

# Toward high-performance online HCCR: a CNN approach with *DropDistortion*, path signature and spatial stochastic max-pooling

Songxuan Lai<sup>a</sup>, Lianwen Jin<sup>a</sup>, Weixin Yang<sup>a</sup>

<sup>a</sup>*School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China*

---

## Abstract

This paper presents an investigation of several techniques that increase the accuracy of online handwritten Chinese character recognition (HCCR). We propose a new training strategy named *DropDistortion* to train a deep convolutional neural network (DCNN) with distorted samples. *DropDistortion* gradually lowers the degree of character distortion during training, which allows the DCNN to better generalize. Path signature is used to extract effective features for online characters. Further improvement is achieved by employing spatial stochastic max-pooling as a method of feature map distortion and model averaging. Experiments were carried out on three publicly available datasets, namely CASIA-OLHWDB 1.0, CASIA-OLHWDB 1.1, and the ICDAR2013 online HCCR competition dataset. The proposed techniques yield state-of-the-art recognition accuracies of 97.67%, 97.30%, and 97.99%, respectively.

**Keywords:** online handwritten Chinese character recognition, deep convolutional neural network, spatial stochastic max-pooling, character distortion, path signature,

---

## 1. Introduction

Deep convolutional neural networks (DCNNs) have brought about breakthroughs in many domains, and their application to online handwritten Chinese character recognition (HCCR) has persistently yielded state-of-the-art results in recent years [4, 5, 22, 21]. These previous work have effectively improved HCCR performance, e.g. application of the path signature theory [4], usage of comprehensive domain knowledge [22] and more advanced training methods [21], etc. However, several challenges still need to be addressed.

The main challenge presented by HCCR results from varied handwriting styles and the large number of Chinese character classes. A large-scale dataset is vital to HCCR performance, especially when using a DCNN. However, data acquisition with high quality ground-truth is a tedious job. One possible solution to this problem may be to apply character distortion to generate artificial sam-

ples [4, 22, 9, 12]. Such kind of approach enables the generation of a large number of training samples to improve the performance. However, in previous studies, character distortion is usually applied at a fixed degree throughout the entire training process. Even though a fixed high-degree distortion reduces overfitting by generating varied samples, it may lead to a distribution that deviates from the underlying data distribution, whereas a fixed low-degree distortion would achieve the opposite. Hence, more proper distortion strategies should be investigated.

Another challenge is to find a good feature representation for online characters. Although DCNN is good at capturing visual concepts from raw inputs, prior knowledge can be encoded into the inputs for a DCNN to improve the performance[22]. In online HCCR, 8-directional features [1], or path signature [8] can be regarded as prior knowledge that enhances a DCNN, but it needs further study to determine how these feature rep-

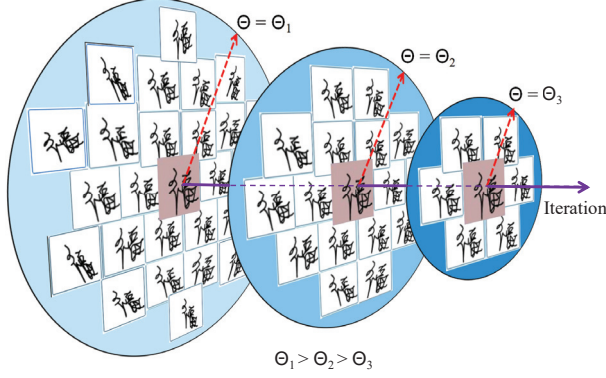


Figure 1: Illustration of *DropDistortion*. The distortion degree  $\Theta$  is lowered after certain epochs at the training stage, in order to gradually reveal the genuine data distribution.

representations could be improved. Rather than incorporating dynamic information into DCNN, [24] used recurrent neural network (RNN) to directly recognize and draw on-line Chinese characters, and showed that RNN is also able to learn complex representation of online characters.

In this paper, we focus on DCNN based models and explore several techniques to address these challenges. Motivated by the great performance of the path signature in HCCR [4], we utilize path signature as features for online characters. Different from previous usage of path signature features for HCCR, we add a time dimension to the online characters in accordance with the writing order to enable extraction of more expressive features. Our DCNN architecture is carefully designed to enable more effective learning from the signature features, with spatial stochastic max-pooling layers performing feature map distortion and model averaging. The *DropDistortion* training strategy, which gradually lowers the character distortion degree during training, is proposed to address the drawbacks of a fixed degree of distortion. Fig. 1 pictorially illustrates this concept. In early training epochs, a high degree of distortion provides improved generalization, whereas in later epochs, a decreasing degree of distortion gradually reveals more genuine data distribution. Experiments on the CASIA-OLHWDB 1.0, CASIA-OLHWDB 1.1, and the ICDAR2013 online HCCR competition dataset achieve state-of-the-art accuracies of 97.67%, 97.30% and 97.99%, respectively.

