Automatic Handwriting Verification and Suspect Identification for Chinese Characters Using Space and Frequency Domain Features

Wei-Cheng Liao, Jian-Jiun Ding
Graduate Institute of Communication Engineering, National Taiwan University
E-mail: b04901151@ntu.edu.tw [1]; jjding@ntu.edu.tw [2], TEL: 886-2-33669652

***Abstract*—Automatic handwriting verification is to identify whether the script was written by a person himself or forged. Compared to related works about handwriting verification, the proposed algorithm adopts the features in both the time domain and the frequency domain. Moreover, in addition to distinguishing the forged manuscript from the genuine one, the proposed algorithm can also identify the suspect. The proposed algorithm is robust to writing instruments. In addition to the information of the luminance of the script, we also adopt the energy distribution on the 2-D frequency domain, the Pearson product-moment correlation coefficient (PPMCC) with genuine scripts, and vital information on characterized script points. Simulations show that the proposed method outperforms many advanced methods, including the deep-learning based method and manual identification by human beings. The proposed algorithm can well identify the script even if it is forged after several times of practice.**

Keyword —handwriting, feature extraction, forensic signal processing, Chinese script, suspect investigation

I. Introduction

In this paper, an advanced algorithm for handwriting verification is proposed. Handwriting verification is very helpful for criminal investigation, testament verification and finance, and security. Handwriting verification is different from character recognition: The former is to judge whether the script was written by a person himself or forged while the latter is to recognize what the word is. Therefore, it is critical for handwriting verification to extract the features that are helpful for distinguish genuine and forged script.

The framework of a typical handwriting verification algorithm is as in Fig. 1. In preprocessing, we normalize the luminance. In feature extraction, we adopt the features of the area of the script, luminance variation, the energy distribution on the 2-D frequency domain, the PPMCC with genuine scripts, and some vital information on characterized script points. Since the scripts are usually written in fixed blankets,

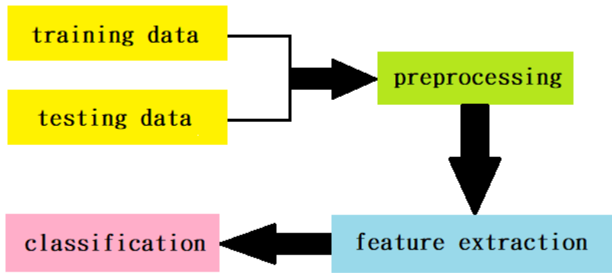

Fig 1: Framework of a handwriting verification system.

Fig 2: Scripts before and after separation.

we have to normalize images in advance before extracting features.

In classification, we first classify scripts into genuine or forged ones in the 1st stage. Then, for the forged scripts, we determine which suspect wrote the script in the 2nd stage. We adopted the support vector machine (SVM) [4, 5] in the 1st stage and the modified SVM for multi-classification in the 2nd stage.

In simulations, we show the results of the proposed method and compare it with state-of-the-art methods and manual identification. Simulations show that the proposed algorithm outperforms other methods, including the deep-learning based method and manual identification.

II. Preprocessing

Before script identification, we first separate the scripts into the characters according to the distances between the side and the regulations of character sizes and distances, as in Fig. 2.

Then, to avoid the results being affected by the written instrument, we perform normalization for the luminance to make the mean of the script equivalent.

The scripts were written by different writing instruments (blue, red, and black pens). Therefore, we choose the luminance of blue pens as the standard and normalize the luminance. For instance, if a red script with luminance 173 has the same rank as a blue script with luminance 129, we normalize the luminance of the red script from 173 to 129. Therefore, we can guarantee that all the scripts are compared under the same standard and the misjudgment caused from different writing instrument is avoided.

III. Feature Extraction

Everyone has their own habit for writing Chinese scripts even when they imitate the scripts of others deliberately. Therefore, we can extract the features that are helpful for identifying whether the scripts are genuine or forged.

A. Basic Features

First, one can use the area of the script, the standard deviation of luminance, and the standard deviations of x and y coordinates. The standard deviation of the luminance is determined from:

$$\sigma_Y = \sqrt{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[Y(x,y)-\mu_Y\right]I(x,y)}{\sum_{x=1}^{M}\sum_{y=1}^{N}I(x,y)}} \tag{1}$$

where Y(x, y) is the luminance, μ_Y is the mean of luminance, $I(x, y) = 1$ if the pixel (x, y) is in the script part and $I(x, y) = 0$ if (x, y) is in the background. The standard deviations of x and y coordinates are:

$$\sigma_x = \sqrt{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(x-x_0\right)^2 I(x,y)}{\sum_{x=1}^{M}\sum_{y=1}^{N}I(x,y)}}, \tag{2}$$

$$\sigma_y = \sqrt{\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(y-y_0\right)^2 I(x,y)}{\sum_{x=1}^{M}\sum_{y=1}^{N}I(x,y)}} \tag{3}$$

where x_0 and y_0 are the means of x and y coordinates of the script part, respectively.

