An End-to-End Classifier Based on CNN for In-Air Handwritten-Chinese-Character Recognition

Mianjun Hu, Xiwen Qu *, Jun Huang and Xuangou Wu

School of Computer Science and Technology, Anhui University of Technology, Maanshan 243032, China; hu791214407@163.com (M.H.); huangjun.cs@ahut.edu.cn (J.H.); wuxgou@ahut.edu.cn (X.W.)

* Correspondence: qxw_ahut@ahut.edu.cn

Abstract: A convolutional neural network (CNN) has been successfully applied to in-air handwritten-Chinese-character recognition (IAHCCR). However, the existing models based on CNN for IAHCCR need to convert the coordinate sequence of a character into images. This conversion process increases training and classifying time, and leads to the loss of information. In order to solve this problem, we propose an end-to-end classifier based on CNN for IAHCCR in this paper, which, to knowledge, is novel for online handwritten-Chinese-character recognition (OLHCCR). Specifically, our model based on CNN directly takes the original coordinate sequence of an in-air handwritten-Chinese-character as input, and the output of the full connection layer is pooled by global average pooling to form a fixed-size feature vector, which is sent to softmax for classification. Our model can not only directly process coordinate sequences such as the models based on recurrent neural network (RNN), but can also obtain the global structure information of characters. We conducted experiments on two datasets, IAHCC-UCAS2016 and SCUT-COUCH2009. The experimental results show a comparison with existing CNN models based on image processing or RNN-based methods; our method does not require data augmentation techniques nor an ensemble of multiple trained models, and only uses a more compact structure to obtain higher recognition accuracy.

Keywords: convolutional neural networks; in-air handwritten-Chinese-character recognition; end-to-end classifier; online handwritten-Chinese-character recognition; global average pooling

1. Introduction

In the field of image processing and pattern recognition, the recognition of online handwritten characters is important work and many impressive achievements have been achieved [1–5]. The data processed by online handwriting recognition is a series of coordinate sequences, which are usually used in human–computer interaction devices, such as handwriting tablets, smart phones and pads. The input range of the traditional human–computer interaction method is limited by the size of the touch device, and damage to the local area will make the entire device unusable. In this context, a new way of human–computer interaction, vision-based and in-handwriting, has attracted more and more researchers’ interest [6–8], e.g., we can use in-air handwriting to switch TV channels remotely or adjust the temperature of an air conditioner. Compared with the traditional online handwriting a using touch pad or wearable device, visual-based in-air handwriting experiences fewer space constraints and allows the writers to freely write in the air; the generated character has no pen-lift information and is finished in one stroke. The jitter of the stroke of a character is quite serious and the strokes overlap each other. These characteristics of in-air handwriting characters bring more difficulties for IAHCCR. Some examples of in-air handwriting Chinese characters and traditional handwritten Chinese characters can be seen in Figure 1.

As in-air handwriting is a new development of traditional online handwriting, the methods used for OLHCCR are available for IAHCCR. Traditional OLHCCR methods do
not directly recognize the original coordinate sequence of an online handwritten Chinese character, but extract features from a preprocessed coordinate sequence according to specific classification rules and specific domain knowledge \[9,10\]. Before deep learning was introduced into OLHCCR, statistical features, classification algorithms based on statistical features and preprocessing algorithms had always been the research hotspots of OLHCCR, and showed excellent performance in OLHCCR \[11,12\]. Related methods have been introduced into IAHCCR, e.g., linear or nonlinear normalization \[9,11\], eight-directional features \[10\], modified quadratic discriminant functions (MQDF \[13\], etc.

