Offline Signature Verification Using Convolution Siamese Network

Zi-Jian Xing; Fei Yin; Yi-Chao Wu; Cheng-Lin Liu
[1] National Laboratory of Pattern Recognition (NLPR), Institute of Automation Chinese Academy of Sciences, 95 Zhongguancun East Road, BEIJING, CHINA
[2] University of Chinese Academy of Sciences, BEIJING, CHINA

ABSTRACT

This paper presents an offline signature verification approach using convolutional Siamese neural network. Unlike the existing methods which consider feature extraction and metric learning as two independent stages, we adopt a deep-leaning based framework which combines the two stages together and can be trained end-to-end. The experimental results on two offline public databases (GPDSsynthetic and CEDAR) demonstrate the superiority of our method on the offline signature verification problem.

Keywords: Signature verification, Convolution Siamese Network, feature learning, metric learning

1. INTRODUCTION

Signature verification aims to verify the identity of a person based on his/her chosen signature, i.e. it classifies signature samples as “genuine”(created by the claimed individual) or “forgery”(created by the impostor) [1]. Because of non-invasive and familiar in daily life, signature verification has huge potential for financial security and information safety and has been widely used to verify a person's identity in legal, financial and administrative areas [2]. Therefore, it is a very important research topic in pattern recognition areas.

In offline signature scenario, the signature is captured after the completeness of writing process, by scanning the document containing the signature, and represented as a digital image [3]. Therefore the dynamic information such as position and velocity of the pen over time is absent, which makes the problem very challenging.

The essence of offline signature verification is very similar to other verification problems, such as face verification and person re-identification. The core problem is to find good representation and metric to evaluate the similarities between samples. During last few decades, many researchers have done some works on the problem. All these works include two separate stages: feature extraction and metric learning for classifying signature samples as “genuine” or “forgery”.

In the feature extraction stage, recent works explored a variety of different feature descriptors: such as contour feature [4] which encodes directional properties of signature contours and the length of regions enclosed inside letters; curvelet transform which uses the energy of the curvelet coefficient computed from the whole of the handwritten signature image [5]; surroundedness feature [6] which contains both shape and texture property of signature image; local features(Histogram of Oriented Gradients, Local Binary Patterns) [7] which are based on gradient information and neighboring information inside local regions of signature image in order to capture the signature's stable parts and alleviate the difficulty of global matching.

In the metric learning stage, there are two main approaches: writer-dependent which are mostly used and writer-independent. In the former case, a specialized metric model is learned for each individual during training phase, using the individual's either only the genuine signatures or genuine and forged signatures. In the testing phase, the learned metric model makes a classification on the signature claimed to be written by the particular individual as genuine or forgery. In the writer-independent scenario, there is only a single metric model for all users. In this case, a distinct set of users is used for training and testing. The model trained with difference vectors of a pair of signatures' feature of all users in the training set, learns how to metric the importance of different type of dissimilarities. The classifiers used for this stage, besides the most basic classifier (e.g. simple thresholding and nearest-neighbors), include neural networks [6], Hidden Markov Model [8], Support Vector Machines [5, 6, 7] and ensemble of these classifiers [9].

All the existing methods for solving offline signature verification problem consider feature extraction and metric learning as two independent stages. In other similar problems, such as face verification and person re-identification, researchers have combined feature extraction and metric learning in a unified framework [10, 11, 12, 13]. These works achieved better performance on corresponding databases compared to traditional works. Therefore, we adopt the Siamese neural network framework [14] which can assess the similarity of two signature images in the pair for solving offline signature verification problem, and combine the separate modules together that is feature extraction and metric learning in a unified framework. Compared with existing methods, our method has some advantages:

1. The system can learn a similarity metric from signature image directly. All layers in network are optimized by the same objective function, which are more effective than hand-crafted features in traditional methods.
2. The multi-channel filters learned in network can capture the global and local feature information simultaneously, which are more reasonable than the simple fusion strategies in traditional methods.
3. The structure of our system is flexible and more easily deployed in a real application.

The rest of this paper is organized as follows: Section 2 describes the system framework and method of our work. Section 3 presents the experimental results on two public databases (GPDSsynthetic [15] and CEDAR [16]). Section 4 concludes the paper with some suggested works in the future.