B. Energy Distribution in the 2-D Frequency Domain

When forging a script, it is impossible to take the features in the frequency domain into account. Therefore, the frequency domain feature, such as the energy distribution on the 2-D frequency domain, is a critical feature in handwriting verification.

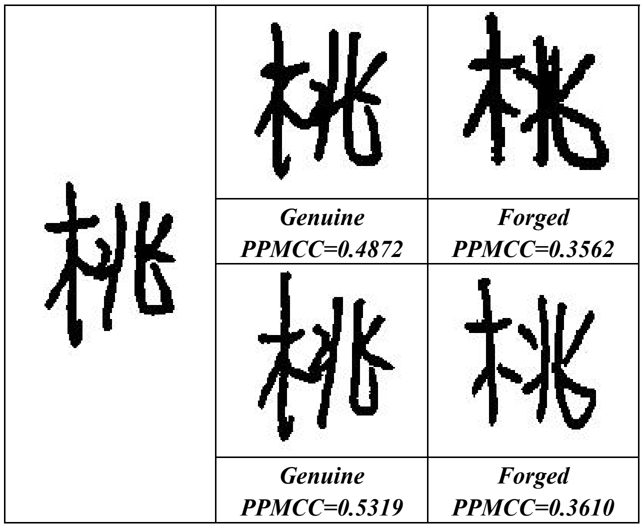

Fig 3: PPMCC of two genuine and two forged script.

First, we normalize the sizes of all the words. After normalization, we perform the 2-D discrete Fourier transform on each word as follows:

$$\hat{x}[l][k] = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{MN}mnlk} x[m+1,n+1], \tag{4}$$

where $x[m, n]$ is the binarized image of the word. Then, the energy on the 2-D frequency domain can be determined by:

$$E[l][k] = \hat{x}[l][k]\hat{x}^*[l][k]. \tag{5}$$

Then, we separate the energy into 6×6 regions and sum up the energy for each region. Due to the symmetry property of

$$E[l][k] = E[M-l][N-k], \tag{6}$$

we only have to extract 36/2 = 18 features.

C. PPMCC with Genuine Scripts

Scripts written by the same person may have similar angle, position and length for each stroke. Therefore, we can use the Pearson product-moment correlation coefficient (PPMCC) to determine whether a script is genuine or forged.

We adopt the normalized scripts with one time of dilation enhance the performance. The correlation coefficient between two images can be calculated as follows:

$$\sigma_Y = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_1(x,y)-p_1\right]\times\left[I_2(x,y)-p_2\right]}{\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_1(x,y)-p_1\right]^2}\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I_2(x,y)-p_2\right]^2}} \tag{7}$$

where

$$p_1 = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}I_1(x,y)}{MN}, \quad p_2 = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}I_2(x,y)}{MN}. \tag{8}$$

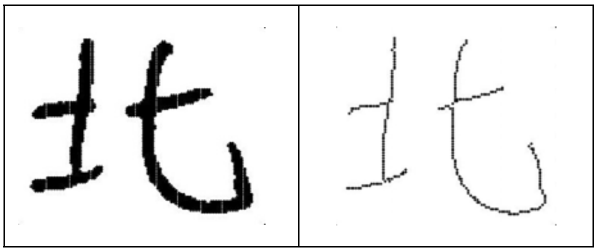

Fig 4: Thinning of a word

The value r is the PPMCC between two images. Fig. 3 shows the PPMCC of an example with two genuine scripts and two forged scripts, respectively.

D. Characterized Points

Characterized points, which are usually the end of strokes and the points with large curvature, are usually important features for manual identification for handwriting verification. In this paper, we try to extract characterized points automatically and adopt them as features for an automatic handwriting verification algorithm.

The process of finding and matching the characterized point can be divided into the following 4 steps:

i. Perform thinning [6] on each script.
ii. Find the characterized points of scripts.
iii. Match the characterized points [7] with those of the demonstration words.
iv. Collect the features of characterized points.

i. Thinning each script.