In recent years, deep learning has achieved great success in the fields of computer vision and pattern recognition \[14–16\], and also been successfully applied to OLHCCR \[17–20\]. Compared with the above-mentioned traditional classification models, the models based on deep learning show an overwhelming advantage in OLHCCR. However, to our knowledge, all the existing models based on CNN for OLHCCR do not directly recognize the coordinate sequence of an online handwritten Chinese character, but convert the coordinate sequence into images or vectors \[2,3,21–23\]. As shown in Figure 2, this conversion process not only causes the loss of training and recognition time, but also only utilizes the spatial information of characters and loses the temporal information of the coordinate sequence, so it is difficult to obtain a higher recognition rate.

![Figure 1](image1.png)  
(a) In-air handwritten Chinese characters. (b) Traditional handwritten Chinese characters.

![Figure 2](image2.png)  
Figure 2. Examples of in-air handwritten Chinese characters.

Since the CNN based model uses a large-scale convolution kernel, it often requires a large number of training patterns and leads to more memory consumption. Unlike the CNN models, the models based on RNN can directly process the coordinate sequences of online handwritten Chinese characters and outperform most CNN structures \[4,24\]. Although RNN-based models are suitable for processing sequence data, they are less suitable for processing long sequence data than CNN and ignore the global structures of online handwritten Chinese characters. In-air handwritten Chinese characters usually contain hundreds of points, so RNN-based models will consume a lot of computation time.

Based on the above analysis, we propose an end-to-end classifier based on CNN for IAHCCR in this paper which has the advantages of both CNN and RNN. First, we directly use the preprocessed coordinate sequences of online handwritten Chinese characters as the input of the network. Then, the coordinate sequences are converted into one-dimensional feature maps through the first layer of convolution, and then the range of the receptive
field is expanded by stacking the number of convolution layers to obtain the contextual
connection. Finally, global average pooling is applied to the output of the convolutional
layer to obtain a fixed-size feature vector, which is sent to the fully connected layer to extract
features and use softmax for classification. The end-to-end CNN can directly recognize
the coordinate sequences of online handwritten Chinese characters. Compared with the
existing CNN models, the end-to-end CNN does not need to convert the original data into
images, nor does it need to design features combined with specific domain knowledge.
Due to selecting the convolution kernel with a smaller scale, the end-to-end CNN needs
fewer parameters and occupies less memory. Compared with the RNN-based models, the
end-to-end CNN can learn the global information of online handwritten Chinese characters
and adapt to different lengths of coordinate sequences.

The rest of the paper is organized as follows. Section 2 briefly introduces the related
works. Section 3 introduces the proposed method in detail. The experimental results are
reported in Section 4. We conclude this paper in Section 5.

2. Related Works

Research on IAHCCR has been underway for several years, and a complete recognition
method usually consists of three stages: preprocessing, feature extraction and classification.
The work related to different stages will be described in detail below.

In the preprocessing stage, as mentioned in Section 1, IAHCCR can use the method of
OLHCRR. Character normalization can reduce within-class variation and improve recogni-
tion accuracy. Traditional methods require the normalization of characters to a uniform size.
Linear normalization, which causes character-shape changes, has been superseded by other
methods [11]. For example, nonlinear normalization [9], pseudo-2D normalization and
line-density projection interpolation [11]. End-to-end methods do not require normalization
to a fixed-size vector but, rather, normalize the distribution of coordinate sequences [24].

Feature extraction and classification are two separate stages in the traditional IAHCCR
method. In the stage of manual feature extraction, compared with the character image
drawn using the original coordinates, the method of decomposing local strokes into differ-
et directions to form multiple feature maps achieves higher recognition accuracy, such as
eight-directional feature maps [10]. Especially for IAHCCR, the whole Chinese character
is more like a curve function defined on a two-dimensional plane that can be expanded
by Taylor. Qu et al. [25] proposed higher-order directional features. Representing Chinese
characters as directional features has been the standard approach for a long time. Classi-
sifiers are also very important for IAHCCR. To further improve the recognition rate and
recognition speed, the learning vector quantization technique-based [26] multi-level classi-
fication technique is reported for IAHCCR in [27]. Qu et al. introduced locality-sensitive
sparse representation-based classifiers (LSRC) [28] into IAHCCR, and achieved a higher
recognition rate than MQDF [25]. In order to further improve the recognition accuracy of
LSRC, a loss function is designed to minimize the reconstruction error of each training
pattern and make the reconstruction of each training pattern as close as possible to the
optimized prototype of its class is suggested, which significantly improves recognition
accuracy [29].

