features and reasoning rules, we chose the following dynamic features, which could effectively characterize the strokes of different handwritten digits.

Feature 1, the distance between the first point and the last point in the handwriting;

Feature 2, the direction of the vector from the first point to the last point;

Feature 3, the algebraic sum of each stroke vector angle;

Feature 4, the changes of each stroke vector angle;

Feature 5, the maximum absolute value of angle changes of each stroke vector;

Feature 6, the change times of rotation directions of all stroke vectors.

Based on the above features, we concluded ten rules to classify the handwritten digits 0-9. Examples of reasoning rules for handwritten digits classification were described as follows:

Condition 1: feature 1 is less than T_1 (a constant threshold);

Condition 2: feature 6 is less than T_6 (a constant threshold);

Rule 1: If condition 1 and condition 2 are both satisfied, then the digit is 0;

Rule 2: If condition 1 is satisfied but condition 2 is not satisfied, then the digit is 8.

The detailed descriptions of dynamic features and reasoning rules were shown in Appendix A.

2.4.3. *Feature selection.* Multi feature recognition could improve the recognition accuracy, but it would also increase the computational complexity of the recognition system. Meanwhile, too many non-essential features would lead to lower recognition adaptability. Therefore, feature selection was necessary. We used PCA method [26] for feature selection of handwritten digits. The essence of PCA was to project the sample data in high dimensional space to a low dimensional space through Karhunen-Loeve (KL) transformation [27].

The number of training samples was N, and every sample was reprented by a feature vetor of $M \times 1$. So N samples could be represented as a matrix of $M \times N$. In this paper, the input data of PCA were static features or dynamic features.

Assuming x_i was a column vector of sample i and the input sample was $X = (x_1, x_2, \ldots, x_N)$, we selected the component features with the steps as follows.

1) The vector of the average sample could be calculated with Equation (2).

$$\bar{X} = \sum_{i=1}^{N} x_i \tag{2}$$

2) The difference between the input sample x_i and the average sample $\bar{X}$ could be calculated with Equation (3).

$$d_i = x_i - \bar{X} \tag{3}$$

3) The covariance matrix of input samples could be calculated with Equation (4).

$$Cov = \frac{1}{N} DD^T = \frac{1}{N} \sum_{i=1}^{N} d_i d_i^T = \frac{1}{N} \sum_{i=1}^{N} \left(x_i - \bar{X}\right) \left(x_i - \bar{X}\right)^T \tag{4}$$

where $D = [d_1, d_2, \ldots, d_N]$.

According to the KL transform principle, the new coordinate system consisted of the eigenvectors with corresponding to the non-zero eigenvalues. To avoid the complexity of calculating eigenvectors and eigenvales directly, we used the Singular Value Decomposition (SVD) [27] method to calculate eigenvectors and eigenvales.

Using SVD method, we got the orthogonal eigenvectors u_i of matrix DD^T according to eigenvectors v_i with corresponding to eigenvalues λ_i of matrix $D^T D$ described by Equation (5) and Equation (6).

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_r > 0 \tag{5}$$

$$u_i = \frac{1}{\sqrt{\lambda_i}} D v_i \quad (i = 1, 2, \ldots, r) \tag{6}$$

Therefore, the eigen digit subspace was: $U = (u_1, u_2, \ldots, u_r)$.

4) The training samples were projected to the eigen digit subspace and the projection vectors were obtained:

$$P_i = U^T d_i \quad (i = 1, 2, \ldots, r) \tag{7}$$

5) The difference between test sample Y and the average feature $\bar{X}$ was projected to the eigen digit subspace, and the vector was:

$$P_Y = U^T (Y - \bar{X}) \tag{8}$$

2.5. **Neural network training and recognizing.** The training set of the network consisted of these feature samples extracted and selected. The training goal was to develop a network with multiple units which could classify all the training samples properly.

In process of the neural network training, the parameters were adjusted iteratively to meet the training goal of minimizing of the network error. The algorithm used in the training was cascade-correlation algorithm, which would be introduced in Section 3.

After network training, both the structure and the parameters were determined. The test feature samples were inputted to the trained CCNN, and the outputs of the CCNN were the recognition results.

The neural network used in this system was the cascade-correlation neural network (CCNN). The number of the hidden layer unit was concluded from learning rather than a constant, which reduced the network size meanwhile ensuring the smaller network error. The structure and the training method of the CCNN were described in detail in Section 3.

