Spiral Based Run-Length Features for Offline Signature Verification

Walid Bouamra[1,2](✉), Moises Diaz[2], Miguel Angel Ferrer[2], and Brahim Nini[1]

[1] Larbi ben M'hidi University, Oum El Bouaghi, Algeria
{walid.bouamra,b.nini}@univ-oeb.dz

[2] Institute IDeTIC, Universidad de Las Palmas de Gran Canaria, Las Palmas, Spain
Walid.bouamra101@alu.ulpgc.es, {moises.diaz, miguelangel.ferrer}@ulpgc.es

Abstract. Automatic signature verification is one of the main modes to verify the identity of the individuals. Among the strategies to describe the signature in the verifiers, run-length features have attracted the attention of many researchers. This work aims to upgrade the classical run-length distribution as an additional representation for off-line signatures. Specifically, we add a fifth direction to the four classical directions of run-length features. Such fifth direction runs the signature in a spiral way providing an outside to inside view of the signature. This paper evaluates the performance of the new run-length direction combined with the classical ones. For classification purposes, we used a one-class support vector machine. Additionally, we study how to combine the new direction with the previous four original ones at both feature and score levels. Our results validate the use of this novel direction in run-length features in our own experiments and external international competition in signature verification.

Keywords: Spiral run-length features · Four-directions run-length features · Offline signature verification · OC-SVM · Feature fusion · Score fusion · Signature verification competition

1 Introduction

Biometrics has become more and more an important need for automatically verifying individuals and evenly for the security of enterprises. Nowadays, among the different modalities of biometrics, the signature remains a very confident, lawfully, and socially accepted modality for verifying identities [1].

To design an Automatic Signature Verifier (ASV), the literature proposes to use two approaches: Writer-Dependent and Writer-Independent [1]. In the first approach, the samples of each individual are trained by a classifier separately from others, whereas in the Writer-Independent approach, only one classifier is used to train all the writers' signatures. In both cases, the aim is to verify whether a questioned signature is genuine or forgery.

Since the texture remains one of the main discriminant characteristics to extract useful information from the images, many ASV systems are based on textural features

© Springer Nature Switzerland AG 2022
C. Carmona-Duarte et al. (Eds.): IGS 2021, LNCS 13424, pp. 26–41, 2022.
https://doi.org/10.1007/978-3-031-19745-1_3

for the signature image analysis and pattern recognition process. In our work, we propose to use a novel handcrafted feature for off-line signature verification based on both textural properties and run-length distributions.

The run-length features have been a favored method in several fields of image processing. They present one of the used features for image classification [2, 3], writer identification [4], and in our case, for offline signature verification. In the latter domain, they give a powerful spatial presentation of pixels, and under the concept of runs [2]. Typically, such spatial distribution is achieved by counting the runs in four directions: horizontal, vertical, and two diagonal directions.

However, a major problem is the well-known high intra-class variability of a user signature. It could be mainly due to changes in shape, size, or other visual aspects, which causes a spatial distribution distortion within the image signature of a user. All this limits the classic run-length features performance.

The main contribution of this paper is the definition of a new direction in the framework of run-length features. This new direction is named spiral direction, which adds a new representation of the image. Moreover, we combine this new direction to the classical four directions to improve the representation of the run-length features. Our work aims to study the efficiency of run-length features when adding the spiral direction for off-line ASV.

It is expected that this new direction will expand the run-length limitations due to its flexibility within the orientation and the size of the scanned lines, which raises its robustness regarding the intra-class variability, and compensates the static of each direction of the run-length features, that traverses the image line by line in only one given direction.

The paper is organized as follows: Sect. 2 includes some related works on run-length features in off-line signature verification. Section 3 defines the previous run-length features whereas the proposed spiral run-length feature is given in Sect. 4. Section 5 is devoted to the experiments and results. We close the paper by the conclusion in Sect. 6.

2 Related Works on Run-Length Features

Many techniques have been used for image texture analysis in signature verification [1, 2]. Run-length features are one of the textural descriptors basing on the lengths of runs. A run can be explained as a set of consecutive pixels in a given direction having the same value [3]. The length of the run is the number of pixels composing this run.

As a consequence, we work out the run-length histograms, which are composed of the numbers of runs of different lengths. This process is generalized for the four principal directions, Horizontal (0°), Vertical (90°), right-diagonal (45°) and left diagonal (135°). As a result, it gives four feature vectors comprising the four directions.