As deep-learning techniques have made great achievements in other fields, the deep
neural network was introduced into IAHCCR. In the deep-learning-based IAHCCR algo-



Ref. [4] proposed an RNN system
with two new computing architectures added. Table 1 summarizes related works. In this paper, we combine the advantages of CNN and RNN to propose an end-to-end CNN model for directly recognizing sequences.

**Table 1. Summary of related works.**

| Methods                                | Types                  | Limitations                                                                 |
|----------------------------------------|------------------------|-----------------------------------------------------------------------------|
| Linear normalization [11]              | Preprocessing          | Neglecting the density of handwritten character sequence coordinates can lead to severe shape distortion |
| Nonlinear normalization [9]            | Preprocessing          | Unable to correct skew, local width or height imbalances in handwritten characters |
| Pseudo-2D normalization and line density projection interpolation [11] | Preprocessing          | Usually dealing with image features                                           |
| Coordinate normalization [24]          | Preprocessing          | Usually dealing with sequence features                                       |
| Eight-directional feature maps [10]    | Feature extraction     | Usually applied to OLHCCR                                                    |
| Higher-order directional features. [25] | Feature extraction     | Usually applied to IAHCCR                                                    |
| Learning-vector-quantization-technique-based multi-level classifier [27] | Classifier             | High computational cost and storage consumption                              |
| MQDF [25]                              | Classifier             | Low recognition accuracy and high storage cost                               |
| LSRC [28]                              | Classifier             | Difficulty obtaining discriminative features and low recognition accuracy    |
| Locality-sensitive sparse representation toward optimized prototype classifier [29] | Classifier             | Low recognition accuracy and high storage cost                               |
| Nine-layer convolutional neural network model combined with data-augmentation technology [23] | Classifier             | Relying on manual feature extraction and large amounts of data               |
| End-to-end recognizer based on recurrent neural networks [20] | Classifier             | Less efficient computational structure                                        |
| RNN system with two new computing architectures added [4] | Classifier             | It has difficulty effectively learning the global spatial features of sequences |

3. Proposed Method

Like other end-to-end recognition methods, the end-to-end CNN method consists of two parts, preprocessing and model architecture.

3.1. Preprocessing

As the writing styles of writers vary widely, the structure, position, shape, sampling-point density and stroke order of the finished in-air Chinese characters are different. These varied intra-class structures and the confusion between similar characters always result in a reduction in recognition accuracy [24]. In this paper, the primary purpose of the preprocessing is to eliminate redundant points and standardize the distribution of coordinate points, so as to improve the recognition accuracy for IAHCCR. The preprocessing steps in this paper are summarized as follows: (1) Remove redundant points in the coordinate sequence of in-air handwritten Chinese characters. (2) Normalize the coordinates to a unified coordinate system.

3.1.1. Remove Redundant Points

Any given in-air handwritten character \( P \) can be represented by its coordinate sequence as

\[
P = [(x_1, y_1), \ldots, (x_t, y_t)]
\]
where \( x_t \) and \( y_t \) are the XY coordinates of the \( t \)th point of \( P \); and \( t = 1, \ldots, T \), \( T \) is the number of the coordinate points of \( P \). For the coordinate sequence of the character, except for the first point, if the Euclidean distance between the \( t \)th point \((x_t, y_t)\) and its adjacent point \((x_{t-1}, y_{t-1})\) is less than the given threshold \( L \), i.e.,

\[
\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} < L
\]  

(2)

then the point \((x_t, y_t)\) is deleted as a redundant point. In the following experiments, \( L \) varies corresponding to different handwritten character \( P \), and can be computed by \( 0.015 \times \max\{h, w\} \), where \( h \) and \( w \) are the space height and width of \( P \), respectively.