To well determine the characterized points and remove the effect of stroke width, we have to thin each script, as in Fig. 4.

ii. Choice of characterized points

After thinning, we need to find the start points, the end points and corners of the strokes. To accomplish it, we calculated the points with local minimal angles (θ). The angle for each point is determined as follows. First, we sort the points along the contour in the clockwise direction. If (x_n, y_n) is the n^{th} point on the contour, then the angle of (x_n, y_n) is

$$\theta = \arccos \frac{(x_{n+L}-x_n)(x_{n-L}-x_n)+(y_{n+L}-y_n)(y_{n-L}-y_n)}{\sqrt{(x_{n+L}-x_n)^2+(y_{n+L}-y_n)^2}\sqrt{(x_{n-L}-x_n)^2+(y_{n-L}-y_n)^2}} . \quad (9)$$

If a point has a local minimal angle, when $0° \leqq \theta \leqq 45°$, it is a start or an end point. When $45° < \theta \leqq 150°$, it is a corner. When $\theta > 150°$, it is just a regular point. Fig. 5 shows θ of some characterized points.

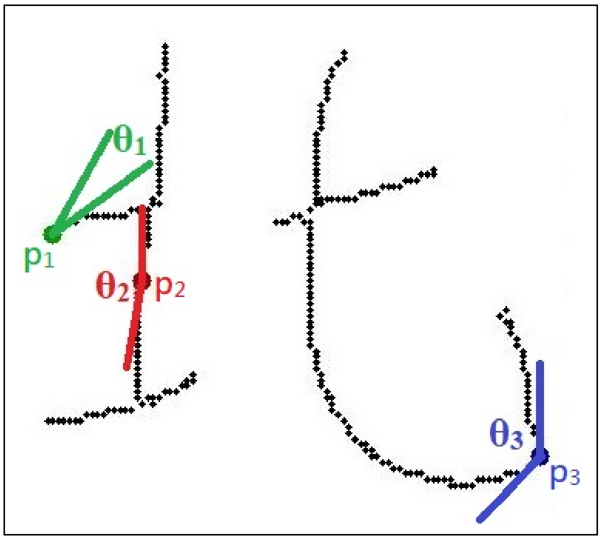

Fig 5: Angles of some points. p_1 and p_3 are characterized points but p_2 is not.

In Fig 5, θ_1=30°, θ_2=162° and θ_3=135°. Therefore, p_1 is a regular point and p_2 is a start or end point while p_3 is a corner.

iii. Matching

We choose the words with the DFKai-SB style as demonstration words and treat them as standard words for matching characterized points.

To match characterized points, we select three vital features, including the x axis ratio, the y axis ratio, and the direction of the normal vector of the characterized points. The x-axis ratio of a point (x_n, y_n) is defined from

$$x \text{ ratio} = \frac{\text{number of script pixels with } x \le x_n}{\text{total number of script pixels}} \quad (10)$$

and the y-axis ratio can be defined in a similar way. Then, we find the characterized points with the minimum cost of the following loss function:

$$L = w_1(\Delta x) + w_2(\Delta y) + w_3(\Delta\theta), \quad (11)$$

where Δx, Δy, and $\Delta\theta$ are the differences of the x-axis ratios, the y-axis ratios [9], and the direction of the normal vectors between the input and the demonstration words, respectively. w_1 and w_2 are related the importance of the x-axis and y-axis ratios ($0.1 \leqq w_1, w_2 \leqq 0.6$) while w_3 is related to the importance of the direction of the characterized point (for a corner, w_3 = 0.002 but for a start / end point, w_3 = 0.03).

However, if a characterized point is in the central part of the word, its x-axis and y-axis ratios may change dramatically. Therefore, to avoid the problem, we only chose the characterized points that have the maximal (or minimal) x (or y) coordinate in a row or column, as in Fig. 6.