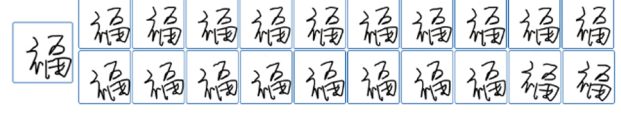


Figure 2: Rotational distortion. The leftmost one is the original Chinese character. The first and second rows correspond to lower and higher degrees of rotational distortion, respectively.

The remaining part of this paper is organized as follows. Section II provides a detailed analysis of the proposed *DropDistortion* method. Section III introduces the path signature. Section IV describes our DCNN architecture and design pipeline. Section V presents the experimental results and its detailed analysis. Finally, Section VI concludes the paper.

## 2. *DropDistortion* training strategy

Character distortion is widely used in HCCR to generate artificial training samples[4, 22]. The basic structures of the distorted samples are the same as those of the originals. Fig. 2 illustrates some rotated samples of a Chinese character. In previous studies, however, character distortion is applied upto a certain extent in an empirical way. In this paper, we study the influence of character distortion in detail and propose a simple but novel method, namely *DropDistortion*, to help enhance HCCR performance.

### 2.1. Character distortion via affine transformation

We use affine transformation to distort input characters. Let  $\Theta$  denote the degree of character distortion,  $\xi$  denote a random number drawn from uniform distribution  $U(-\Theta, \Theta)$ , and  $[x \ y]$  denote the coordinate series of a character stroke, a matrix of size  $n \times 2$  where  $n$  indicates the number of data points. Then we can apply affine transformation to distort online handwritten Chinese characters:

$$[x \ y] \Leftarrow [x \ y] \cdot \begin{bmatrix} 1 + \xi_x & 0 \\ 0 & 1 + \xi_y \end{bmatrix}, \quad (1)$$

$$[x \ y] \Leftarrow [x \ y] \cdot \begin{bmatrix} 1 & \xi \\ 0 & 1 \end{bmatrix}, \quad (2)$$

$$[x \ y] \Leftarrow [x \ y] \cdot \begin{bmatrix} 1 & 0 \\ \xi & 1 \end{bmatrix}, \quad (3)$$

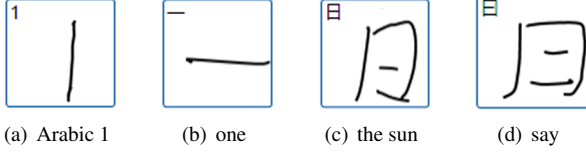


Figure 3: Easily confused Characters when distorted.

$$[\mathbf{x} \ \mathbf{y}] \leftarrow [\mathbf{x} \ \mathbf{y}] \cdot \begin{bmatrix} \cos(\xi) & -\sin(\xi) \\ \sin(\xi) & \cos(\xi) \end{bmatrix}, \quad (4)$$

where (1) stretches or shrinks the stroke, (2) and (3) slant the stroke and (4) performs rotational distortion. A character is distorted by simply applying one or more of the above equations to all its strokes, with  $\xi$  in the same equation fixed within a character. We also randomly translate the characters to achieve a better distortion diversity.

## 2.2. Analysis of character distortion from the Bayesian perspective

First, consider a simple example with two handwritten characters, the Arabic numeral “1” and the Chinese character “one” (Fig. 3). We use  $A$  and  $B$  to denote them respectively for convenience. When written by hand  $A$  and  $B$  look quite similar except for their orientations. In other words, they have the same topological structure. Let  $\theta$  denote the angle between their orientations and the horizontal direction, and assume  $\theta$  obeys a Gaussian distribution:  $\theta_A \sim N(\pi/2, \sigma^2)$ ,  $\theta_B \sim N(0, \sigma^2)$ .