2. SYSTEM DESIGN

Unlike most of pattern recognition problems, the input of neural network is a sample and the output is a predicted label. For signature verification problem, the "sample → label" style of neural network is not suitable, we constructed Siamese neural network which includes two sub-CNN which working in a "sample pair → label" mode [11]. The framework of our method is shown in Figure 1. The input of the system is a pair of signature images and the output can be viewed as one of two classes (class I: both of two signature images are from the same writer and class II: two signature images are from different writers). The input images in the pair were first preprocessed in the same size and fed into the CNN in Siamese neural network to extract respective feature. After that, we used the metric method in our unified framework to calculate the distance of the respective extracted feature in the pair as the metric for the final classification. The whole system can be trained end-to-end. The following subsections we will introduce the preprocess, feature extraction and metric learning in details.

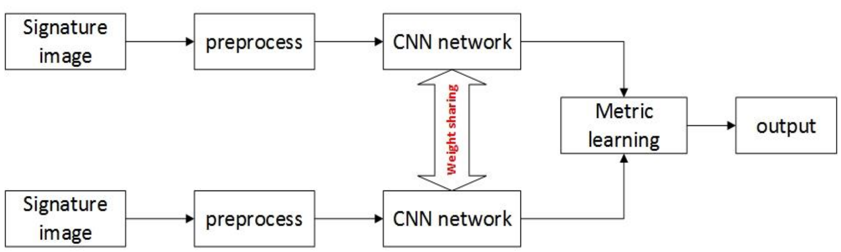

Figure 1. The diagram of the proposed signature verification system.

2.1 Preprocessing

Preprocessing is a very important step for solving any document analysis problem. The signature images from GPDSsynthetic and CEDAR databases are shown in Figure 2, from which we can see there are some differences between these signature images.

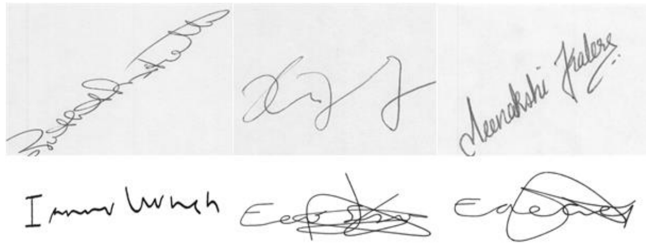

Figure 2. Some samples from the CEDAR (first row) and GPDSsynthetic (second row) signature databases.

The signature images in CEDAR database are skewed and not clean in background. Therefore, compared with the signature images in GPDSsynthetic database, we need more steps to preprocess the signature images in CEDAR database. For these images, firstly, we binarized the images using Otsu's method [17]; secondly, we used connected component analysis to clean grain noise, we chose a threshold T (T = 7, selected experimentally), and isolated particles of size less than T pixels were removed; thirdly, we used Hough Transform to detect and remove some lines which are not part of the signature images. After these steps, we got binary images of the original images. At next step we would process the original images according to the binary images. Specifically, we only remained the signature pixels unchanged where in corresponding binary image the pixels values were 1, and the other pixels values of the original images were set to be 255. Finally, we used the skew correction method introduced in [16] to rectify the image. After all these steps a sample of original images and preprocessed images is shown in Figure 3.

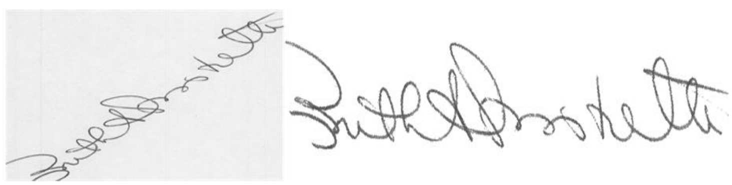

Figure 3. The left is the original signature image and the right is the signature image after skew correction.