### 3.1.2. Normalize Coordinates

Since in-air handwriting has fewer space constraints and an unstable writing position, the positions of the written Chinese characters vary: some are higher, some are lower, some are left and some are right. The end-to-end CNN directly takes a coordinate sequence as input, so the variation in coordinate-points distribution will greatly reduce the recognition rate. In order to decrease the variation in the spatial sizes and positions of characters, we employed coordinates normalization, following the method presented in [24]. Let \( l_t = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \) be the distance between two consecutive points \((p_t = (x_t, y_t)\) and \( p_{t-1} = (x_{t-1}, y_{t-1})\)). In order to normalize the coordinates to a standard interval, it is first necessary to estimate the mean of the coordinates projected on the coordinate axis. We can calculate the mean \( \mu_x \) and \( \mu_y \) of XY coordinates, respectively, by

\[
\mu_x = \frac{1}{2} \sum_{t=2}^{T} l_t (x_t + x_{t-1}) \sum_{t=2}^{T} l_t
\]  

(3)

\[
\mu_y = \frac{1}{2} \sum_{t=2}^{T} l_t (y_t + y_{t-1}) \sum_{t=2}^{T} l_t
\]  

(4)

The standard deviation \( \delta_x \) on the x axis can be estimated as

\[
\delta_x = \sqrt{\frac{\sum_{t=2}^{T} l_t [(x_t - \mu_x)^2 + (x_t - \mu_x)(x_{t-1} - \mu_x) + (x_{t-1} - \mu_x)^2]}{3 \times \sum_{t=2}^{T} l_t}}
\]  

(5)

The normalized trajectory needs to keep the original character shape and stroke writing direction, the characters are only scaled by the standard deviation \( \delta_x \). For the \( t \)th point of \( P \), we can obtain the normalized point \((\bar{x}_t, \bar{y}_t)\) by

\[
\bar{x}_t = \frac{x_t - \mu_x}{\delta_x}, \quad \bar{y}_t = \frac{y_t - \mu_y}{\delta_x}
\]  

(6)

Some examples of processing through the above steps are shown in Figure 3. From Figure 3, we can see that the processed coordinates are evenly distributed on both sides of \((0, 0)\) and the number of points has been reduced from 723 to 355.

### 3.2. Designing End-to-End CNN Architecture

For the proposed architecture, a brief overview is first given. As shown in Figure 4, the preprocessed sequence of coordinates is used as input to the recognizer. A larger receptive field can be obtained by stacking convolutional layers (Conv1 and Conv2) with small convolution kernels and reducing the sequence length by downsampling, so that more discriminative features can be learned. The fully connected layer (FC) requires a fixed-size feature vector input, but the length of the coordinate sequence is variable. Therefore, it is necessary to average over the coordinate sequence to obtain a fixed-size feature vector.
For the output of the convolutional layer, the sequence mean $m_i$ of each feature map $Z_i = [z_{i1}, \ldots, z_{iT}, \ldots, z_{iT}]$ is estimated by

$$m_i = \frac{1}{T} \sum_{t=1}^{T} z_{it}$$  \hspace{1cm} (7)

Since the number of output feature maps is fixed, we can combine these means into a fixed-size feature vector $f = [m_1, \ldots, m_c, \ldots, m_C]$, where $C$ is the number of feature maps. Finally, the softmax function is used to estimate the probability distribution of all classes. Next, we will introduce the specific configuration of end-to-end CNN in detail.

**Figure 3.** Example of normalization. The character (left) before the normalization and that (right) after the normalization are given, respectively.

**Figure 4.** Overview of end-to-end convolutional neural network architecture. The stride of the convolutional layer shown in the figure is 1, and the convolution kernel size is 3.