Then from the Bayesian perspective, we have

$$f(A|\theta) = f(\theta|A)f(A)/f(\theta), \quad (5)$$

$$f(B|\theta) = f(\theta|B)f(B)/f(\theta), \quad (6)$$

where  $f(A|\theta)$  and  $f(B|\theta)$  are probabilities predicted by a classifier,  $f(\theta|A)$  and  $f(\theta|B)$  model the orientation information,  $f(A)$  and  $f(B)$  model the topological structure information, and  $f(\theta)$  is a normalized constant determined by the training set and can be safely removed:

$$f(A|\theta) \propto f(\theta|A)f(A), \quad (7)$$

$$f(B|\theta) \propto f(\theta|B)f(B). \quad (8)$$

If the topological structure of  $A$  and  $B$  is nearly the same, we have

$$f(A) \approx f(B). \quad (9)$$

Although  $f(A) \approx f(B)$ , the classifier is still able to correctly classify these two characters according to  $\theta$  due to the difference between  $f(\theta|A)$  and  $f(\theta|B)$ .

If rotational distortion is applied to  $A$  and  $B$  with  $\xi$  drawn from the uniform distribution  $U(-\Theta, \Theta)$ , then the distribution of  $\theta$  is shown in Fig. 4. If rotational distortion is applied to an extreme, i.e.,  $\Theta = 2\pi$ , then  $\theta$  obeys a uniform distribution as well. Under these circumstances, the classifier is unable to distinguish between  $A$  and  $B$  because  $f(\theta|A) = f(\theta|B)$  and  $f(A) \approx f(B)$ .

However, when it comes to HCCR, the situation is somewhat different. A rotated Chinese character will rarely resemble other Chinese characters. A high degree of rotational distortion actually removes the conditional term  $f(\theta|C)$  and forces the classifier to predict  $f(C)$  more accurately, i.e., to achieve improved learning of the topological structure of the character  $C$ . By preventing co-adaptation of  $f(C)$  and  $f(\theta|C)$ , a high degree of rotational distortion reduces overfitting. Other distortions such as stretching can be analyzed in the same way. Hence using character distortion to generate artificial samples is a reasonable and effective way to enhance HCCR performance.

## 2.3. DropDistortion: stepwise character distortion

Although effective in improving HCCR performance, character distortion changes the data distribution. Specially, it confuses some similar characters inevitably. For example, the Chinese characters “the sun” (Fig. 3 (c)) and “say” (Fig. 3 (d)) would be indistinguishable if they were to be highly stretched. The change of distribution should be considered to give further improvements, but no such efforts have been reported in previous studies [4, 22, 9, 12], where the character distortion is carried out upto a fixed degree during the entire training process. The proposed *DropDistortion* method is a novel strategy that is designed to take the change of distribution into consideration. It’s based on a simple idea that the DCNN should be fine-tuned with more genuine samples, i.e., low-degree or non distorted samples. The proposed *DropDistortion* algorithm is given in algorithm 1.

In the proposed method, a high degree of distortion is used in the early training epochs to generate varied samples to help the DCNN learn effective features and reduce the risk of overfitting. In the later epochs, the degree of distortion decreases gradually to allow a subsequent finer

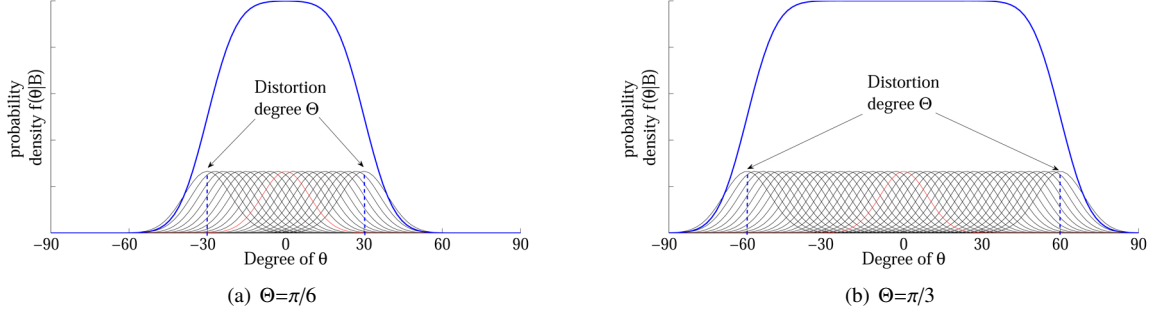


Figure 4: Conditional distribution of  $\theta$  given  $B$ , a mixture of Gaussians.  $f(\theta|A)$  is exactly the same case.