After specific steps for preprocessing the signature images of CEDAR database, we further preprocessed images in both of two databases with the same strategy. We extracted the signature regions from images and found that the signature regions from both of two databases have a variable size. In GPDSsynthetic database, the sizes range from 80×24 pixels to 2792×1158 pixels and in CEDAR database the sizes range from 168×48 pixels to 1054×440 pixels. However, when training a convolutional neural network, all inputs are needed to be the same size. So we normalized the signature images by the method introduced in [2]. We first normalized the image to the largest image size, by padding the images with white background. Next, in terms of CEDAR database we centered the signatures in a canvas of size 1060×450 pixels, for GPDSsynthetic database we centered the signatures in a canvas of size 2800×1200 pixels, aligning the center of mass of the signature to the center of the image. After that, we resized all the signature images to a fixed size of 224×512 pixels, using bi-linear interpolation. Note that we performed rescaling without deformations. That is, when the original image had a different width-to-height ratio, we padded the less in the smaller dimension.

2.2 Feature extraction

The feature extraction module in our system is Siamese neural network which is composed of two convolutional neural network branches sharing the same parameters. So, we extracted features through CNN with the power of automatic feature learning. However, in offline signature verification problem, the famous network architectures such as AlexNet [18], VGG [19] and ResNet [20] are not suitable. So we designed a novel CNN which based on the property of these famous CNN architectures to fit signature verification problem. The architecture of our network is depicted in Figure 4.

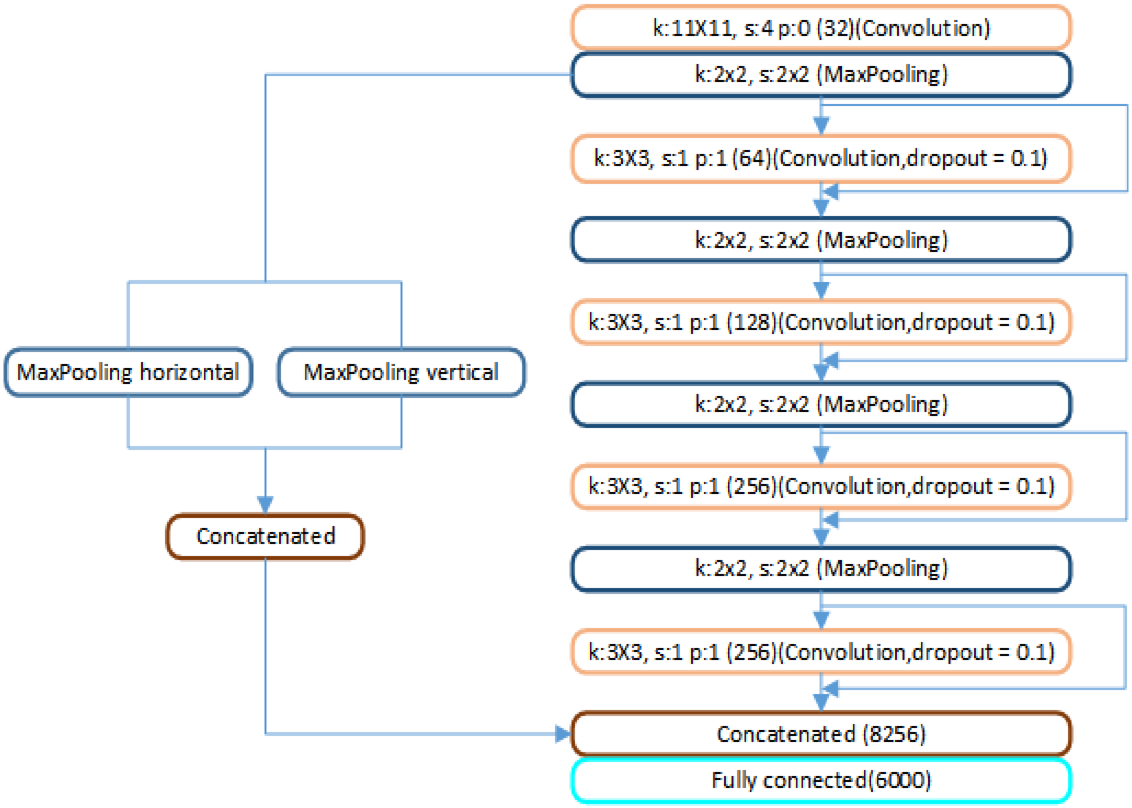

Figure 4. The CNN architecture of our system.