3. CCNN Used for Handwritten Digit Recognition.

3.1. Structure of the CCNN. The cascade architecture of the CCNN was illustrated in Figure 4(a). It began with some inputs and one or more output units, but with no hidden units. Every input unit was connected to every output unit by a connection with an adjustable weight. The output unit's activation function f was a symmetric sigmoid function (as shown in Formula (9)). The net input z_i of the output unit was the weighted sum of network input units x (as shown in Formula (10)).

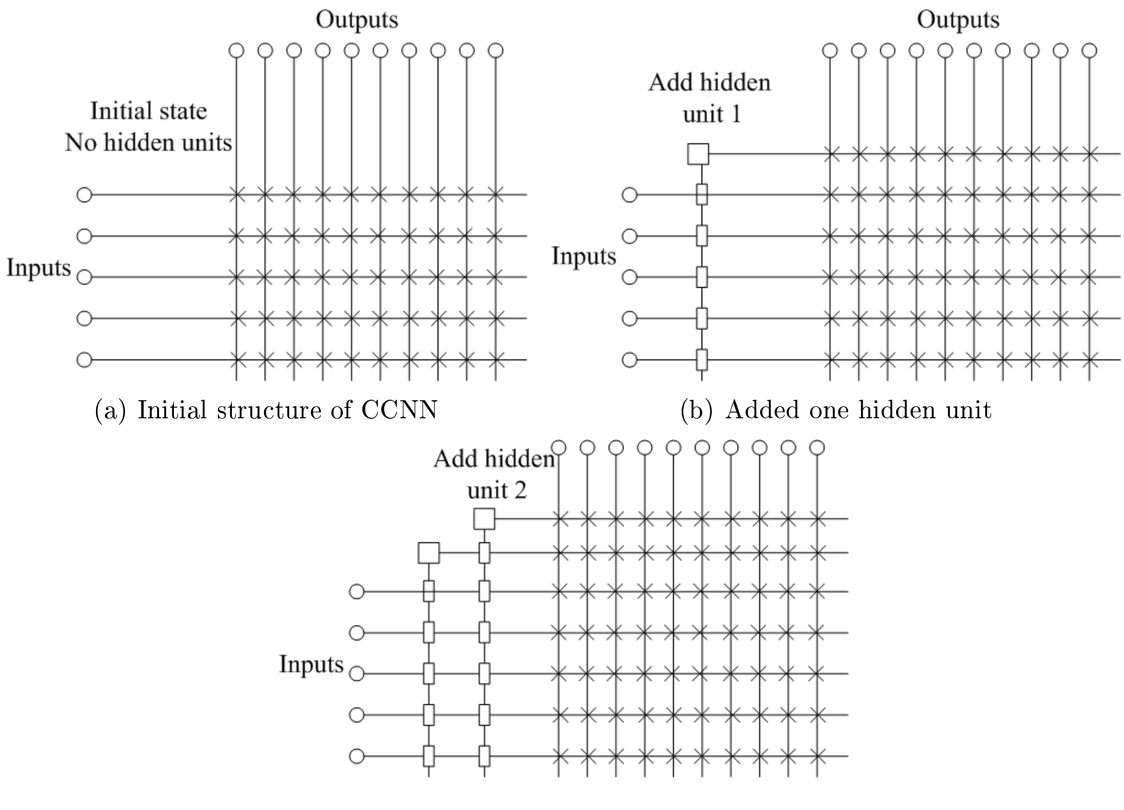
$$f_i(z_i) = \frac{1}{1 + \exp(-\beta_i z_i)} \tag{9}$$

$$z_i = w_i \cdot x - \theta_i \tag{10}$$

where β_i was the slope parameter of the symmetric sigmoid function, z_i was the net input of the output unit, w_i was the connection weight between the input unit and the output unit, and θ_i was the threshold value of the output unit.

When the initial neural network could not meet the requirement, we attempted to achieve this by adding a new hidden unit to the network (as shown in Figures 4(b) and 4(c)), using cascade-correlation algorithm described below. The new unit was added to the net and its input weights were frozen. Then all the output weights were trained once again. This cycle was repeated until the error was acceptably small (or until we gave up).