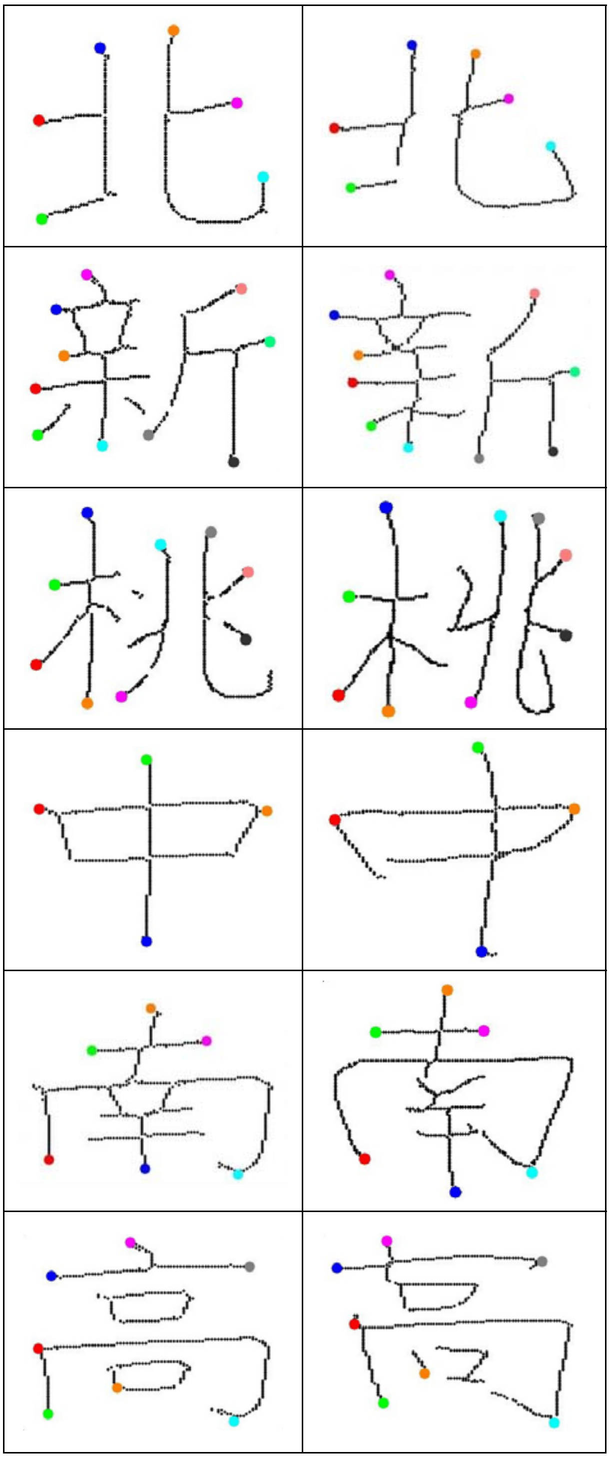

Fig 6: Characterized points of words. Only the points whose x (or y) coordinate is the maximum (or minimum) in a row (or column) are chosen.

iv. Features for Characterized Points

We selected three features for each characterized point, including its x and y coordinates after normalization and the difference between its luminance and the mean of the luminance of the word.

IV. Classification

After feature extraction, we apply classifiers to determine whether the script was forged and in the forged case we also determine which suspect forged the scripts.

A. Binary Classification Problem

To determine whether the script was forged (a binary classification problem), we select the modified support vector machine (modified SVM) with the following loss function for classification:

$$L = \lambda \|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(w \cdot x_i - b)\right) \tag{12}$$

where y_i is determined from the ground truth by

$$y_i = \begin{cases} 1, & \text{for genuine scripts} \\ -1, & \text{for forged scripts} \end{cases}, \tag{13}$$

w is the weight vector, x_i is the feature vector after normalization (training data with mean = 0 and standard=1) and b is the bias. We can adjust a suitable λ to avoid overfitting and achieve better performance.

B. Multi-Label Problem

To determine which of the k suspects forge the script (a multi-label problem), we apply another modified SVM with the following loss function for classification:

$$L = \lambda \|W\|_F^2 + \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} \max\left(0, 1 - y_{i,j} + T_{i,j}\right) \tag{14}$$

where $T_{i,j}$ and $y_{i,j}$ are defined as:

$$T_{i,j} = \left(W_j^T x_i - b_j\right) - y_i\left(W^T x_i - b\right), \tag{15}$$

$$y_{i,j} = \begin{cases} 1, & j = l \\ 0 & j \neq l \end{cases} \tag{16}$$

if the i^{th} training data belongs to the l^{th} class in the ground truth.

In (14)-(16), n is the number of the training data, m is the number of features, y_{ij} is the j^{th} element of k-D row vector y_i, W is the $m \times k$ weight matrix where m is the number of features, W_j is the j^{th} column vector of W, b is a k-D bias vector, and $\|W\|_F{}^2$ is the Frobenius inner product of matrix W.

V. Simulations

Several simulations are performed in this section. In the 1st experiment, we classify the script into two classes (genuine or forged scripts). In the 2nd experiment, we have to classify the scripts into 6 classes (genuine scripts, scripts forged by suspects 1, 2, 3, and 4, and other forged scripts).

A. Scrips Collection

In Experiment 1, we adopted the following scripts:

a. Number of genuine scripts (G): 72
b. Number of non-deliberately forged scripts (NDF): 48
c. Number of deliberately forged scripts (DF): 36

In part a, a volunteer wrote each set of words 24 times using 3 writing instruments (blue, red, and black pens) (24×3 = 72).

In part b, 4 volunteers were asked to imitate the genuine scripts using 3 writing instruments but just did 4 times for each imitator.