The first layer in our network is a convolutional layer with 11×11 kernel (including spatial normalization layer and rectifier layer) which makes the feature map compact. Then the feature maps produced by the first convolution and max pooling layer are processed by two paths. One path is composed of five convolutional layer with shortcut inspired from ResNet and four max-pooling layers. The second path is to make use of low level features. The details of the second path are as follows: assuming the feature map size after the first convolutional layer and max-pooling layer is $w \times h$, where w, h are the width and height of the feature map; Because the discriminative feature in the feature map may appear at horizontal or vertical direction, so we used max-pooling operation on the feature map on horizontal and vertical directions respectively; After the operation, the sizes of the output maps of second path are $h \times 1$ and $w \times 1$. Before last fully connected layer, the response maps from two paths are flattened then concatenated and fed to the following fully-connected layer. Finally, we got 6000 dims feature used for metric learning.

2.3 Metric learning

Assuming that the feature pair extracted from CNN learning feature module is $f_1, f_2 \in \Re^d$. Now we need to evaluate the similarity between a feature pair. Usually, Cosine or Euclidean distance is the way to measure similarity between a feature pair. The Euclidean and Cosine distances are formulated in Eq. (1) and Eq. (2), respectively.

$$d_E^2(f_1, f_2) = (f_1 - f_2)^T (f_1 - f_2) \tag{1}$$

$$d_{\cos}(f_1, f_2) = \frac{f_1 .* f_2}{\|f_1\|_2 \|f_2\|_2} \tag{2}$$

From these two equations, we can see that Euclidean distance is the summation of square differences in each dimension and Cosine distance is summation of correlations in each dimension. Inspired by the paper [13], we proposed a hybrid similarity layer which combines the property of Euclidean and Cosine distances. It is composed of two parts, the element-wise absolute difference and multiplication of a feature pair. The forward propagation of the hybrid similarity layer is formulated as follows:

$$f_{hybrid} = [f_{abs}, f_{multi}] \tag{3}$$

$$f_{abs} = |f_1 - f_2| \tag{4}$$

$$f_{multi} = f_1 .* f_2 \tag{5}$$

Where Eq. (4) means element-wise absolute difference of a feature pair which utilizes the absolute difference to replace the square difference for further simplifying the computation, compared with Euclidean distance, and Eq. (5) means the element-wise multiplication.

Then we added a linear layer to project the feature vector f_{hybrid} to a 2-dim vector $\left(\hat{p}_1, \hat{p}_2\right)$, which represents the predicted probability that the two signature images belong to the same identity. The whole metric learning module is shown as in Figure 5.

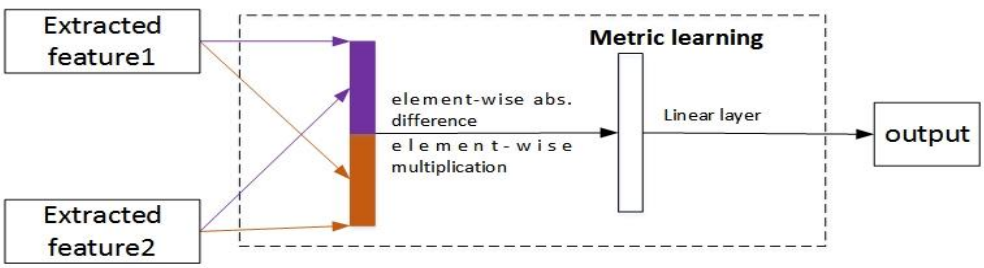

Figure 5. The metric learning scheme of our system.

We treated signature verification as binary-class classification problem and used cross-entropy loss.

$$Loss\left(p, \hat{p}\right) = -\left(p\log\left(\hat{p}_1\right) + (1-p)\log\left(\hat{p}_2\right)\right) = \sum_{i=1}^{2} -p_i \log\left(\hat{p}_i\right) \tag{6}$$

Where p is the target class(same/different) and $\hat{p}$ is the predicted probability. If the signature images in the pair belong to the same writer $p_1 = 1, p_2 = 0$; if not $p_1 = 0, p_2 = 1$.