3.2. Cascade-correlation algorithm. Cascade-correlation algorithm combined two key ideas [25]. The first one was the cascade architecture, in which hidden units were added to the network one at a time and were not altered after they had been added. The second



(a) Initial structure of CCNN

(b) Added one hidden unit

(c) Added two hidden units

FIGURE 4. Structure of CCNN

one was the learning algorithm, which created and installed new hidden units. For each new hidden unit, we attempted to maximize the magnitude of the correlation between the new unit's output and the residual error signal that we were trying to eliminate.

The processing flow for the CC algorithm was briefly described as follows.

(1) Network initialization: some input and output units were structured without hidden units. The number of inputs and the number of outputs were determined by the I/O representation in the problem.

(2) Train weights connected to output units: the output units' weights were trained by the gradient descent algorithm, and the target function was shown in Formula (11)

$$J = \sum_{k=1}^{5} J_k = \sum_{k=1}^{5} \frac{1}{2} e_k^2 = \sum_{k=1}^{5} \frac{1}{2} \left(\hat{y}(k) - y(k)\right)^2 \tag{11}$$

(3) Judge the performance of the network: the error of the network was measured according to Formula (11). If the network's performance was satisfactory, the process would stop; else when no significant error reduction occurred after a certain number of training cycles (controlled by a "patience" parameter set by the user), the process would go to step (4); else it would go to step (2).

(4) Train the candidate hidden units: the candidate units received trainable input connections from all of the network's external inputs and from all pre-existing hidden units. Each candidate hidden unit was trained. The goal was to maximize C, as shown in Formula (12), the sum over all output units j of the magnitude of the correlation between y_h, the candidate unit's value, and e^j, the residual output error observed at unit j.

$$C = \sum_{j=1}^{q} \sum_{k=1}^{N} \left[y_h^j(k) - \bar{y}_h\right] \left[e^j(k) - \bar{E}\right] \tag{12}$$

where j was the network output at which the error was measured and k was the training pattern. The quantities $\bar{y}_h$ and $\bar{E}$ were the values of y_h and e^j, averaged over all patterns. $e^j(k) = \hat{y}^j(k) - y(k)$.

(5) Add a new hidden unit: the candidate unit with the maximum correlation was chosen as the new unit to the network, the weights of the input units were frozen, and the process would go to step (2).

4. **Experiments and Results.**

In this section, handwritten digit recognition experiments using the CCNN classifier were implemented. And the CCNN classifier was compared with the Fisher classifier, the RBR (rules based reasoning) classifier, and the BPNN classifier. The static features and the dynamic features were extracted in experiments. The validity of the CCNN classifiers was examined through three experiments: the first experiment was based on static features, the second one was based on dynamic features, and the last one was based on hybrid features (that included static and dynamic features).

4.1. **Experimental setup.**

Online handwritten digit samples were obtained from the temperature sensing terminal. The training data included 800 digits from approximately 40 different writers, and the testing data included 200 digits from other writers. The handwritten digits were composed of the digits from 0 to 9. Each digit was a temperature matrix of 8×16.

The handwritten digit recognition experiments were implemented with Matlab R2010b and tested on a 2.5GHz Intel Core4 PC with 4GB memory.

4.2. **Evaluation criteria.** We evaluated the recognition rate of each classifier using the *correct rate* (*CR*), the *error rate* (*ER*) and the *rejection rate* (*RR*). *CR* was the rate that a correct identify claim was accepted, ER was the rate that a false identity claim was accepted, and RR was the rate that a true user identity claim was falsely rejected. $CR = 1 - ER - RR$. We computed *CR*, *ER* and *RR* of each classifier individually to evaluate the performance of each classifier.

The consuming time of each classifier was evaluated by *training time* (*TT*) and *recognition time* (*RT*). *TT* was the consuming time in the training process and *RT* was the consuming time in the recognition (or testing) process.

4.3. **Experiments based on static features.** The coarse grid features were extracted with a 5×5 grid, and the feature space of the training data was expressed by a matrix of 25×800. The dimensionality of the feature space was reduced using PCA method with the threshold 0.9 and the 13 eigenvalues corresponding to the principle components were obtained according to the contribution of every eigenvalue. So the size of the feature vector of each training sample was converted to 13×1 from 25×1.

Then the CCNN classifier was trained on the training data and tested on the testing data. The algorithm used in the process of the output layer training was the gradient descent method. The maximum value of the training time was 2000.