In 1975, Galloway [3] applied the run-length features to a set of textures representing nine terrain types, each one with six samples. He arranged two adjustments on the run-length technique to obtain numerical texture measures: the first one was based on all diagonal run lengths should be multiplied by $\sqrt{2}$, while the second one was the short-run emphasis function. The classification results were quite promising.

The use of run-length features has been spread frequently in the field of texture analysis. Further, they have been adopted for purposes related to handwriting, such as writing or writer identification and verification and, more specifically, the verification of off-line handwritten signatures.

Djeddi et al. [5] applied the run-length and the 2D autoregressive coefficient features in signature verification. They used 521 writers from the GPDS960 dataset and the Support Vector Machine as the classifier. They performed the run-length on black pixels which correspond to the ink trace of the signatures and considered only runs of a maximum of 100 pixels for each direction (0°, 45°, 90°, 135°). A final vector of 400 values was obtained as a feature vector of 100 values per direction.

Serdouk et al. [6] proposed a combination of two data features, the orthogonal combination of local binary patterns and the Longest Run Features (LRF). The LRF calculated the connected pixels through the four principal directions: horizontal, vertical, right diagonal and left diagonal. For each direction, the longest run of the signature pixels was selected, the total sum of these numbers (lengths) constituted the LRF value in the given direction. This procedure was repeated for the remaining directions in order to get four LRF features. Finally, the four LRF features were combined with the other features to define each image-based signature. The proposed features were employed on GPDS300 and CEDAR databases, using SVM classifiers for the automatic verification task.

In Bouamra et al. [2], a new off-line ASV was designed by using run-length features. They were applied to black and white pixels, which corresponded to the signature and the background, respectively. The four run-length vectors for each color contained 400 values and the black and white output feature vector had, therefore, 800 values. They used only genuine signatures for training and employing the 881 writers of the GPDS960 (281 users for generating signature models and choosing optimal threshold, and 600 for the evaluation step). The One-Class Support Vector Machine (OC-SVM) was used for the classification phase. Some standard metrics were used to quantify the performance of the system, obtaining competitive performances.

In another work related to the prior one, Bouamra et al. [8] implemented multidirectional run-length features for automatic signature verification. The new features were based on the standard run-length features [2], with four supplementary angles added to the four primary directions: horizontal, vertical, left-diagonal, and right-diagonal direction; each angle is enhanced by its neighborhood to generate a composite one formed by three adjacent angles. Finally, eight composite angles are obtained as explicit orientations for scanning the signature image. The researchers employed the OC-SVM as a classifier to apply their features on the GPDS960 database.

The run-length features were also used on off-line ASV by Ghanim and Nabil [7]. In their study, they used different features including run-length, slant distribution, entropy, the histogram of gradient features and geometric features. Then, they applied machine learning techniques on the computed features like bagging trees, rand forest and support vector machines. The study aimed to calculate the accuracy of different approaches and to design an accurate system for signature verification and forgery detection. The Persian Offline Signature Data-set was utilized for evaluating the system, and the obtained results were satisfactory.

3 Classical Run-Length Features

Let be assumed a binarized image-based signature, in run-length histograms, $RL_b(i|\theta)$ is the *(i)th* element describing the number of runs with black values and length *i*, occurring in the image along an angle θ. Thus, $RL_w(j|\theta)$ is the *(j)th* element describing the number of runs with white value and length *j* occur in the image along angle θ.

Let's indicate the following notations:

- RL_b is the number of black run lengths in the image.
- RL_w is the number of white runs lengths in the image.
- N_B is the black run-length histograms for the four directions.
- N_W is the white run-length histograms for four directions.
- RL_4D is the Global black and white Run-Length histograms for four directions.

The black and white run-length histograms are defined, respectively, as:

$$RL_b(\theta) = \sum\nolimits_{i=1}^{RL_b} N_b(i|\theta) \tag{1}$$

$$RL_w(\theta) = \sum\nolimits_{j=1}^{RL_w} N_w(j|\theta) \tag{2}$$

$$\forall\ 1 \le i \le N_b \ \mathit{and}\ 1 \le j \le N_w.$$

The black and white run-length histograms for a given direction are concatenated as

$$RL(\theta) = [RL_b(\theta), RL_w(\theta)] \tag{3}$$

According to the pixel color, the black and white run-length histograms for the four directions are processed as:

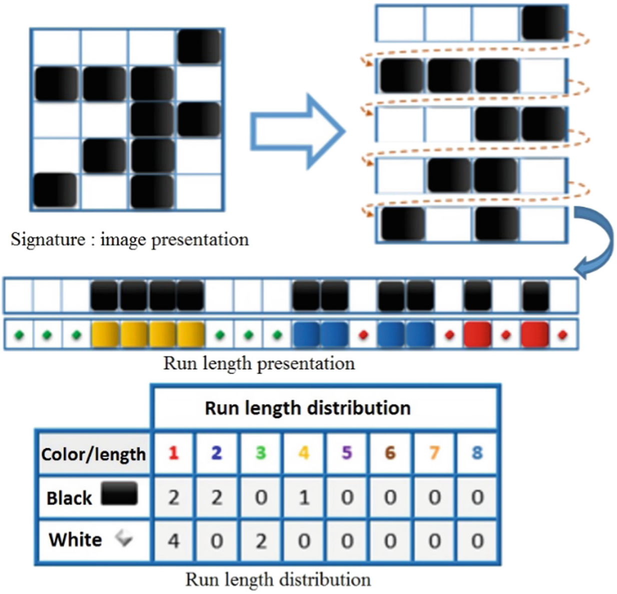
$$RL_B = \left[RL_b\left(0^\circ\right), RL_b\left(45^\circ\right), RL_b\left(90^\circ\right), RL_b\left(135^\circ\right)\right] \tag{4}$$

$$RL_W = \left[RL_w\left(0^\circ\right), RL_w\left(45^\circ\right), RL_w\left(90^\circ\right), RL_w\left(135^\circ\right)\right] \tag{5}$$

where the final feature vector based on run-length histograms are concatenated as [2]:

$$\begin{aligned} RL_{4D} = [RL_B, RL_W] = [&RL_b(0^\circ), RL_b(45^\circ), RL_b(90^\circ), RL_b(135^\circ), RL_w(0^\circ), \\ &RL_w(45^\circ), RL_w(90^\circ), RL_w(135^\circ)] \end{aligned} \tag{6}$$

In our work, we vectorized the 2D image to get a single long line. At this level, the run-lengths are calculated for both black and white pixels. This procedure is applied to the other three directions, i.e. vertical, right-diagonal and left-diagonal. In another meaning, before calculating the lengths of runs, we juxtaposed the lines of the image in the desired direction, line by line in a way to form a single vector that denotes a new different presentation of the image. On this vector, we apply the same algorithm to calculate the Run-Length distributions for this given direction, and so for the other directions.



	Run length distribution							
Color/length	1	2	3	4	5	6	7	8
Black	2	2	0	1	0	0	0	0
White	4	0	2	0	0	0	0	0

Fig. 1. Run-length distribution for the horizontal direction

In Fig. 1 we illustrated a toy example of this procedure for the horizontal direction. In the black pixels, we observed that there is no run of length one, two runs of length two, one run of length three and one run of length four, as indicated in the first row. A similar observation can be made for white pixels. As such, the final horizontal vector is about 800 values (400 + 400 for black and white pixels, respectively). The procedure is repeated for the remaining directions. The resultant run-length feature vector has 3200 values due to the final concatenation of the four directions.

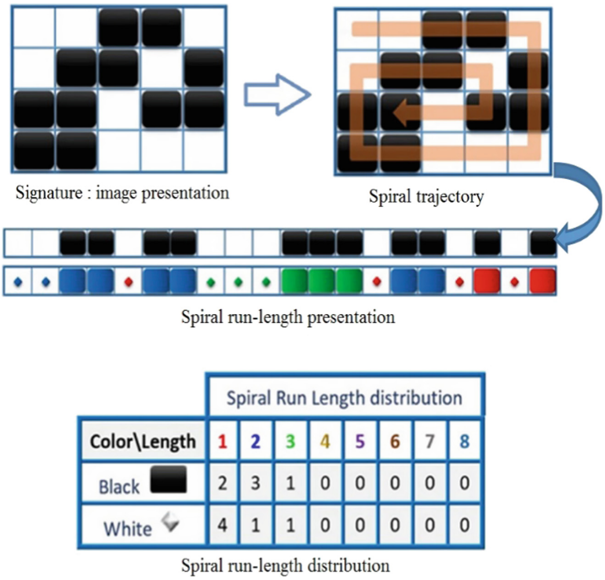
4 Spiral Run-Length Features

In this section, we describe first the proposed spiral run-length feature. Next, we propose two combinations to fuse the new feature with the previous four directions.

4.1 Spiral Feature Vector

A uniform displacement describes it on a rotating line until reaching a final center point. This way, the spiral run-length feature traverses the entire image in a spiral counter-clockwise curve starting from the first pixel at the upper left corner of the image. Then it moves away more and more towards a last central point. This spiral movement rotates between the horizontal and the vertical directions. The procedure is shown in Fig. 2.