In part c, 4 volunteers were asked to practice in advance before imitating the genuine scripts. They also used 3 writing instruments and handed out only 3 scripts that are most similar to the genuine ones.

In Experiment 2, we adopted the following scripts instead:

a. Number of genuine scripts (G): 48
b. Number of forged scripts for 4 suspects: (SF_1~ SF_4): 28 × 4 = 112
c. Other forged scripts (OF) × 56

In part a, a volunteer wrote each set of words 48 times with just blue pens. In part b, 4 volunteers were asked to write 16 sets of forged scripts non-deliberately and 12 sets of forged scripts deliberately. In part c, other volunteers were asked to write 32 sets of forged scripts non-deliberately and 24 sets of forged scripts deliberately.

B. Performance of the Proposed Algorithm

In Experiment 1, we selected 39 of the script sets as the training data and the remained 117 script sets are adopted as the test data. The performance of the proposed method is shown in Table 1, which shows that the proposed algorithm can achieve a high accuracy rate of 94.30% (The comparison with other methods is given in Table 6).

In Fig. 7, we show the distributions of one of the features for genuine scripts (red), non-deliberately forged scripts (blue), and deliberately forged scripts (green). It shows that the proposed features can well distinguish genuine and forged scripts.

In Experiment 2, we selected 101 of the script sets as the training data and the remained 115 script sets are treated as the testing data. There are 6 classes: genuine scripts (G), scripts forged by suspects 1, 2, 3, and 4 (SF_1~ SF_4), and other forged scripts (OF).

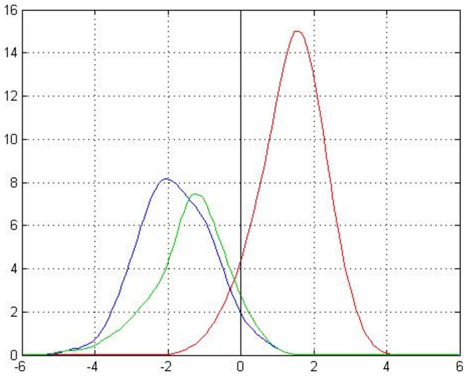

Fig. 7: Distributions of a feature value for genuine scripts (red), non-deliberately forged scripts (blue), and deliberately forged scripts (green) in Experiment 2.

Table 2 shows the performance of the proposed algorithm in Experiment 2. The comparison with other methods is shown in Table 7. These results show that the proposed algorithm also has high performance for suspect identification. The confusion matrix of the proposed algorithm in Experiment 2 is shown in Table 3.

In Fig. 8, we show the distributions of two of the features for the 6 classes (G, SF_1~ SF_4, OF) of scripts. Different classes are shown by different colors. The result shows that these 6 classes can be well separated in the feature domain.

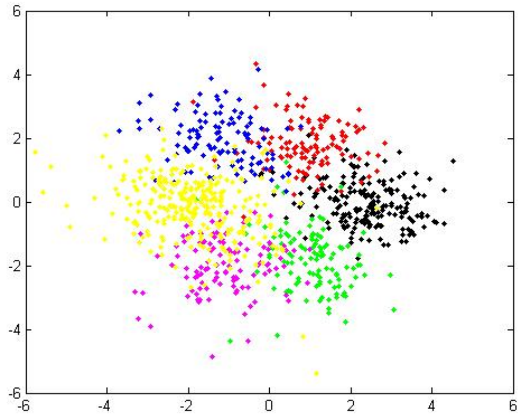

Fig. 8: Distribution of two of the features for the 6 classes of scripts in Experiment 2.

Table 1: Accuracy of the proposed algorithm in Experiment 1 (determining genuine and forged scripts). The comparisons with other methods are shown in Table 6.

	Word	北(North)	新(New)	桃(Peach)	中(Center)	南(South)	高(Tall)	AVE
	accuracy	**89.74%**	**97.44%**	**91.45%**	**94.87%**	**94.87%**	**97.44%**	**94.30%**
pen	blue	92.31%	94.87%	97.44%	94.87%	92.31%	97.44%	94.87%
	red	84.62%	97.44%	92.31%	94.87%	97.44%	97.44%	94.02%
	black	92.31%	100.00%	84.62%	94.87%	94.87%	97.44%	94.02%
type	genuine	98.15%	98.15%	87.04%	94.44%	88.89%	98.15%	94.14%
	forged (non-deliberately)	86.11%	97.22%	100.00%	97.22%	100.00%	97.22%	96.30%
	forged (deliberately)	77.78%	96.30%	88.89%	92.59%	100.00%	96.30%	91.98%

Table 2: Accuracy of the proposed algorithm in Experiment 2 (determining the suspects who forged the scripts). The comparisons with other methods are shown in Table 7.