For comparison, the Fisher classifier and BPNN classifier were tested based on the same static features. The RBR method had not been adopted because the static features were statistical ones and they had no appropriate physical meaning for reasoning. The statistical results of these classifiers in 50 time Monte Carlo tests were shown in Table 1. From Table 1, we found that the *correct rate* of the CCNN classifier was the highest and the Fisher classifier was the lowest. The *error rate* of the CCNN classifier was also the lowest. The *recognition time* of the BPNN classifier was the shortest one and the *recognition time* of the CCNN classifier was 0.2178 which was fast enough for the online handwritten digit recognition for a temperature sensing terminal. However, the *training time* of the CCNN classifier was the longest.

TABLE 1. Recognition results of the classifiers based on static features

Classifier	***CR*** (%)	***ER*** (%)	***RR*** (%)	***RT*** (s)	***TT*** (s)
Fisher	41.5	58.5	0	0.2845	0
RBR	–	–	–	–	–
BPNN	86.5	6	7.5	0.0026	3.5247
CCNN	92	4	4	0.2178	121.24

4.4. **Experiments based on dynamic features.** The dynamic features were extracted from many feature extracting experiments and six features were obtained ultimately. Based on these dynamic features, four classifiers were adopted for handwritten digit recognition for the temperature sensing terminal. The comparison results of four classifiers (Fisher, RBR, BPNN, and CCNN) were shown in Table 2. From Table 2, we found that the *correct rate* of the CCNN classifier was the highest and the Fisher classifier was the lowest. The *error rate* of the CCNN classifier was the lowest and *rejection rate* of it was 1.5%. The *recognition time* of the BPNN classifier was the shortest one and the *recognition time* of the CCNN classifier was 0.2081 which was fast enough for the online handwritten digit recognition for a temperature sensing terminal. However, the *training time* of the CCNN classifier was the longest.

Table 2. Recognition results of the classifiers based on dynamic features

Classifier	***CR*** (%)	***ER*** (%)	***RR*** (%)	***RT*** (s)	***TT*** (s)
Fisher	18	82	0	0.2639	0
RBR	67	33	0	0.0043	0
BPNN	68	10	22	0.0012	3.9435
CCNN	90.5	8	1.5	0.2081	231.44

4.5. **Experiments based on hybrid features.** The hybrid features included the coarse grid features with the size of 5×5 and the six dynamic features. The feature space of the training data was expressed by a matrix of 31×800. The dimensionality of the feature space was reduced using PCA method with the threshold 0.9 and the 15 eigenvalues corresponding to the principle components were obtained according to the contribution of every eigenvalue. So the size of the feature vector of each training sample was converted to 15×1 from 31×1.

Based on these selected hybrid features, three classifiers (Fisher, BPNN, and CCNN) were adopted for handwritten digit recognition. The comparison results of three classifiers were shown in Table 3. From Table 3, we found that the *correct rate* of the CCNN classifier was the highest and the Fisher classifier was the lowest. The *error rate* of the CCNN classifier was 0 and the *rejection rate* of it was 1%. The *recognition time* of the BPNN classifier was the shortest one and the *recognition time* of the CCNN classifier was 0.2432 which was fast enough for online handwritten digit recognition for a temperature sensing terminal. However, the *training time* of the CCNN was the longest.

Table 3. Recognition results of the classifiers based on hybrid features

Classifier	***CR*** (%)	***ER*** (%)	***RR*** (%)	***RT*** (s)	***TT*** (s)
Fisher	30.5	69.5	0	0.4586	0
RBR	–	–	–	–	–
BPNN	92.5	7.5	0	0.0027	3.8360
CCNN	99	0	1	0.2432	181.69

4.6. **Discussion.** From the above experiment results, we found that the CCNN recognizer had the highest correct recognition rate based on the three features, i.e., static features, dynamic features or hybrid features. However, the training time of the CCNN was the longest in three experiments, which was caused by the increase of hidden layer units and the corresponding higher complexity of the training algorithm. The training process of the CCNN was accomplished offline, so the long training time would not bring much trouble to the online handwritten digit recognition. In the recognition process, the computational complexity was not much and the recognition time was short enough for the online handwritten digit recognition for the temperature sensing terminal.

Furthermore, in order to compare the CCNN classifiers based on different features, Table 4 was provided.