It could be said that the spiral feature treats four orthogonal directions differently, as shown in Fig. 2. The movement hither is done permanently, starting with a horizontal direction with an angle $\theta_1 = 0°$, followed by a descending vertical scan with an angle $\theta_2 = -90°$. On reaching the end of the vertical column, the direction changes again moving towards the horizontal direction but on the contrary direction to the first angle with an angle of $\theta_3 = 180°$. The last direction to progress is the vertically upward direction by exploring the entire column from bottom to top on an angle $\theta_4 = 90°$. This round of four directions is iterated until browsing the entire signature image.


Signature : image presentation

Spiral trajectory

Spiral run-length presentation

	Spiral Run Length distribution							
Color\Length	1	2	3	4	5	6	7	8
Black ■	2	3	1	0	0	0	0	0
White	4	1	1	0	0	0	0	0

Spiral run-length distribution

Fig. 2. New run-length direction: spiral based feature.

For counting the length of runs, the same procedure described in Sect. 3 is applied to the resulting vector of the spiral function. Accordingly, the final spiral vector size contains 800 values (400 for black pixels + 400 for white ones).

We consider the next notations:

- SP_B is the number of black run lengths in the image.
- SP_W is the number of white runs lengths in the image.
- N_b is the black run-length histograms in spiral direction.
- N_w is the white run-length histograms in spiral direction.
- SP is the global black and white run-length histograms in spiral direction.
- θ_k is the browsing spiral angle:

$$\theta_1 = 0°, \theta_2 = -90°, \theta_3 = 180°, \theta_4 = 90°.$$

The black and white run-length histograms are defined, respectively, as follows:

$$SP_B = \sum_{i=1}^{SP_B} \sum_{k=1}^{4} N_b(i|\theta_k) \tag{7}$$

$$SP_W = \sum_{j=1}^{SP_W} \sum_{k=1}^{4} N_w(j|\theta_k) \tag{8}$$

$$\forall \quad 1 \leq i \leq SP_B \; and \; 1 \leq j \leq SP_W.$$

The global Spiral Run-Length histograms are then concatenated as

$$SP = [SP_B, SP_W] \tag{9}$$

Therefore, the spiral transformation of the image is dynamic in direction (two changes: vertical/horizontal) and in orientation (two changes for every direction: (→, ←) and (↑, ↓)). It is also dynamic in size; with every change of direction, we subtract a pixel. This transformation is based on four changes of the directions, and every current movement is starting from the second pixel (the first of this current movement is the last of the precedent one, so it is already calculated).

The spiral feature regroups both of two horizontal and vertical directions at the same time. It helps to add complementary information to the four previous run-length directions. Thus, the spiral run-length feature can be considered as the fifth direction.

Algorithm 1. Spiral vector extraction

```
    Input
    Read_image;
    Initialize
    [M, N] ← Image_size;
    i←1; j←1;
    SP ←[]; % Spiral vector
    While (i < =M) and (j<=N) do
      Begin
      image=image(i:M,j:N);
      [M, N] ← Image_size;
      If i=M % the current image zone  is composed of  one row (or one pixel)
      SP ← [SP, first  row(j:N  | 0°)];
      Else if j=N % the current image zone  is composed of one column (with more than one
pixel)
        SP ← [SP, first_row(j:N  | 0°), last_column(i+1:M  | -90°)];
        Else if i=M-1 % the current image zone is composed of two lignes (with more than
one column)
          SP←[SP, first_row(j:N  | 0°), last_column(i+1:M  | -90°), last_row(N-1:j:-1
| 180°)];
          Else % General case
            SP←[SP, first_row(j:N| 0°),last_column(i+1:M| -90°),last_row(N-1:j:-1| 180°),
first_column(M-1:i+1:-1| 90°)];
      End_If
      i++;j++; M--, N --;
      End;
    End While;
```

The steps of the proposed feature are highlighted in the pseudo-code Algorithms 1 and 2. They describe the spiral vector extraction and the spiral run-length features, respectively.