---

**Algorithm 1** *DropDistortion*


---

**Input:** training set  $X=(x_i, y_i)$ ,  $i=1, \dots, m$  of  $k$  classes.

**Initialize:** index  $n \leftarrow 1$ ; distortion degree  $\Theta_1 > \Theta_2 > \dots > \Theta_N$ .

**Output:** DCNN parameters  $W$ .

**Begin:**

```

while  $n < N$  do
   $\Theta \leftarrow \Theta_n$ 
  while not converge do
    Sample distortion at degree  $\Theta_n$ 
    Update  $W$  through back propagation algorithm
  end while
   $n \leftarrow n + 1$ 
end while

```

---

adjustment of the DCNN with more genuine samples. The only extra complexity *DropDistortion* introduces is to monitor the training loss and decrease the distortion rate if a certain condition is fulfilled. In practice, *DropDistortion* can be simply implemented in a multi-step way, and in this paper it is implemented in a three-step way as explained in Fig. 1.

### 3. Representation of online handwritten characters with path signature

In mathematics, a path signature is a collection of iterated integrals [8, 3] of a path. Let  $X:[0, T] \rightarrow \mathbb{R}^d$  denote a continuous path of bounded variation, mapping from time interval  $[0, T]$  to space  $\mathbb{R}^d$ . Then the  $k$ -th iterated integral

of path  $X$  is

$$I^k = \int_{0 < t_1 < \dots < t_k < T} 1 dX_{t_1} \otimes \dots \otimes dX_{t_k}, \quad (10)$$

where  $\otimes$  represents the tensor product. By convention,  $I^0$  is the number one. The signature of path  $X$  is the collection of all the iterated integrals of  $X$ , denoted by  $S(X)$ . Since this is an infinite series, in practice one often considers the first  $m$ th-order integrals, namely truncated path signature:

$$S(X)|_m = \{1, I^1, I^2, \dots, I^m\}. \quad (11)$$

As is often the case,  $X$  is sampled and approximated by a set of discrete points. Then the iterated integrals can be approximated by using some simple tensor algebra [4].

Online handwriting can be seen as a path mapping from time interval  $[0, T]$  to  $\mathbb{R}^2$ . Graham [4] first introduced truncated path signature features to online handwritten character recognition as inputs for a DCNN and achieved remarkable performance. Some previous work [4, 20, 19] experimented with truncated level  $m$  and found that the accuracy increases with the increase of  $m$  and gradually saturates. In our work,  $m$  is empirically set as 4 because integrals of a higher order typically characterize more trivial details of a path and do not lead to further improvement.

The signature is invariant to translations and time re-parameterization of the path, and uniquely characterizes the path if it contains no part that exactly retraces itself [6, 2]. However, in handwritten characters, some local parts do retrace themselves due to joined-up writing. For example, path  $([0,0], [1.5,2.5], [3,3], [1.5,2.5])$

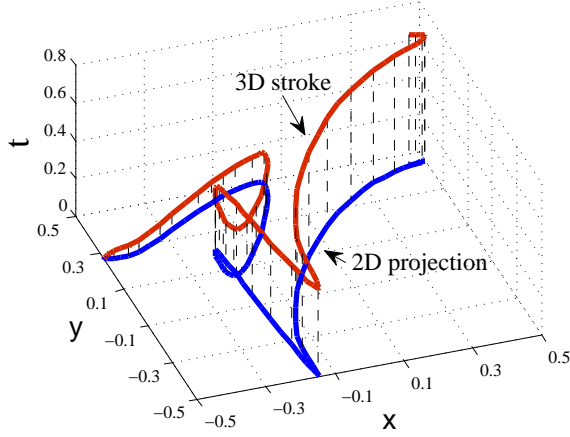


Figure 5: Time dimension is added as a third dimension. The 2D projection is the original stroke.

has the same signature with path  $([0,0], [1.5,2.5])$  because  $([1.5,2.5], [3,3], [1.5,2.5])$  exactly retraces itself. To handle this problem, we introduce a monotone time dimension to the original handwriting sequences, i.e., the  $n \times 2$  matrix  $[x \ y]$  is appended by a column vector  $t=[0 \ 1 \ 2 \ \dots \ n-1]^T$ . This concept is demonstrated in Fig. 5. The time dimension ensures the uniqueness for the path signature, hence features extracted from the  $n \times 3$  matrix  $[t \ x \ y]$  would be more expressive.

In practice, if  $m=4$ , then  $N=31$  for a 2D character and  $N=121$  for a 3D character, where  $N$  denotes the number of input channels.

#### 4. Deep convolutional neural network with spatial stochastic max-pooling

Deep convolutional neural networks (DCNNs) have shown great success in computer vision and pattern recognition, and different architectures of DCNN have been explored [10, 13, 17, 7]. However, these designs did not evaluate the max-pooling (MP) operation, which incorporates a degree of invariance with respect to translations and elastic distortions into the DCNN [11]. Max-pooling operates in a sliding window method, and conventionally the pooling region slides with a fixed integer stride. Usually the stride is two, hence the size of feature map is reduced by a factor of two.