From Table 4, we found that the CCNN classifier based on the hybrid features had the highest *correct rate* and the lowest *error rate* (= 0). The *rejection rate* of it was 1%, and the *recognition time* of it was 0.2432 second, which was a little slower than the other two classifiers, and it was fast enough for the online handwritten digit recognition. The *training time* of CCNN was long, so the training algorithm of the CCNN is in need of further research and improvement when online training is needed.

Table 4. Recognition results of CCNN classifiers based on three kinds of features

Features	***CR*** (%)	***ER*** (%)	***RR*** (%)	***RT*** (s)	***TT*** (s)
Static	92	8	0	0.2178	121.24
Dynamic	90.5	8	1.5	0.2081	231.44
Hybrid	99	0	1	0.2432	181.69

5. **Conclusions.** We have presented an effective approach to improve the correct rate of the online handwritten digit recognition by using hybrid feature extraction and CCNN classification. With experiments on 1000 handwritten digits on the handwriting inputting terminal based on temperature sensing the correct rate was improved, which demonstrated the effectiveness of our approach. The correct rate of the CCNN recognizer was 99%, and the rejection rate was 1%. The correct rate of the CCNN recognizer improved 5.5% based on the static features, 21.5% based on the dynamic features, and 6.5% based on the hybrid features than three other methods. The correct rate of the CCNN recognizer based on the hybrid features improved 7% or 8.5% than the CCNN recognizer based on static features or dynamic features.

Though meaningful results have been obtained, there are some unsolved problems to be considered for the future research. With the increasing of the hidden units' number of CCNN, the training process slows down dramatically for the large network structure. To realize online training of CCNN, we can limit the maximum of the hidden layer units' number of CCNN or search more quick algorithms for the weight training. In addition, the feature extraction method is worth researching in the future. One promising research direction is the deep learning used for feature extraction automatically in the handwritten digit recognition system for the temperature sensing terminal.

Acknowledgment. This study is partially supported the scientific research fund project of Nanjing Institute of Technology (No. YKJ201219, No. CKJB201405, No. CKJA201203). This study is also supported by the Open Fund of the Key Laboratory of remote measurement and control technology in Jiangsu Province (No. YCCK201401, YCCK201006), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 201300921 10060), the Chinese National Support Program (No. 2012BAI14B04), the Chinese Natural Science Foundation (No. 71101072, No. 61305011), and the Natural Science Foundation of Jiangsu Province (No. BK2012560). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] C. L. Liu, K. Nakashima and H. Sako, Handwritten digit recognition: Benchmarking of state-of-the-art techniques, *Pattern Recognition*, vol.36, pp.2271-2285, 2003.

[2] S. Impedovo, More than twenty years of advancements on frontiers in handwriting recognition, *Pattern Recognition*, vol.47, pp.916-928, 2014.

[3] R. Plamondon and S. N. Srihari, On-line and off-line handwriting recognition – A comprehensive survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.22, pp.63-84, 2000.

[4] K. C. Santosh and C. Nattee, A comprehensive survey on on-line handwritten recognition technology and its real application to the nepalese natural handwriting, *Kathmandu University Journal of Science, Engineering and Technology*, vol.6, pp.30-54, 2009.

[5] J. L. Aroyan, P. I. Gomes and J. Kent, Touch sensitive screen and method, *U.S. Patent No. 6,163,313.19*, 2000.

[6] I.-S. Yang and O.-K. Kwon, A touch controller using differential sensing method for on-cell capacitive touch screen panel systems, *IEEE Trans. Consumer Electronics*, vol.57, pp.1027-1032, 2011.

[7] J. F. Wu, L. Wang and J. Q. Li, *A Handwriting Input Device Based on the Thermal Cue of the Fingertip*, submitted, 2015.