	Word	春(Spring)	夏(Summer)	秋(Autumn)	冬(Winter)	前(Front)	後(Back)	左(Left)	右(Right)	AVE
	accuracy	**78.26%**	**77.39%**	**78.26%**	**72.17%**	**74.78%**	**86.96%**	**78.26%**	**80.87%**	**78.37%**
type	G	100.00%	95.83%	95.83%	79.17%	87.50%	100.00%	91.67%	95.83%	93.23%
	SF_1	64.29%	100.00%	85.71%	57.14%	71.43%	78.57%	85.71%	92.86%	79.46%
	SF_2	92.86%	71.43%	78.57%	71.43%	92.86%	100.00%	78.57%	92.86%	84.82%
	SF_3	92.86%	78.57%	78.57%	85.71%	64.29%	92.86%	78.57%	71.43%	80.36%
	SF_4	71.43%	85.71%	92.86%	92.86%	85.71%	85.71%	78.57%	85.71%	84.82%
	OF	60.00%	54.29%	57.14%	60.00%	60.00%	74.29%	65.71%	62.86%	61.79%

Note: G means genuine scripts, SF_1, SF_2, SF_3, and SF_4 mean the forged scripts written by suspects 1, 2, 3, and 4, respectively, and OF means other forged.

Table 3: Confusion matrix of the proposed algorithm in Experiment 2.

Actual \ Predicted	G	SF_1	SF_2	SF_3	SF_4	OF
G	179	5	3	0	1	4
SF_1	12	89	2	1	1	7
SF_2	1	4	95	4	1	7
SF_3	1	3	3	90	1	14
SF_4	1	4	3	3	95	6
OF	9	17	36	21	24	173

N=920

C. *Comparison When Only Parts of Features Are Adopted*

In Tables 4 and 5, we show the performance when only parts of features are adopted (1=basic information, 2=energy distribution on 2-D frequency domain, 3=PPMCC with genuine scripts, 4=characterized points). The results show that the best performance can be achieved when all of the four types of features are adopted.

Table 4: Comparison with the cases when only parts of features are adopted in Experiment 1.

Method	**Proposed**	1	2	3
Average Accuracy	**94.30%**	80.91%	81.20%	85.75%
Method	4	1+2	1+3	1+4
Average Accuracy	82.05%	85.75%	90.46%	88.03%
Method	1+2+3	1+2+4	1+3+4	
Average Accuracy	92.74%	91.74%	92.02%	

Table 5: Comparison with the cases when only parts of features are adopted in Experiment 2.

Method	**Proposed**	1	2	3
Average Accuracy	**78.37%**	58.80%	57.93%	59.02%
Method	4	1+2	1+3	1+4
Average Accuracy	61.20%	67.50%	66.41%	67.28%
Method	1+2+3	1+2+4	1+3+4	
Average Accuracy	69.78%	72.93%	73.37%	

Table 6: **Comparison with other algorithms** in Experiment 1 where Best and Worst mean the best and the worst results among the 6 characters.

Methods	Best	Worst	AVE
Proposed	***97.44%***	***89.74%***	***94.30%***
P	75.21%	63.25%	**70.80%**
E	75.21%	60.68%	**70.51%**
BW	64.10%	47.86%	**55.13%**
P+E+BW	82.05%	70.09%	**75.93%**
CNN	80.34%	67.52%	**75.93%**
DNN	96.58%	81.20%	**87.82%**
PCA+SVM	93.16%	82.05%	**87.75%**

Table 7: **Comparison with other algorithms** in Experiment 2 where Best and Worst mean the best and the worst results among the 8 characters.

Methods	Best	Worst	AVE
Proposed	***86.96%***	***72.17%***	***78.37%***
P	59.13%	30.43%	**49.67%**
E	43.48%	32.17%	**38.91%**
BW	40.00%	20.87%	**30.65%**
P+E+BW	58.26%	38.26%	**50.54%**
CNN	66.96%	42.61%	**51.96%**
DNN	80.00%	61.74%	**70.54%**
PCA+SVM	72.17%	57.39%	**62.28%**

D. Comparison of Algorithms

In this subsection, some well-known algorithms were compared, including the conventional methods, including the conventional methods based on projection (P) [1], erosion (E) [2], and boundary-wised labeling (BW) [3]. Moreover, we also compare the proposed algorithm with the methods based on the convolution neural network (CNN) [8], the deep neural network (DNN, with extracted features and 4 hidden layers) [9] and principal components analysis plus the support vector machine (PCA+SVM with extracted features) [10].