The configuration of the end-to-end classifier can be seen in Figure 5a. In Figure 5a, “Conv k:2 × 3, s:1, 64” denotes that the kernel size a convolutional layer is 2 × 3, the stride is 1 and the number of feature maps is 64, respectively. PReLU stands for the parametric rectified linear unit [31]. “Max-pool k:1 × 2 s:2” denotes that the size of max pooling is 1 × 2, and the stride is 2. “Dropout 0.2” denotes that the dropout rate is 0.2 [32]. “FC 160” denotes a fully connected layer, and the number of channels is 160. “Block $N$” represents a residual block, which is illustrated in Figure 5b, $N = 64, 128, 256$.

In more detail, the preprocessed in-air handwritten character $P = [p_1, \ldots, p_T, \ldots, p_T] \in \mathbb{R}^{2 \times T}$ are directly used as the input of the classifier, $T$ varies according to each different character. In order to identify the coordinate sequence, the network needs to stack multiple convolutional layers to expand the receptive field, so that the model can extract as much spatial and temporal information of the sequence as possible. $P$ is first processed using a convolutional layer in the time dimension to obtain a series of feature maps of $1 \times T_1$, $T_1$ changes with $T$, the kernel size is $2 \times 3$ and the stride is 1. After that, dropout is employed to avoid overfitting, and the max pooling is used to reduce the sequence feature length, which can further increase the receptive field range and reduce the impact of zero padding.
on sequence recognition. Since the constructed network is very deep, the residual link [33] can efficiently train the network. As shown in Figure 5b, we construct a residual block which contains a total of three convolutional layers. In Figure 5b, “⊕” denotes the element-wise sum, three convolutional layers are denoted by “Conv1”, “Conv2”, and “Conv3”, respectively. Conv1 and Conv2 are used to directly extract sequence features with kernel size 1 × 3 and stride 1. Conv3 is designed to make the feature size of the input and output the same during the connection operation, the size of the convolution kernel is 1 × 1, and the stride is 1. The residual block is repeated until the number of feature maps is increased from 64 to 256. Then, global average pooling (GAP) [34] is employed to obtain a fixed-size feature vector f. Finally, f is input into the fully connected layer (FC) and softmax is used to classify it.

![Figure 5. (a) The configuration of end-to-end convolutional neural network. (b) The architecture of residual convolution block.](image)

**4. Experiments**

**4.1. Datasets**

We conducted experiments on the IAHCC-UCAS2016 [23] which is an in-air handwritten Chinese character dataset and the GB1 dataset in the SCUT-COUCH2009 [35] database, which is a traditional handwritten Chinese character dataset. The samples in IAHCC-UCAS2016 consist of projections on a 2D plane of a sequence of 3D coordinates recorded by a sensor worn on the fingertip. The samples in the GB1 dataset are the trajectory coordinates written directly on the tablet. Both datasets are publicly available. The GB1 dataset involves 3755 character classes of the first level set of GB2312-80 and each class has 188 patterns. The IAHCC-UCAS2016 covers 3811 Chinese character classes, and each class contains 115 samples. For each class, 80% were randomly selected as training sets and the remaining 20% as test sets.

**4.2. Model Training Strategy**

Our network structure was implemented based on PyTorch, initialized with the default parameters of the framework. The optimizer used Adam; the initial learning rate was set
to 0.001. The learning rate was decreased when the accuracy on the training set no longer increased or increased slowly, with a decay rate of 0.1. All experiments were conducted on RTX-2080ti.