[8] M. Mori, S. Uchida and H. Sakano, Global feature for online character recognition, *Pattern Recognition Letters*, vol.35, pp.142-148, 2014.
[9] M. Kherallah, L. Haddad, A. M. Alimi and A. Mitiche, On-line handwritten digit recognition based on trajectory and velocity modeling, *Pattern Recognition Letters*, vol.29, pp.580-594, 2008.
[10] M. Chikano, K. Kise, M. Iwamura, S. Uchida and S. Omachi, Recovery and localization of handwritings by a camera-pen based on tracking and document image retrieval, *Pattern Recognition Letters*, vol.35, pp.214-224, 2014.
[11] S.-J. Cho and J. H. Kim, Bayesian network modeling of strokes and their relationships for on-line handwriting recognition, *Pattern Recognition*, vol.37, pp.253-264, 2004.
[12] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, 2nd Edition, Wiley, New York, 2000.
[13] J. Yang, A. F. Frangi, J. Y. Yang, D. Zhang and Z. Jin, KPCA Plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.27, pp.230-244, 2005.
[14] D. Coomans and D. L. Massart, Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules, *Analytica Chimica Acta*, vol.136, pp.15-27, 1982.
[15] T. Cover and P. Hart, Nearest neighbor pattern classification, *IEEE Trans. Information Theory*, vol.13, pp.21-27, 1967.
[16] A. R. Ahmad, C. Viard-Gaudin, M. Khalia et al., Online handwriting recognition using support vector machine, *Proc. of the 2nd International Conference on Artificial Intelligence in Engineering & Technology*, Kota Kinabalu, Sabah, Malaysia, pp.250-256, 2004.
[17] X. X. Niu and C. Y. Suen, A novel hybrid CNN-SVM classifier for recognizing handwritten digits, *Pattern Recognition*, vol.45, pp.1318-1325, 2012.
[18] K. J. Wilder, *Decision Tree Algorithms for Handwritten Digit Recognition*, University of Massachusetts Amherst, ProQuest, UMI Dissertations Publishing, 1998.
[19] R. Agrawal, H. Mannila, R. Srikant et al., Fast discovery of association rules, *Advances in Knowledge Discovery and Data Mining*, pp.307-328, 1996.
[20] Y. L. Cun, A learning scheme for asymmetric threshold networks, *Proc. of Cognitiva*, Paris, France, pp.599-604, 1985.
[21] S. Basu, N. Das, R. Sarkar et al., An MLP based approach for recognition of handwritten 'Bangla' numerals, *Proc. of the 2nd Indian International Conference on Artificial Intelligence*, Pune, pp.407-417, 2005.
[22] Y. L. Cun, B. Boser, J. S. Denker et al., Handwritten digit recognition with a back-propagation network, *Advances in Neural Information Processing*, Denver, Colorado, USA, 1989.
[23] Y. L. Cun, Handwritten digit recognition using k nearest-neighbor, radial-basis function, and back-propagation neural networks, *Neural Computation*, pp.440-449, 1991.
[24] B. Lemarie, Radial basis function network for handwritten digit recognition, *Optical Engineering and Photonics in Aerospace Sensing*, International Society for Optics and Photonics, pp.645-652, 1993.
[25] S. E. Fahlman and C. Lebiere, The cascade-correlation learning architecture, *Carnegie Mellon University Technical Report CMU-CS-90-100*, 1991.
[26] J. Shlens, A tutorial on principal component analysis, *arXiv preprint arXiv:1404.1100*, 2014.
[27] J. Yang and J. Y. Yang, Generalized K-L transform based combined feature extraction, *Pattern Recognition*, vol.35, pp.295-297, 2002.

Appendix A: Dynamic Feature Recognition Based on Rules.

A1. The dynamic features extracted were described as Table A1.

A2. The conditions used in reasoning in the handwritten digit recognition based on dynamic features were described as Table A2.

A3. The reasoning process for the handwritten digit recognition based on the dynamic features was shown in Figure A1.

A4. According to the reasoning process for the handwritten digit recognition based on the dynamic features, ten rules were concluded and shown in Table A3. In these rules, one of them was corresponding to the recognizing process for one handwritten digit.

TABLE A1. Dynamic features of handwritten digits

Feature number	Physical significance	Feature symbol
1	the distance between the first point and the last point in the handwriting;	f_1
2	the direction of the vector from the first point to the last point;	f_2
3	the algebraic sum of each stroke vector angle;	f_3
4	the changes of each stroke vector angle;	f_4
5	the maximum absolute value of angle changes of each stroke vector;	f_5
6	the change times of rotation directions of all stroke vectors.	f_6

TABLE A2. The conditions for reasoning in the handwritten digit recognition based on the dynamic features

Condition number	Physical significance	Condition symbol
1	$f_1 < T_1$	C_1
2	$f_6 > T_6$	C_2
3	$f_2 < T_2$	C_3
4	$f_3 < T_3$	C_4
5	$\|f_2 - 90^0\| < T_2$	C_5
6	The number of components on the condition of $f_4 > T_4$ is greater than 1.	C_6
7	f_3 is located in the third quadrant.	C_7
8	$f_5 > T_5$	C_8
Notation	T_1, T_2, T_3, T_4, T_5, T_6 denote six threshold values for six characteristic parameters respectively.	