In Tables 6 and 7, we compare the accuracies of these algorithms. The results show that the proposed algorithm much outperforms these advanced methods and has very good performance for handwriting verification.

Table 8: **Comparison with manual identification** in Experiment 1.

Methods	Best	Worst	AVE
Proposed	***100.00%***	***84.21%***	***92.98%***
MI top12%	84.21%	71.93%	76.90%

Table 9: **Comparison with manual identification** in Experiment 2.

Methods	Best	Worst	AVE
Proposed	***96.43%***	***78.57%***	***86.61%***
MI top12%	78.57%	64.29%	71.43%

E. Comparison with Manual Identification

In addition to comparing with automatic handwriting verification methods, we also compare the proposed algorithm with the method of manual identification by human beings.

We ask several participants to identify whether the script is forged and the suspect who forged the script. Since there are so many scripts and it is impossible to ask participants to identify all scripts, we randomly selected 8 genuine scripts and 8 forged scripts as training data and 19 scripts as testing data in Experiment 1. Likewise, in Experiment 2, we selected 24 scripts as training data and 28 scripts as testing data. Moreover, only the top 12% manual identification (denoted by MI top 12%) results are considered. The results are shown in Tables 8 and 9.

Conventionally, whether a script was forged is determined manually. However, the results in Tables 8 and 9 show that the proposed algorithm has even better performance than manual identification.

F. Identification after Grouping and Polling

If a script consists of more than 1 characters, we can apply the method of polling to determine whether the script is forged. That is, suppose that we have known that a script consisting of N characters were written by the same person. If more than $N/2$ characters are concluded as forged ones, then we can conclude that the overall script was forged.

We separated 6 characters in Experiment 1 and 8 characters in Experiment 2 into two groups, respectively, and we have known that the words in the same group were by the same person. Then, we use the method of grouping and polling (GAP) to determine whether the script in each group was forged. The performances are shown in the GAP columns of Tables 10, 11, 12.

From these results, we can see that if the script consists of many characters, one can apply the method of grouping and polling to further improve the performance.

Table 10: Accuracy of determining whether the script consisting of several characters was forged in Experiment 1.

word	北	新	桃	***GAP***	LSTM
accuracy	89.74%	97.44%	91.45%	***100%***	96.58%
word	中	南	高	***GAP***	LSTM
accuracy	94.87%	94.87%	97.44%	***100%***	97.44%

Table 11: Accuracy of determining whether the script consisting of several characters was forged in Experiment 2 (First four characters).

word	春 (spring)	夏 (summer)	秋 (autumn)	冬 (winter)	***GAP***	LSTM
accuracy	78.26%	77.39%	78.26%	72.17%	***91.30%***	86.09%
G	100.0%	95.83%	95.83%	79.17%	***100%***	95.83%
SF_1	64.29%	100.0%	85.71%	57.14%	***100%***	92.86%
SF_2	92.86%	71.43%	78.57%	71.43%	***100%***	100%
SF_3	92.86%	78.57%	78.57%	85.71%	***100%***	92.86%
SF_4	71.43%	85.71%	92.86%	92.86%	***100%***	92.86%
OF_1	71.43%	57.14%	57.14%	61.90%	***80.95%***	71.43%
OF_2	42.86%	50.00%	57.14%	57.14%	***57.14%***	57.14%

Table 12: Accuracy of determining whether the script consisting of several characters was forged in Experiment 2 (Last four characters)

word	前(front)	後(back)	左(left)	右(right)	***GAP***	LSTM
accuracy	74.78%	86.96%	78.26%	80.87%	***92.17%***	85.22%
G	87.50%	100.0%	91.67%	95.83%	***100%***	100%
SF_1	71.43%	78.57%	85.71%	92.86%	***100%***	92.86%
SF_2	92.86%	100.0%	78.57%	92.86%	***100%***	92.86%
SF_3	64.29%	92.86%	78.57%	71.43%	***100%***	85.71%
SF_4	85.71%	85.71%	78.57%	85.71%	***100%***	100%
OF_1	66.67%	71.43%	61.90%	66.67%	***71.43%***	71.43%
OF_2	50.00%	78.57%	71.43%	57.14%	***78.57%***	50.00%

In addition, Tables 10-12 show that the method of GAP also has better performance than long short-term memory network (LSTM, with extracted features in from the words that have been processed) [14].

G. *Decision Rules for Different Cases*

Moreover, we can apply the following decision rule:

$$\begin{cases} \text{if } w\cdot x-b\geq t: & \text{viewed as a genuine script,} \\ \text{if } w\cdot x-b<t: & \text{viewed as a forged script,} \end{cases} \tag{17}$$

and adjust t for different applications in Experiment 1.