##### 4.1. Spatial stochastic max-pooling

Compared to traditional MP layers, fractional max-pooling (FMP) [5] layers reduce the feature map size by a factor of  $\alpha$  with  $\alpha$  as a fraction. If  $1 < \alpha < 2$ , then FMP reduces the size in a slower manner than MP. As  $\alpha$  is a fraction, the pooling stride cannot be a fixed integer. In this case, the pooling region slides with a pre-calculated stride series. We prefer to call FMP as spatial stochastic max-pooling (SSMP), which describes the pooling process more precisely.

Let  $N_{in}$  and  $N_{out}$  denote the input and output feature map size respectively. Then

$$N_{out} = \lfloor N_{in}/\alpha + 0.5 \rfloor, \quad (12)$$

$$\alpha \leftarrow N_{in}/N_{out}. \quad (13)$$

$\alpha$  is renewed above due to the rounding effect. The stride series is a  $1 \times N_{out}$  vector for each dimension of the output image. Then for the  $i$ -th ( $i=0,1,\dots,N_{out}-2$ ) position of the stride series,

$$a_i = \lfloor i \times \alpha + th_i \rfloor, \quad (14)$$

$$s_i = a_i - a_{i-1}. \quad (15)$$

where  $s_i$  is the stride series,  $a_i$  is the accumulated stride series with  $a_{-1} = 0$  and  $a_{N_{out}-1} = N_{in} - 2$  (the pooling size is  $2 \times 2$ ), and  $th_i \in [0, 1)$  is a randomly drawn threshold to round  $a_i$  up or down. In practice,  $th_i$  can be set independently at different position  $i$ , or set only once and shared across different positions as a constant. We denote these two strategies as SSMP<sub>1</sub> and SSMP<sub>2</sub> respectively. Different to [5] which employs SSMP<sub>1</sub>, we propose a new strategy that  $th_i$  is set independently at each position  $i$  in early training epochs and set only once in the last epochs. We call this strategy as SSMP<sub>3</sub>.

During the pooling process, SSMP introduces a certain degree of randomness into the pooling regions by a random choice of the feasible stride series. Fig. 6 illustrates this concept vividly. The colored  $2 \times 2$  squares are the pooling regions chosen. By setting  $\alpha$  as 1.5, the feature map size is reduced from  $6 \times 6$  to  $4 \times 4$  with multiple feasible pooling choices; hence every forward path may have a slightly different output. Moreover, SSMP achieves elastic distortion [5] of the feature maps of the previous layer, which implicitly distorts the input characters and improves the generalization ability of a DCNN. In test phase, running the network for multiple times and

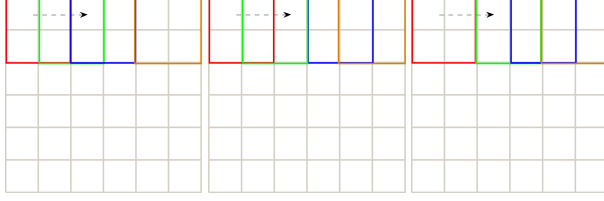


Figure 6: Random pooling regions of SSMP(1.5). Pooling stride series from left to right: 112, 121, 211.

averaging the outputs achieve the same effect as an ensemble of similar networks.

#### 4.2. Other details

In our DCNN, we use  $3 \times 3$  or  $2 \times 2$  convolutional filters [4], which also makes the network deeper compared to larger filter size. The path signature features are extracted from online handwriting data sequences, and then padded into  $N \times 50 \times 50$  bitmaps embedded in larger grids (pad zeros around the characters). The first pooling layer is max-pooling rather than SSMP, as we find max-pooling works better than SSMP when dealing with the input, partly due to the input noise introduced by padding a continuous handwriting path onto a discrete bitmap. We apply the leaky ReLU activation function [15, 18] with  $a=0.333$ , which outperforms ReLU [16] in terms of convergence speed and representation capability [18].