Algorithm 2. Spiral run-length features extraction

```
Input
Read_image;
Initialize
i←1; j←1;
[M, N] ←Image_size;
Spiral_Black_Hist← zeros (1,400); % Black Histograms
Spiral_White_Hist ← zeros (1,400); % White Histograms
Spiral_Hist ← zeros (1,800); % Spiral_Histograms
Read(SP); % Spiral vector outcoming from image by spiral transformation.
For current_pixel = 1: Length(Spiral_Vector) do
    Begin
    Score ← 0;
    If current_pixel=0
        REPEAT
        score ++;
        Go to next pixel;
        UNTIL (current_pixel_value different from next_pixel_value) or (score == 400)
        Spiral_Black_Hist(score)= Spiral_Black_Hist(score)+1;
    Else
        REPEAT
        score ++;
        Go to next pixel;
        UNTIL (current_pixel_value different from next_pixel_value) or (score == 400)
        Spiral_White_His(score)= Spiral_White_Hist(score)+1;
    End_If
End_For
Spiral_Hist=[Spiral_Black_Hist,Spiral_White_Hist];
```

4.2 Combining Spiral with the Previous Directions

Two combinations are proposed to use the new spiral feature along with the previous run-length features. Specifically, they consist of combining the run-lengths features at the feature and score level.

On the feature level, the combination consists of concatenating the four run-length features and the spiral feature. On the one hand, we concatenate all the five black run-length histograms and, on the other hand, the five white run-length histograms. This way, the combined histograms contain the five directions. Let *RL_5D* be the combined run-length histograms, it is defined as follows:

$$RL\_5D = [RL_B, SP_B, RL_W, SP_W]$$

$$\begin{aligned} RL_{5D} = [&RL_b(0°), RL_b(45°), RL_b(90°), RL_b(135°), SP_B, \\ &RL_w(0°), RL_w(45°), RL_w(90°), RL_w(135°), SP_W] \end{aligned} \tag{10}$$

On the score level combination, this fusion is concerned by the scores generated by classifiers. The global score is a combination of the two scores of the previous four run-length features and the spiral one. A weight sum of the two scores performs the combination:

$$Sc = \alpha.\mathrm{Sc}_1 + (1 - \alpha).\mathrm{Sc}_2 \tag{11}$$

Sc being the final score, Sc_1 being the score of four directions run-length features and Sc_2 being the score of the spiral one, we heuristically set α in 0.5. In both cases of features, we process the black and white pixel distribution.

The experiments are carried out on each of the two levels of combination, with further details provided in the next passage.

5 Experiments

In this section, we present the used databases, the experimental protocol and the experiments with the two types of combinations: at both feature and score level when run-length features are used in ASV.

5.1 Database

We used the following two databases to evaluate our system:

GPDS75 Database. This database was introduced by Ferrer et al. [9]. It contains the first 75 writers; each one has 24 genuine signatures and 30 skilled forgeries.

CEDAR Database. IT is one of the most frequently used database for off-line ASV [10]. This database comprises a total of 55 signatures of different signers. Each individual signed 24 genuine signatures and has a total of 24 forged specimens.

5.2 Preprocessing

Our experiments necessitated the preprocessing phase since both GPDS75, and CEDAR datasets contain greyscale signatures, whereas our system's application relies mostly on binary signatures.

The signatures were first extracted from the datasets, then binarized using Otsu's method [17, 23], which involved determining a global threshold from the greyscale signature image. The threshold was accordingly employed to transform the greyscale signature into a binary signature by reducing the intra-class variance of the thresholded pixels.

5.3 One-Class Support Vector Machine

The availability of positive and negative training examples is one of the criteria of a classic Support Vector Machine (SVM) classifier.

The OC-SVM classifier employs only the genuine signatures for the training. The target class is discriminated from all other classes using only training data from the target class. The objective is to achieve a border that separates the target class examples from the rest of the space, a barrier that takes as many examples as possible targets [2, 11]. This border is defined by a decision function that is positive within a class S but negative outside of S: ($\overline{\mathrm{S}}$) as described in Fig. 3.

$$f(x) = \begin{cases} +1 \ \mathit{if} \ x \in S \\ -1 \ \mathit{if} \ x \in \overline{S} \end{cases}$$

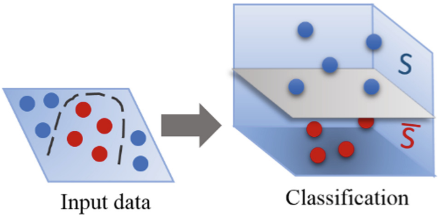


Fig. 3. One-class SVM classification

The parameters to be determined for the OC- SVM include the proportion of outliers ($\vartheta \in [0\ 1]$) and the radial basis function kernel parameter ($\gamma \in [0\ 1]$). The RBF kernel was chosen after experimenting with several kernel functions [2].

5.4 Experimental Protocol

Our signature verification system comprises four steps: selecting a set of signers, building the signature models, locating the optimal decision threshold, and finally achieving the classification step.