In academic research, we hope that we can minimize the total error rate (TER). Therefore, we choose $-0.4 \leq t \leq 0.4$ in this condition.

In criminal prosecution, we have to reduce the false rejection rate (FRR) as much as possible to avoid convicting an innocent person. Therefore, we choose $t < -0.4$ in this case.

Oppositely, in finance verification, each forged signature may cause a lot of loss. Therefore, we have to reduce the false accept rate (FAR) as much as we can and choose $t > 0.4$. In Table 13, we show the result of using different values t in (17) and the possible application in each case.

Table 13: Effect of choosing different values of t in (17) and their suitable applications.

t	TER	FAR	FRR	Suitable application
-1.2	21.37%	39.68%	**0.00%**	criminal prosecution
-0.8	13.82%	25.40%	**0.31%**	
-0.4	**7.83%**	12.43%	2.47%	academic research
0.0	**5.70%**	5.56%	5.86%	
0.4	**6.41%**	2.65%	10.80%	
0.8	9.54%	**0.26%**	20.37%	finance verification
1.2	14.96%	**0.00%**	32.41%	

VI. Conclusion

In this paper, an advanced algorithm to identify whether the scripts were forged and which suspect forged the script was proposed. With the frequency domain features, the characterized point features, and the modified SVM using the loss function consisting of the regularity term, a high accurate handwriting verification result can be achieved. Simulations show that the proposed algorithm outperforms deep-learning based methods and the manual identification method. Moreover, the proposed algorithm is robust to writing instrument and luminance and suitable for the practical applications in criminal investigation.

References

[1] Z. Zhang, C. L. Liu, and M. Zhao, "Handwriting representation and recognition through a sparse projection and low-rank recovery framework," in *Int. Joint Conference on Neural Network*, Dallas, pp. 1-8, 2013.

[2] J. Tan, J. H. Lai, C. D. Wang, and M. S. Feng, "Off-line Chinese handwriting identification based on stroke shape and structure," in *Int. Conf. Information Engineering and Computer Science*, Wuhan, China, pp. 1-4, 2010.

[3] S. D. Chame and A. Kumar, "Overlapped character recognition: An innovative approach," in *IEEE Int. Conf. Advanced Computing*, Bhimavaram, India, pp. 464-469, 2016.

[4] P. Kumawat, A. Khatri, and B. Nagaria, "Offline handwriting recognition using invariant moments and curve let transform with combined SVM-HMM classifier," in *International Conference on Communication Systems and Network Technologies*, Gwalior, pp. 144-148, 2013.

[5] A. T. Nasser and N. Dogru, "Signature recognition by using SIFT and SURF with SVM basic on RBF for voting online," in *Int. Conf. Engineering and Technology*, Antalya, pp. 1-5, 2017.

[6] Y. Luo, R. Xia, and M. Abdulghafour, "Offline Chinese handwriting character recognition through feature extraction," in *Int. Conf. Computer Graphics, Imaging and Visualization*, pp. 394-398, 2016.

[7] C. H. Leung, Y. S. Cheung, and Y. L. Wong, "A knowledge-based stroke-matching method for Chinese character recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 17, issue 6, pp. 993-1003, 1987.

[8] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Int. Conf. Document Analysis and Recognition*, Beijing, China, pp. 1135-1139, 2011.

[9] D. Ciresan and U. Meier, "Multi-column deep neural networks for offline handwritten Chinese character classification," in *Int. Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, pp. 1-6, 2015.

[10] V. Deepu, S. Madhvanath, and A. G. Ramakrishnan, "Principal component analysis for online handwritten character recognition," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Cambridge, UK, pp. 327-330, 2004.

[11] H. N. Champa and K. R. AnandaKumar, "Automated human behavior prediction through handwriting analysis," in *Int. Conf. Integrated Intelligent Computing*, Bangalore, India, pp.160-165, 2010.

[12] P. X. Lee, J. J. Ding, T. C. Wang, and Y. C. Lee, "Automatic writer verification algorithm for Chinese characters using semi-global features and adaptive classifier," in *IEEE Int. Conf. Multimedia & Expo Workshops (ICMEW)*, pp.1-6, 2018.

[13] S. C. Pei and J. J. Ding, "New corner detection algorithm by tangent and vertical axes and case table," *IEEE Int. Conf. Image Processing*, Genova, Italy, vol. 1, pp. 365-368, 2005

[14] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *Int. Conf. Frontiers in Handwriting Recognition*, Heraklion, Greece, pp. 285-290, 2014.