## 5. Experiments

### 5.1. Dataset

The datasets we used are CASIA-OLHWDB 1.0 (DB 1.0), CASIA-OLHWDB 1.1 (DB 1.1) [14] and ICDAR2013 competition dataset [23]. DB 1.0 contains 3740 Chinese character classes in standard level-1 set of GB2312-80 (GB1) and is obtained from 420 writers (336 $\times$ 3740 samples for training, 84 $\times$ 3740 for testing). DB 1.1 contains 3755 classes in GB1 and is obtained from 300 writers (240 $\times$ 3755 samples for training, 60 $\times$ 3755 for testing). The database for the ICDAR2013 online HCCR competition consists of three datasets containing isolated Chinese characters, namely, CASIA-OLHWDB 1.0~1.2 (DB 1.0~1.2). The test set was released after the competition, which contains 3755 classes in GB1.

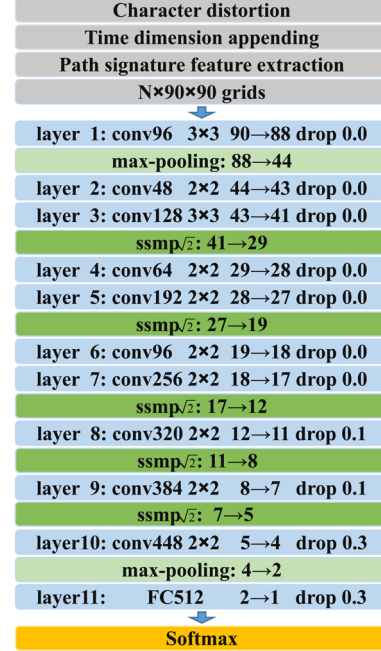


Figure 7: Our proposed DCNN architecture. The “ $\rightarrow$ ” denotes the reduction of feature map size. The value following “drop” means the dropout ratio.

### 5.2. Network training

In the experiments, we use nesterov momentum with momentum  $\mu=0.9$ . The learning rate is set as 0.003 initially, and halved after the first iteration, and then exponentially decreases to 0.00001. The training mini-batch size is 96. We trained the DCNN for 70 epochs.

### 5.3. Experimental results

#### 5.3.1. Individual evaluation of the proposed methods

We first conducted experiment on DB1.1 to evaluate the effectiveness of our proposed method. i.e, *DropDistortion*, SSMP<sub>3</sub> and 3D signature. We designed a baseline DCNN with simple 0-1 bitmaps as input, and fixed the distortion rate  $\Theta = 0.3$ . We evaluated the three techniques individually through the variable-controlling approach. The architecture of the baseline DCNN is Input - 32C3 - MP2 - 64C3 - 96C3 - MP2 - 128C3 - 160C3 - MP2 - 192C3 - 224C3 - MP2 - 256C3 - Output.

In order to evaluate the *DropDistortion*, the baseline DCNN is trained with *DropDistortion*.



In order to evaluate the 3D signature features, the baseline DCNN is trained with 2D signature and 3D signature.

For the evaluation of the SSMP and the  $th_i$  drawing methods as described in section 4.1, i.e. SSMP<sub>1</sub>, SSMP<sub>2</sub> and SSMP<sub>3</sub>, two MP layers in the baseline DCNN is replaced with four SSMP layers with  $\alpha=1.5$ . Refer to Table 1. The last convolutional layer is 256C2 instead of 256C3 because  $1.5^4 > 2^2$ . In the test phase, repeatedly running the network produce a slightly different output each time because of SSMP. Averaging these outputs can improve the accuracy.

The experimental results are presented in Table 2. The employed three techniques, namely *DropDistortion*, 3D signature and SSMP<sub>3</sub>, all improve the performance over the baseline and their corresponding counterparts. The 2D signature substantially improves the performance, while the 3D signature gives further improvement. SSMP greatly decreases the error rates by averaging 10 test scores, and the SSMP<sub>3</sub> we proposed is better than SSMP<sub>1</sub> and SSMP<sub>2</sub>. *DropDistortion* decreases the error rate with little extra cost. As *DropDistortion* and SSMP can be used jointly with path signature, we conducted further experiments to evaluate the joint effects.

### 5.3.2. Joint evaluation of the proposed methods

To achieve state-of-art results, we extended the baseline model to be deeper and wider. The overall architecture is shown in Fig. 7. The second (48C2), fourth (64C2) and sixth layer (96C2) reduce the dimension in lower layers and act as regularization. The experimental results are presented in Table 3, where we can see *DropDistortion* always achieves the best performance, and a higher degree distortion is usually better than a lower degree distortion. *DropDistortion* brings about an obvious improvement when using bitmaps as inputs, as bitmaps only contains structural information and *DropDistortion* helps to learn the structure.

Compared to rendering online characters as bitmaps, the path signature greatly improves the recognition accuracy, which proves its effectiveness in information extraction. And the 3D signature also gives consistent improvement over the 2D signature.