4.3. Comparison Experiments

In order to determine the appropriate mini-batch size under the condition that the learning rate is set to 0.001, the mini-batch size was set to 64, 128, 256 and 512, respectively. Similar to other studies in this field, and since the amount of data for each class in the dataset was equal, the accuracy criterion was used to evaluate the proposed method. The accuracy was calculated by

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FN + FP}
\]

In Equation (8), \(TP\) is the number of samples that are correctly assigned to the goal class, \(TN\) is the number of samples that are correctly not assigned to the goal class, \(FN\) is the number of samples that are wrongly assigned to the goal class, and \(FP\) is the number of samples that are wrongly assigned to the other classes. We performed five-fold cross-validation on the dataset and reported the average recognition rate for each mini-batch size in Figure 6. The model converged at 20 epochs and performed best when the mini-batch size is set to 128.

![Figure 6. The average recognition accuracy of different batch sizes on the IAHCC-UCAS2016 varied with the training epoch.](image)

For convolution operation on \(P\), we want the size of the sequence before and after convolution operation to be same, so we need to zero-pad the sequence. We designed three padding methods, padding1, which is evenly filled at both ends of the sequence; padding2, which is filled at the head of the sequence; and padding3, padded at the end of the sequence. The change in the recognition rate during the training process is shown in Figure 7. The best recognition rates of the three methods are shown in Table 2. From Figure 7 and Table 2, we can see that padding1 is the best padding method. Similar to using CNN to recognize images, when CNN is used to recognize coordinate sequences, zero padding on the edge can effectively retain the information of the edge position. Compared with method 2 and method 3, method 1 has a higher recognition accuracy by 0.07% and 0.28%. Filling methods 2 and 3 both lose the feature of the edge position to some extent. For the fill method 3, in particular, the header position information of the character sequence is lost, which is quite important to recognize the character.
Figure 7. The recognition accuracy of different zero-pad methods on the IAHCC-UCAS2016 varies with the training epoch.

Table 2. Accuracy of different padding methods on the IAHCC-UCAS2016 dataset.

| Methods       | Accuracy (%) |
|---------------|--------------|
| padding1      | 96.07        |
| padding2      | 96.0         |
| padding3      | 95.79        |

To verify the effectiveness of the end-to-end CNN, we compared the proposed method with traditional CNN architectures on both datasets. We used the nine-layer CNN presented in [23] as a benchmark (including Cov8d using eight-direction feature maps, CovCd1 using a combination of higher-order direction feature maps and curved feature maps, and CovCd2 using a combination of eight-direction feature maps, high-order eight-direction feature maps and curved feature maps). In addition, we also compared our method with end-to-end RNN (RNN1) [20] and RNN combined with new computing architectures (RNN2) [4] on the IAHCC-UCAS2016 dataset. In Table 3–6, the column “DA” indicates whether the data-augmentation technique was adopted during training, and the column “Ensemble” indicates whether the recognition decision was made by the ensemble of multiple trained models. As shown in Table 3 and 4, the traditional CNN that recognizes directional features [23] does not directly recognize coordinate sequences, but recognizes the extracted directional feature images. This indirect learning method will affect the recognition accuracy, and it is difficult to obtain a high recognition rate when the amount of data is insufficient, so it is necessary to use data-augmentation techniques to expand the dataset during the training process. Our method does not use data augmentation techniques and achieves a recognition accuracy of 96.07% on the IAHCC-UCAS2016 dataset and 98.02% on the GB1 dataset. Although RNN can directly identify the coordinate sequence and extract the time-series features between the coordinates, it is difficult to consider the global spatial features. RNN [4,20] often use multiple models to jointly participate in recognition decisions to improve recognition accuracy. However, this strategy will exponentially increase the storage cost and the improvement in recognition accuracy is not ideal. As shown in Table 3, the recognition accuracy of RNN2-Ensemble is 0.8% higher than that of RNN2, but the storage cost is about 3.67 times that of RNN2. Compared with RNN, end-to-end CNN can extract more discriminate spatiotemporal features more comprehensively, requires only a single model and the storage cost is only 6.48 MB.
Table 3. Performance comparison of different deep-learning methods on the IAHCC-UCAS2016 dataset.