3. EXPERIMENTAL RESULTS

We carried out experiments on two public databases, namely GPDSsynthetic and CEDAR signature databases separately. Because the most challenging job for offline signature verification problem is to discriminate between genuine signature and skilled forgery, so in this paper, we only considered skilled forgery. Descriptions of data and experimental setup are as follows.

3.1 Experimental setup

GPDSsynthetic Signature database is an offline signature database. It contains data from 4000 synthetic individuals: 24 genuine signatures for each individual, plus 30 forgeries of his/her signatures. All the signatures were generated with different modeled pens.

The database is large and we followed the method in paper [6], divided it equally, and made training and testing set. It means the signature images of randomly selected 2000 individuals are used for training and the rest are used for testing. Since for each individual there are 24 genuine and 30 forged signatures, we got $C_{24}^2 = 276$ genuine-genuine pairs of signatures and $C_{24}^{1} \times C_{30}^{1} = 720$ genuine-forged pairs of signatures for each individual. The number of genuine-forged pairs is significantly larger than that of genuine-genuine pairs. To avoid the classes imbalance problem, we applied undersampling on genuine-forged pairs to make the number of genuine-forged pairs are the same as genuine-genuine pairs. Totally, we had $2000 \times 2 \times 276 = 1104000$ signature pairs for training and testing separately.

CEDAR database is an offline signature database with much smaller size compared to GPDSsynthetic Signature database. It contains signatures of 55 volunteer signers belonging to versatile cultural backgrounds. There are 24 genuine signatures and skilled forged signatures for each individual. Totally, the database comprises of 1320 genuine and 1320 forged signatures of 55 signers.

Since the CEDAR database contains only 55 signature writers, following the divided method in paper [6], we selected 5 individuals randomly as testing data and used the signatures of remaining 50 individuals as training data. For each individual there are 24 genuine and 24 forged signatures, we got $C_{24}^{2} = 276$ genuine-genuine pairs and $C_{24}^{1} \times C_{24}^{1} = 576$ genuine-forged pairs of signatures for each individuals. Similarity, as discussed in GPDSsynthetic Signature database, finally we got $2 \times 50 \times 276 = 27600$ pairs as training samples and $2 \times 5 \times 276 = 2760$ pairs as testing samples. Because we only selected 5 individuals as testing set, to avoid the effect of selection, we repeated the experiment 10 times and reported the average performance on CEDAR database.

3.2 Model training

In our verification system, feature learning and metric learning are unified in one framework trained by end-to-end with SGD algorithm on torch7 [21] platform.

For GPDSsynthetic database, we trained the model with hyper-parameters listed in Table 1:

Table 1 The training hyper-parameters of GPDSsynthetic database.

Parameter	Value
Initial learning Rate	0.01
Learning Rate schedule	$LR \leftarrow LR * 0.1$ (every 3 epochs)
Weight Decay	$5e^{-4}$
Momentum	0.9
Batch Size	120
Epoch times	10

Table 2 The training hyper-parameters of CEDAR database.

Parameter	Value
Initial learning Rate	0.001
Learning Rate schedule	$LR \leftarrow LR * 0.1$ (every 10 epochs)
Weight Decay	$5e^{-4}$
Momentum	0.9
Batch Size	120
Epoch times	30

For CEDAR database, we used the model trained in GPDSsynthetic database for fine-tuning, similar to training strategy in GPDSsynthetic with small modification. The training hyper-parameters of CEDAR database are listed in Table 2:

3.3 Experimental result

There are two metrics for evaluating the offline signature verification system: False Rejection Rate (FRR) and False Acceptance Rate for skilled forgeries ($FAR_{skilled}$). The first one is fraction of the genuine signatures classified as forgery, while the second one is fraction of skilled forgery classified as genuine signatures. These two metrics heavily depend on suitable selection of threshold values. ROC analysis [22] is commonly consulted to get a suitable threshold. We drew an ROC curve for GPDSsynthetic and CEDAR databases to determine the equal error rate (EER) where FAR rate is the same as FRR rate. The ROC curve of GPDSsynthetic and CEDAR databases are shown in Figure 6, 7.