The set of signers to be selected includes the first five (R5) and ten (R10) genuine signatures that are kept as reference signatures in the training stage. Then, the testing stage is conducted by employing the next ten genuine samples (g6…g15 in the case of R5, and g11…g20 in the case of R10) and the first ten skilled forgeries (f1…f10) for the experiments in both databases.

The optimal decision threshold is deduced from the false rejection rate (FRR) and the false acceptance rate (FAR) curves using the equal error rate (EER) [24, 25], as described in the next figure. The choice of the (EER) metric, which is defined as the system error rate when FRR = FAR [24], was chosen since it has been used in a variety of relevant studies (Fig. 4).

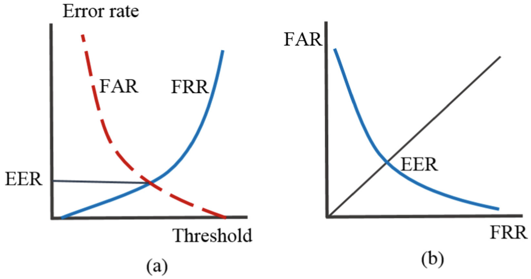


Fig. 4. EER performance measure

5.5 Results

We discuss here the combination at two levels: feature level and score level. The results of such fusions on GPDS75 and CEDAR databases are shown in Table 1 and Table 2, respectively.

For the feature level combination applied to the GPDS75 database, we gained EER = 9.24% and EER = 8.26% using 5 and 10 reference samples, respectively. Whereas, using the same database and number of references for the score level combination generated EER = 7.98% and as the best outcome we earned EER = 6.86%.

On the other hand, employing the CEDAR database affected the results illustrated in Table 2. The feature level fusion gained EER = 0.55% and EER = 0.36%, respectively, with 05 and 10 reference samples, while the results attained are EER = 0.73% and EER = 0.18% performing the score level fusion. This last outcome (EER = 0.18%) is the best value obtained operating the score level combination with 10 reference samples.

For both types of fusion, the experimental results in Tables 1 and 2 reveal that fusing the features raises the rate and improves system performance.

Table 1. Results in EER (%) on GPDS75 by combining at feature and score level.

System	GPDS-75	
	R5	R10
Basic RL (RL)	10.78	9.38
Spiral RL (SP)	12.88	11.62
[4RL, SP]*: Feature level	**9.24**	**8.26**
[4RL, SP]*: Score level	**7.98**	**6.86**

*[4RL, SP]: Combination of the classical run-length (4RL) features with the spiral one.

Table 2. Results in EER (%) on CEDAR by combining at feature and score level.

System	CEDAR	
	R5	R10
Basic RL (RL)	0.73	0.55
Spiral RL (SP)	0.91	0.55
[4RL, SP]*: Feature level	**0.55**	**0.36**
[4RL, SP]*: Score level	**0.73**	**0.18**

*[4RL, SP]: Combination of the classical run-length (4RL) features with the spiral one.

Furthermore, we compare our results with previous works. Table 3 shows different works that have used the GPDS75 database. We can observe that our performances are in line with state of the art. For instance, Maergner et al. obtained the best EER = 6.49%, while in another work they got an EER = 6.84%. When we combine the five run-length features at the score level, our best performance was 6.86% on GPDS75.

According to Table 4, our results were competitive compared with previous works in CEDAR database. We observe a gap getting two minimal rates: EER = 0.18% and EER = 0.36%, followed by Hamadene et al. with AER = 2.10%, then Hafemann et al. with EER = 4.63% accompanied by Sharif et al. with EER = 4.67%. We conclude that our system was more performant with CEDAR database than GPDS75 database.

Table 3. Results on GPDS75 – comparison between the state-of-the-art and our system.

Reference	Samples/user	EER %
Maergner et al. [12]	10	6.84
Maergner et al. [13]	10	9.42
Maergner et al. [14]	10	6.49
Ferrer et al. [15]	10	16.01
This work (score level)	**5**	**7.98**
This work (score level)	**10**	**6.86**

Table 4. Results on CEDAR – comparison between the state-of-the-art and our system.

Reference	Samples/user	EER %
Guerbai et al. [11]	12	5.6
Sharif et al. [16]	12	4.67
Hafemann et al. [17]	12	4.63 (±0.42)
Hamadene et al. [18]	5	2.10
This work (feature level)	**10**	**0.36**
This work (score level)	**10**	**0.18**

5.6 Spiral Run-Length Features in External Competition

The evaluation of spiral run-length features against other handwritten signature verification systems was a critical step. For this reason, we submitted our features to the international competition on Short answer Assessment and Thai Student Signature and Name Components Recognition and Verification (SASIGCOM 2020) [19] which was organized in conjunction with the 17th International Conference on Frontiers in Handwriting Recognition (ICFHR 2020).