The SSMP averaging results are given in Table 4 (*DropDistortion* + 3D signature + SSMP<sub>3</sub>). By averaging 10 test scores of the network, the test error rates decrease

significantly on both DB1.1 and ICDAR2013 competition dataset.

The best results for DB1.1 and ICDAR2013 competition dataset are both produced by jointly using *DropDistortion*, 3D signature and SSMP, which results in relative error reduction of 36.3%  $(=(4.24-2.70)/4.24*100\%)$  and 37.2%  $(=(3.20-2.01)/3.20*100\%)$  respectively.

### 5.3.3. Comparison with published state-of-the-art results

We conducted further experiments on DB 1.0, and in Table 5 we compared our method with some state-of-the-art achieved for HCCR in previous studies. It can be seen from Table 5 that our method has achieved the highest recognition accuracies for all the three datasets, showing the effectiveness of the proposed approach.

Our DCNN architecture in Fig. 7 follows that of DeepCNet and FMP network, e.g.  $2 \times 2$  convolutional kernels, increasing kernel numbers and stacked SSMP layers, but it is no deeper than the network in [5]. Domain knowledge and *DropSample* can be adapted into our approach to give further improvement, but this is beyond the scope of this paper. 8-DirectMap contains directional information, which is equivalent to the first level of path signature[21]. Higher levels of path signature we used here contain richer information and can further improve the performance.

Fig. 8 demonstrates some randomly chosen misclassified samples, from which we can discern that the misclassified samples are usually illegible. Some are even mislabeled or wrongly written.

## 6. Conclusion

This paper presents several new techniques for online HCCR that effectively boost the recognition accuracy. We proposed a simple but effective character distortion method called *DropDistortion*, which improves the recognition accuracy with little additional computational cost. Path signature acts as an effective feature representation for online characters, and the SSMP layers in our DCNN perform feature map distortion and model averaging. Experiments on CASIA-OLHWDB 1.0, CASIA-OLHWDB 1.1 and the ICDAR2013 online HCCR competition dataset achieved new state-of-the-art accuracies of 97.67%, 97.30% and 97.99%, respectively. Compared

Table 1: Configuration of SSMP DCNN by replacing MP layers with SSMP layers.

dimension	32	–	64	96	–	128	–	160	–	192	–	224	–	256
Baseline DCNN	C3	MP2	C3	C3	<b>MP2</b>	C3	()	C3	<b>MP2</b>	C3	()	C3	<b>MP2</b>	C3
SSMP DCNN	C3	MP2	C3	C3	<b>SSMP1.5</b>	C3	<b>SSMP1.5</b>	C3	<b>SSMP1.5</b>	C3	<b>SSMP1.5</b>	C3	<b>MP2</b>	C2

Table 2: Evaluation of the proposed three techniques: *DropDistortion*, SSMP and path signature (test error rates, %)

Models	1 test	10 tests
Baseline( $\Theta = 0.3$ )	4.70	—
<i>DropDistortion</i>	4.55	—
2D signature	3.18	—
3D signature	3.08	—
SSMP <sub>1</sub>	4.78	4.30
SSMP <sub>2</sub>	4.75	4.25
SSMP <sub>3</sub>	4.68	4.23

with previous best result [25] on the ICDAR2013 competition dataset, our method has achieved an error reduction of 17.9%, showing the effectiveness of the proposed approach.

Although we mainly focus on online HCCR, the proposed *DropDistortion* method is expected to serve as a general technique in many other tasks in machine learning, such as image classification. The *DropDistortion* training strategy was applied in a simple three-step way in this paper, and in future work it is of great worth to investigate self-adaptive *DropDistortion*, where the distortion degree could be automatically adjusted.

## References

- [1] Bai, Z.L., Huo, Q., 2005. A study on the use of 8-directional features for online handwritten Chinese character recognition, in: Proc. Eighth International Conference on Document Analysis and Recognition (ICDAR), pp. 262–266.
- [2] Boedihardjo, H., Geng, X., Lyons, T., Yang, D., 2014. The signature of a rough path: Uniqueness. CoRR abs/1406.7871.
- [3] Chen, K.T., 1957. Integration of paths, geometric in-



Figure 8: Examples of some misclassified samples. The printed Chinese characters under the samples are the ground truth class (left) and the predicted class (right).

variants and a generalized Baker-Hausdorff formula. Annals of Mathematics , 163–178.