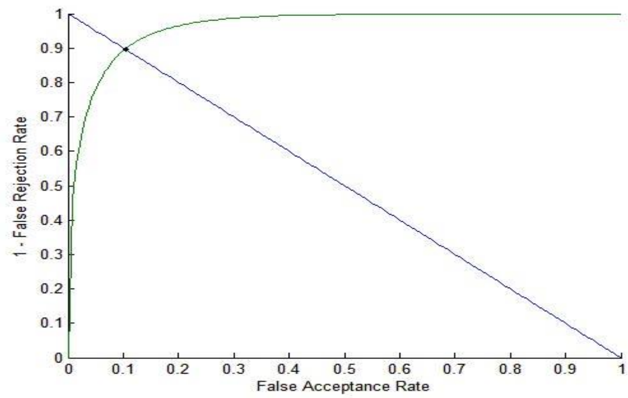

Figure 6. A typical ROC curve for GPDSsynthetic database.

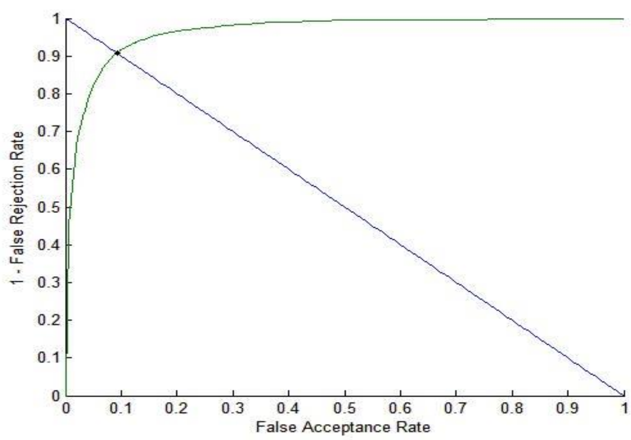

Figure 7. A typical ROC curve for CEDAR database.

We compared our system's result with other results on GPDS corpus and CEDAR database; the comparisons are shown in Table 3, 4.

From Table 3, 4 we can see that the proposed system outperforms all the compared methods on GPDS corpus and achieves comparable result with the state-of-the-art in CEDAR database. The GPDSsynthetic is the biggest public offline signature database; the performance on this database is more persuasive which demonstrate the superiority of our system. Our proposed method also has advantage in its workability as writer independence system. However, most of the systems in the Table 3, 4 (expect surroundness features by Kumar et al. [6]) work depending on the writer, which is not economical if a new writer is introduced these systems have to be updated. Finally, our proposed method combines feature learning and metric learning in a unified framework which is simple and elegant with good performance. We also evaluated different metric methods. The result in Table 5 shows the hybrid similarity learning is better than simple Cosine and Euclidean distances which is consistent with the theory inference introduced in metric learning subsection.

To evaluate the effects of undersampling, we did experiments with and without undersampling on training sets and testing sets. These results are listed in Table 6. We can see that understanding influences the performance only slightly. This demonstrates the robustness of our system.

Table 3 Comparison between proposed and other published method on CEDAR database.

Systems	#Signer	Accuracy	EER
Chen et al. [23]	55	83.60	16.30
Kumar et al.[24]	55	88.41	11.59
Kumar et al.[6]	55	**91.67**	**08.33**
Ours	55	91.50	08.50

Table 4 Comparison between proposed and other published method on GPDS signature corpus.

Systems	#Signer	Accuracy	EER
Ferrer et al.[25]	160	86.65	13.35
Vargas et al.[26]	160	87.67	12.23
Kumar et al.[6]	300	86.24	13.76
Soleim et al.[27]	4000	86.70	13.30
Ours	4000	**89.63**	**10.37**

Table 5 The EER results of our system with different metric learning method on GPDSsynthetic database.

Metric Method	EER
Cosine Distance	20.46
Euclidean Distance	12.30
Hybrid Similarity learning	**10.37**

Table 6 The EER results of our system with different sampling on GPDSsynthetic database.

sampling on train	sampling on test	EER
No	No	10.65
Yes	No	10.87
Yes	Yes	**10.37**

4. CONCLUSION

This paper proposed an offline signature verification approach for combining feature extraction and metric learning in a unified framework using convolutional Siamese neural network. We designed CNN module and metric learning module suitably for offline signature verification problem. The performances on two public databases GPDS and CEDAR demonstrate the superiority of the proposed approach. In the future, we will design better network architecture which extracts more discriminative feature and make the whole system more practical.