In the competition, six tasks were prepared for the competitors including the signature verification task, the thai student signature dataset was employed for this task shown in Table 5. Three type of forgery were adopted: simple, skilled and random forgeries. The Equal Error Rate (EER) was employed as the judge the different participating systems performance.

Table 5. Signature verification dataset (SASIGCOM 2020).

Dataset	No. of users	Train	Test		
		Gn*	Gn*	Skld*	Smpl*
Thai student signature dataset	100	5	25	12	12

*Gn: Number of genuine samples/user. *Skld: Number of skilled samples/user.
*Smpl: Number of simple samples/user.

Our system based on the spiral run-length feature get EER = 0.1108% for the random forgeries, EER = 0.2045% for the skilled forgeries and an EER = 0.1459% for simple forgeries with an average of 0.1537%. The results cited in Table 6 show also that the classical run-length features get EER = 0.1308%, EER = 0.2145% and EER = 0.1599% for random, skilled and simple forgeries, while the multidirectional run-length feature obtained an average of 0.1415%. The first ranking was for a learned system with EER = 0.0019%, EER = 0.0710% and EER = 0.0090% for the same forgeries types respectively with an average of 0.0273%.

Table 6. Results of the signature verification task (SASIGCOM 2020).

Rank	Algorithm	EER %			
		Random forgeries	Skilled forgeries	Simple forgeries	AVG
1	SCUT-CNN [20]	0.0019	0.0710	0.0090	0.0273
2	LTP + oBIFs [21]	0.0109	0.1091	0.0712	0.0637
3	ERL [2]	0.0302	0.1780	0.0955	0.1012
4	oBIFs [21]	0.0444	0.1876	0.1010	0.1110
5	LTP [21]	0.0511	0.1901	0.1105	0.1172
6	MDRL [8]	0.0986	0.2000	0.1258	0.1415
7	SPIRAL-RL	0.1108	0.2045	0.1459	0.1537
8	RL400 [2]	0.1308	0.2145	0.1599	0.1686
9	RL [4]	0.1308	0.2145	0.1599	0.1686
	Benchmark [22]	0.0201	0.1108	0.0031	0.0447

According to Tables 3, 4, and 6, we notice that the different systems' results obtained by using the GPDS75 database are more elevated than those acquired by using the CEDAR and the SASIGCOM databases.

More clearly, our system could reach very lowered EER values using the CEDAR database; this differentia is due to the system-dataset ratio. How the system scrutinizes the signature, the characteristic of each database, and how the signatures were pre-processed before including them in the database. For instance, the GPDS75 dataset is greyscaled, whereas the SASIGCOM database signatures are already binarized. Also, the background of the GPDS database is almost similar, whereas we find a difference in the CEDAR signatures background between the genuine and the forged signatures.

6 Conclusion

In this work, we propose a new direction for run-length features based on the signature's spiral path. We observe performance improvements by combining the previous well-known four directions in run-length features with the proposal spiral direction. Thus, the spiral run-length feature can be understood as the fifth direction, which is more robust to intra-class variability and get better results than using only the four run-length features. In this work, we show results when combining the run-length features at the feature and score level, obtaining better performances at score level combination.

In our future works, we seek to improve the performance of automatic signature verification by applying other techniques of fusion and combination. In addition, we study other methods to process the run-length features and to extend its use in on-line signatures.