| Methods          | Accuracy (%) | Storage (MB) | DA | Ensemble |
|------------------|--------------|--------------|----|----------|
| Cov8d [23]       | 91.62        | 20.2         | No | No       |
| CovCd1 [23]      | 92.32        | 20.2         | Yes| No       |
| CovCd2 [23]      | 92.93        | 20.2         | Yes| No       |
| RNN1 [20]        | 92.50        | 7.03         | No | No       |
| RNN1-Ensemble [20] | 93.40     | 61.59        | No | Yes      |
| RNN2 [4]         | 93.60        | 7.01         | No | No       |
| RNN2-Ensemble [4] | 94.40     | 25.75        | No | Yes      |
| Proposed method  | 96.07        | 6.48         | No | No       |

Table 4. Performance comparison of different deep-learning methods on the GB1 dataset.

| Methods          | Accuracy (%) | Storage (MB) | DA | Ensemble |
|------------------|--------------|--------------|----|----------|
| Cov8d [23]       | 96.10        | 19.4         | No | No       |
| CovCd1 [23]      | 97.15        | 19.4         | Yes| No       |
| CovCd2 [23]      | 97.43        | 19.4         | Yes| No       |
| Proposed method  | 98.02        | 6.44         | No | No       |

We also compared our proposed method with traditional methods on both datasets. These methods include nearest prototype classifier (NPC) [36], nearest prototype classifier trained by MCE (NPC-MCE) [28], multistage classifiers (Multi1) [29], discriminative multistage classifiers (Multi2) [26], modified quadratic discriminant functions (MQDF) [25], locality-sensitive sparse representation-based classifiers (LSRC) [27] and locality-sensitive sparse representation toward optimized prototype classifier (LSROPC) [29]. Table 5 and 6 summarize the recognition performance of the various methods on both datasets. From Table 5 and 6, compared with traditional machine-learning methods, deep-learning technology has huge advantages, and the above methods all identify the features extracted by artificial means, which inevitably loses the timing information of the trajectory sequence to a certain extent. Therefore, using CNN to directly identify coordinate sequences can achieve an overwhelming performance improvement.

Table 5. Performance comparison of various methods on the IAHCC-UCAS2016 dataset.

| Methods          | Accuracy (%) | Storage (MB) | DA | Ensemble |
|------------------|--------------|--------------|----|----------|
| NPC [36]         | 86.90        | 8.52         | No | No       |
| NPC-MCE [28]     | 88.93        | 8.52         | No | No       |
| Multi1 [29]      | 88.28        | 128.8        | No | No       |
| Multi2 [26]      | 88.90        | 128.8        | No | No       |
| MQDF [25]        | 89.96        | 191.11       | No | No       |
| LSRC [27]        | 88.93        | 31.78        | No | No       |
| LSROPC [29]      | 90.31        | 70.00        | No | No       |
| Proposed method  | 96.07        | 6.48         | No | No       |

Table 6. Performance comparison of various methods on the GB1 dataset.

| Methods          | Accuracy (%) | Storage (MB) | DA | Ensemble |
|------------------|--------------|--------------|----|----------|
| NPC [36]         | 91.30        | 4.76         | No | No       |
| NPC-MCE [28]     | 92.96        | 4.76         | No | No       |
| Multi1 [29]      | 92.36        | 124.5        | No | No       |
| Multi2 [26]      | 93.42        | 124.5        | No | No       |
| MQDF [25]        | 95.21        | 184.84       | No | No       |
| LSRC [27]        | 94.40        | 27.86        | No | No       |
| LSROPC [29]      | 95.50        | 65.53        | No | No       |
| Proposed method  | 98.02        | 6.44         | No | No       |
5. Conclusions

This paper proposes an end-to-end classifier based on CNN for IAHCCR. Our method achieves 96.08% recognition accuracy on the IAHCC-UCAS2016 dataset with a storage cost of 6.48MB. Compared with the directional feature-extraction strategy that indirectly identifies temporal features, the direct identification of trajectory sequences can directly extract more discriminate temporal features to obtain better results, and no complex feature-extraction process is required. Unlike RNN, the macroscopic structure of trajectory sequences can also be considered. The experimental results show that the proposed method is very suitable for OLHCCR. In future work, we plan to explore more robust and efficient CNN architectures for IAHCCR.