- [4] Graham, B., 2013. Sparse arrays of signatures for online character recognition. CoRR abs/1308.0371.
- [5] Graham, B., 2014. Fractional max-pooling. CoRR abs/1412.6071.
- [6] Hambly, B., Lyons, T., 2010. Uniqueness for the signature of a path of bounded variation and the reduced path group. Annals of Mathematics , 109–167.
- [7] He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. CoRR abs/1512.03385.
- [8] Ilya, C., Andrey, K., 2016. A primer on the signature method in machine learning. CoRR abs/1603.03788.



Table 3: Test error rates(%) on CASIA-OLHWDB 1.1 and ICDAR2013 online HCCR competition dataset

Dataset	Features	Character Distortion $\Theta$		
		fixed 0.1	fixed 0.3	DropDistortion: 0.3→0.2→0.1
DB 1.1	Bitmap	4.24	4.18	4.08
	2D signature	3.10	3.02	2.99
	3D signature	3.01	2.95	2.92
ICDAR2013	Bitmap	3.20	3.25	3.13
	2D signature	2.32	2.27	2.24
	3D signature	2.30	2.26	2.21

Table 4: Test error rates(%) of SSMP model averaging

Database	1 test	2 tests	3 tests	4 tests	5 tests	6 tests	7 tests	8 tests	9 tests	10 tests
DB 1.1	2.92	2.82	2.78	2.75	2.75	2.73	2.71	2.70	2.70	2.70
ICDAR2013	2.21	2.11	2.09	2.08	2.05	2.05	2.03	2.03	2.03	2.01

- [9] Jin, L., Huang, J., Yin, J., Motorol, 2002. A novel deformation transformation and its application to handwritten Chinese character shape correction. *Journal of Image and Graphics* 7, 170–175.
- [10] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks, in: *Advances in neural information processing systems*, pp. 1097–1105.
- [11] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 2278–2324.
- [12] Leung, K., Leung, C.H., 2009. Recognition of handwritten Chinese characters by combining regularization, Fisher’s discriminant and distorted sample generation, in: *Proc. 10th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1026–1030.
- [13] Lin, M., Chen, Q., Yan, S., 2013. Network in network. *CoRR* abs/1312.4400.
- [14] Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F., 2011. CASIA online and offline Chinese handwriting databases, in: *Proc. 11th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 37–41.
- [15] Maas, A.L., Hannun, A.Y., Ng, A.Y., 2013. Rectifier nonlinearities improve neural network acoustic models, in: *Proc. ICML*, p. 1.
- [16] Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines, in: *Proc. 27th International Conference on Machine Learning (ICML)*, pp. 807–814.
- [17] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.
- [18] Xu, B., Wang, N., Chen, T., Li, M., 2015. Empirical evaluation of rectified activations in convolutional network. *CoRR* abs/1505.00853.
- [19] Yang, W., Jin, L., Liu, M., 2015a. Chinese character-level writer identification using path signature feature, DropStroke and deep CNN, in: *Proc. 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 546–550.

Table 5: Comparison with published state-of-the-art accuracies(%)

Methods	DB 1.0	DB 1.1	ICDAR2013 Competition
DeepCNet [4]	—	96.42	97.39
Domain Knowledge [22]	97.20	96.87	97.20
FMP ensemble [5]	—	97.03	—
DropSample [21]	97.33	97.06	97.51
DirectMap+convnet [25]	—	—	97.55
Our Method 1 test	97.5	97.08	97.79
Our Method 10 tests	<b>97.67</b>	<b>97.30</b>	<b>97.99</b>

[25] Zhang, X.Y., Bengio, Y., Liu, C.L., 2017. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition* 61, 348–360.

- [20] Yang, W., Jin, L., Liu, M., 2016a. DeepWriterID: An end-to-end online text-independent writer identification system. *IEEE Intelligent Systems* 31, 45–53.
- [21] Yang, W., Jin, L., Tao, D., Xie, Z., Feng, Z., 2016b. DropSample: A new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten chinese character recognition. *Pattern Recognition* 58, 190–203.
- [22] Yang, W., Jin, L., Xie, Z., Feng, Z., 2015b. Improved deep convolutional neural network for on-line handwritten Chinese character recognition using domain-specific knowledge, in: *Proc. 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 551–555.
- [23] Yin, F., Wang, Q.F., Zhang, X.Y., Liu, C.L., 2013. ICDAR 2013 Chinese handwriting recognition competition, in: *Proc. 12th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1464–1470.
- [24] Zhang, X., Yin, F., Zhang, Y., Liu, C., Bengio, Y., 2016. Drawing and recognizing chinese characters with recurrent neural network. *CoRR* abs/1606.06539.