References

1. Diaz, M., Ferrer, M.A., Impedovo, D., Malik, M.I., Pirlo, G., Plamondon, R.: A perspective analysis of handwritten signature technology. ACM Comput. Surv. (CSUR) **51**(6), 117 (2019)
2. Bouamra, W., Djeddi, C., Nini, B., Diaz, M., Siddiqi, I.: Towards the design of an offline signature verifier based on a small number of genuine samples for training. Expert Syst. Appl. **107**, 182–195 (2018)
3. Galloway, M.M.: Texture analysis using gray level run lengths. Comput. Graph. Image Process. **4**(2), 172–179 (1975)
4. Djeddi, C., Siddiqi, I., Souici-Meslati, L., Ennaji, A.: Text-independent writer recognition using multi-script handwritten texts. Pattern Recogn. Lett. **34**(10), 1196–1202 (2013)
5. Djeddi, C., Siddiqi, I., Al-Maadeed, S., Souici-Meslati, L., Gattal, A., Ennaji, A.: Signature verification for offline skilled forgeries using textural features. In: 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp. 76–80. IEEE (2015)
6. Serdouk, Y., Nemmour, H., Chibani, Y.: Combination of OC-LBP and longest run features for off-line signature verification. In: Conference: Proceedings - 10th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2014 (2015). https://doi.org/10.1109/SITIS.2014.36
7. Ghanim, T., Nabil, A.: Offline signature verification and forgery detection approach. In: 13th International Conference on Computer Engineering and Systems (ICCES) (2018). https://doi.org/10.1109/ICCES.2018.8639420
8. Bouamra, W., Díaz, M., Ferrer, M.A., Nini, B.: Off-line signature verification using multidirectional run-length features. In: ICIST 2020, vol. 43, pp. 1:8–43:8 (2020)
9. Ferrer, M.A., Travieso, C., Alonso, J.: Off-line handwritten signature GPDS-960 corpus. In: Ninth International Conference on Document Analysis and Recognition, ICDAR 2007, vol. 2, pp. 764–768. IEEE (2007)
10. Kalera, M.K., Srihari, S., Xu, A.: Offline signature verification and identification using distance statistics. Int. J. Pattern Recogn. Artif. Intell. **18**(07), 1339–1360 (2004). https://doi.org/10.1142/S0218001404003630
11. Guerbai, Y., Chibani, Y., Hadjadji, B.: The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. Pattern Recogn. **48**(1), 103–113 (2015). https://doi.org/10.1016/j.patcog.2014.07.016
12. Maergner, P., Howe, N., Riesen, K., Ingold, R., Fischer, A.: Offline signature verification via structural methods: graph edit distance and inkball models. In: ICFHR 2018, pp. 163–168 (2018)
13. Maergner, P., Riesen, K., Ingold, R., Fischer, A.: A structural approach to offline signature verification using graph edit distance. In: Proceedings of International Conference on Document Analysis and Recognition (ICDAR), pp. 1216–1222. IEEE (2017)
14. Maergner, P., et al.: Offline signature verification by combining graph edit distance and triplet networks. In: Bai, X., Hancock, E.R., Ho, T.K., Wilson, R.C., Biggio, B., Robles-Kelly, A. (eds.) S+SSPR 2018. LNCS, vol. 11004, pp. 470–480. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97785-0_45
15. Ferrer, M.A., Vargas, J.F., Morales, A., Ordonez, A.: Robustness of offline signature verification based on gray level features. IEEE Trans. Inf. Forensics Secur. **7**(3), 966–977 (2012)
16. Sharif, M., Khan, M.A., Faisal, M., Yasmin, M., Fernandes, S.L.: A framework for offline signature verification system. Best features selection approach. Pattern Recogn. Lett. **139**, 50–59 (2018). https://doi.org/10.1016/j.patrec.2018.01.021

17. Hafemann, L.G., Sabourin, R., Oliveira, L.S.: Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recogn. **70**, 163–176 (2017)
18. Hamadene, A., Chibani, Y.: One-class writer-independent off-line signature verification using feature dissimilarity thresholding. IEEE Trans. Inf. Forensics Secur. **11**(6), 1 (2016). https://doi.org/10.1109/TIFS.2016.2521611
19. Das, A., Suwanwiwat, H., Pal, U., Blumenstein, M.: ICFHR 2020 Competition on Short answer ASsessment and Thai Student SIGnature and Name COMponents Recognition and Verification (SASIGCOM 2020), ICFHR 2020, pp. 222–227 (2020)
20. Zhu, Y., Lai, S., Li, Z., Jin, L.: Point-to-set similarity based deep metric learning for offline signature verification. In: ICFHR 2020 (2020)
21. Hadjadj, I., Gattal, A., Djeddi, C., Ayad, M., Siddiqi, I., Abass, F.: Offline signature verification using textural descriptors. In: Morales, A., Fierrez, J., Sánchez, J.S., Ribeiro, B. (eds.) IbPRIA 2019. LNCS, vol. 11868, pp. 177–188. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31321-0_16
22. Das, A.: Face recognition in reduced Eigen-plane. In: Proceedings of the 2012 International Conference on Communications, Devices and Intelligent Systems, CODIS 2012, pp. 620–623. IEEE (2012)
23. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(10), 62–66 (1979). https://doi.org/10.1109/TSMC.1979.4310076
24. Impedovo, D., Pirlo, G.: Automatic signature verification: the state of the art. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **38**(5), 609–635 (2008)
25. Wirtz, B.: Technical evaluation of biometric systems. In: Chin, R., Pong, T.-C. (eds.) ACCV 1998. LNCS, vol. 1351, pp. 499–506. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-63930-6_160