Author Contributions: Conceptualization, X.Q.; methodology, X.Q. and M.H.; software, X.Q. and M.H.; validation, M.H.; resources, X.Q.; data curation, X.Q.; writing—original draft preparation, M.H.; writing—review and editing, X.Q., J.H. and X.W.; Funding acquisition, X.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China Youth Fund under Grant No. 61906003, and the University Synergy Innovation Program of Anhui Province under Grant No. GXXT-2021-004.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All the datasets used in this research are publicly accessible. The IAHCC-UCAS2016 dataset is available at: http://cvmt.ucas.ac.cn/dataset/, accessed on 21 June 2022. The GB1 dataset in the SCUT-COUCH2009 database is available at: http://www.hcii-lab.net/data/scutcouch/, accessed on 21 June 2022.

Acknowledgments: We sincerely thank the editors and the reviewers for their valuable comments in improving this paper. We would also like to thank L.W. Jin et. al. for helping to provide the experimental data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, C.L.; Jaeger, S., Nakagawa, M. Online recognition of chinese characters: The state-of-the-art. *IEEE Trans. Pattern Anal. Mach. Intell.* 2004, 26, 198–213.
2. Yin, F.; Wang, Q.F.; Zhang, X.Y.; Liu, C.L. ICDAR 2013 Chinese handwriting recognition competition. In Proceedings of the 2013 International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1464–1470.
3. Gan, J.; Wang, W.Q.; Lu, K. A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN. *Inform. Sci.* 2019, 378, 375–390. [CrossRef]
4. Ren, H.Q.; Wang, W.Q.; Liu, C.L. Recognizing online handwritten Chinese characters using RNNs with new computing architectures. *Pattern Recognit.* 2019, 93, 179–192. [CrossRef]
5. Li, Y.; Qian, Y.; Chen, Q.C.; Hu, B.T.; Wang, X.L.; Ding, Y.X.; Ma, L. Fast and Robust Online Handwritten Chinese Character Recognition with Deep Spatial & Contextual Information Fusion Network. *IEEE Trans. Multimed.* 2022, doi: 10.1109/TMM.2022.3143324. [CrossRef]
6. Xu, S.B.; XUE, Y.; CHEN, Y.Q. Quantitative Analyses on Effects from Constraints in Air-Writing. *IEICE Trans. Inform. Syst.* 2019, E120, 867–870. [CrossRef]
7. Fu, Z.J.; Xu, J.S.; Zhu, Z.D.; Liu, A.X. Writing in the air with WiFi signals for virtual reality devices. *IEEE Trans. Mobile Comput.* 2019, 18, 473–484. [CrossRef]
8. Gadekallu, T.R.; Srivastava, G.; Liyanage, M.; Iyapparaja, M.; Chowdhary, C.L.; Koppu, S.; Maddikunta, P.K.R. Hand gesture recognition based on a Harris Hawks optimized Convolution Neural Network. *Comput. Electr. Eng.* 2022, 100, 107836. [CrossRef]
9. Bai, Z.L.; Huo, Q. A study of nonlinear shape normalization for online hand-written Chinese character recognition: Dot density vs. line density equalization. In Proceedings of the 2006 International Conference on Pattern Recognition, Hong Kong, China, 20–24 August 2006; pp. 921–924.
10. Bai, Z.L.; Huo, Q. A study on the use of 8-directional features for online handwritten Chinese character recognition. In Proceedings of the 2005 International Conference on Document Analysis and Recognition, Seoul, Korea, 31 August–1 September 2005; pp. 262–266.