TABLE A3. Reasoning rules for the handwritten digit recognition based on the dynamic features

Rule number	Physical significance	Rule symbol
1	If C_1 is true and C_2 is true, then the digit is 8	R_1
2	If C_1 is true and C_2 is false, then the digit is 0	R_2
3	If C_1 is false and C_3 is true, then the digit is 5	R_3
4	If C_1 and C_3 are false, C_4 is true, then the digit is 1	R_4
5	If C_1, C_3 and C_4 are false, C_5 and C_6 are true, then the digit is 4	R_5
6	If C_1, C_3, C_4 and C_6 are false, C_5 is true, then the digit is 3	R_6
7	If C_1, C_3, C_4 and C_5 are false, C_7 and C_8 are true, then the digit is 6	R_7
8	If C_1, C_3, C_4, C_5 and C_8 are false, C_7 is true, then the digit is 9	R_8
9	If C_1, C_3, C_4, C_5 and C_7 are false, C_8 is true, then the digit is 7	R_9
10	If C_1, C_3, C_4, C_5, C_7 and C_8 are false, then the digit is 2	R_{10}

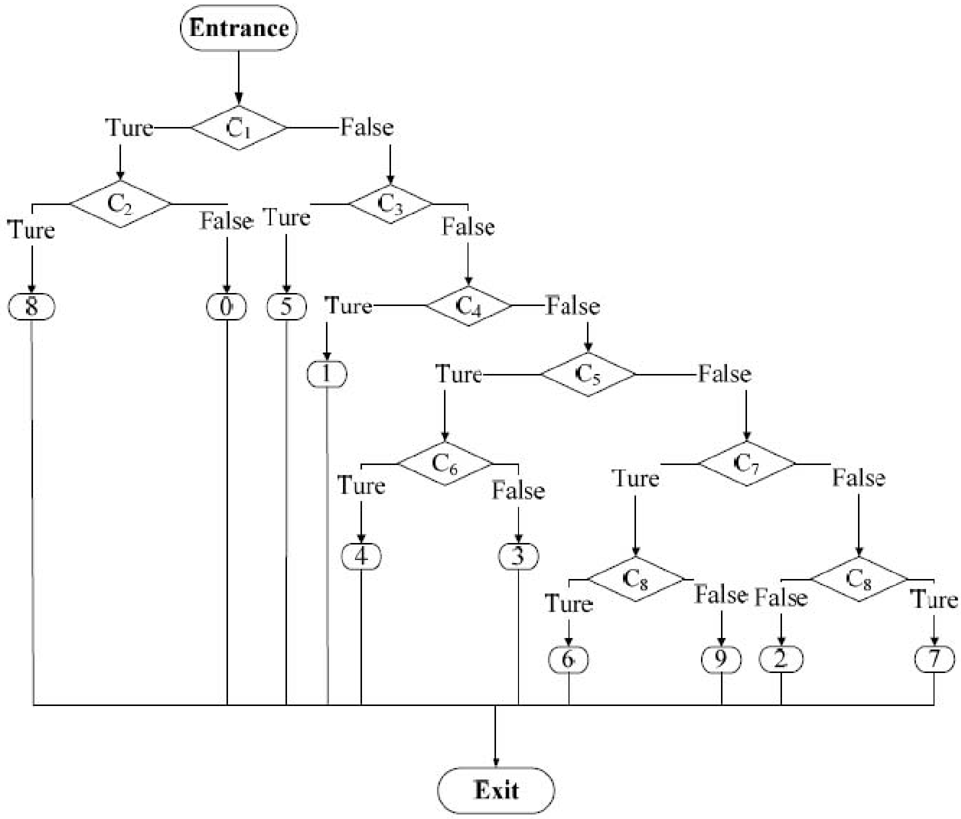


Figure A1. Reasoning process for the handwritten digit recognition based on the dynamic features