REFERENCES

[1] A.K. Jain, A. Ross and S. Prabhakar, “An introduction to biometric recognition,” IEEE Trans. CSVT, 14(1), 4–20 (2004)

[2] L.G. Hafemann, R. Sabourin and L.S. Oliveira, “Writer independent feature learning for offline signature verification using deep convolutional neural networks,” Proc. IJCNN, 2576–2583 (2016)

[3] D. Impedovo and G. Pirlo, “Automatic signature verification:The state of the art,” IEEE Trans. on Systems, Man, andCybernetics, Part C, 38(5), 609–635 (2008)

[4] A. Gilperez, F. Alonso-Fernandez, S. Pecharroman, J. Fierrez, and J. Ortega-Garcia, “Off-line signature verification using contour features,” Proc. ICFHR (2008)

[5] Y. Guerbai, Y. Chibani, and B. Hadjadji, “The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters,” Pattern Recognition, 48(1), 103–113 (2015)

[6] R. Kumar, J. Sharma, and B. Chanda, “Writer-independent off-line signature verification using surroundedness feature,” Pattern Recognition Letters, 33(3),301–308 (2012)

[7] M. B. Yilmaz, “Offline signature verification with user-based and global classifiers of local features,” PhD thesis, Sabancı University (2015)

[8] L. S. Oliveira, E. Justino, C. Freitas, and R. Sabourin, “The graphology applied to signature verification,” Proc. In 12th Conferenceof the International Graphonomics Society, 286–290 (2005)

[9] J. Hu and Y. Chen, “Offline signature verification using real adaboost classifier combination of pseudo-dynamic features,” Proc. ICDAR, 1345–1349 (2013)

[10] J. Hu, J. Lu, and Y.-P. Tan, “Discriminative deep metric learning for face verification in the wild,” Proc. CVPR, 1875–1882 (2014)

[11] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” Proc. ICPR, 34-39 (2014)

[12] Z. Zheng, L. Zheng, and Y. Yang, “A discriminatively learned cnn embedding for person re-identification,” arXiv preprint arXiv:1611.05666 (2016)

[13] J. Zhu, H. Zeng, S. Liao, Z. Lei, C. Cai, and L. Zheng, “Deep hybrid similarity learning for person re-identification,” arXiv preprint arXiv:1702.04858 (2017)

[14] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a “siamese” time delay neural network,” Proc. NIPS, 737–744 (1994)

[15] M.A. Ferrer, M. Diaz-Cabrera, and A. Morales, “Static signature synthesis: A neuromotor inspired approach for biometrics,” IEEE Trans. PAMI, 37(3),667–680 (2015)

[16] M.K. Kalera, S. Srihari, and A. Xu, “Offline signature verification and identification using distance statistics,” International Journal of Pattern Recognition and Artificial Intelligence, 18(07), 1339–1360 (2004)

[17] N. Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. on Systems, Man, and Cybernetics, 9(1), 62–66 (1979)

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Proc. NIPS, 1097–1105 (2012)

[19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556 (2014)

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. CVPR, 770–778 (2016)

[21] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning,” In BigLearn, NIPS Workshop (2011)

[22] T. Fawcett, “An introduction to roc analysis,” Pattern Recognition Letters, 27(8), 861–874 (2006)

[23] S. Chen and S. Srihari, “A new off-line signature verification method based on graph,” Proc. ICPR, 869–872 (2006)

[24] R. Kumar, L. Kundu, B. Chanda, and J. Sharma, “A writerindependent off-line signature verification system based on signature morphology,” Proc. Int.Conf.on Intelligent Interactive Technologies and Multimedia on ACM, 261–265 (2010)

[25] M.A. Ferrer, J.B. Alonso, and C.M. Travieso, “Offline geometric parameters for automatic signature verification using fixed-point arithmetic,” IEEE Trans. PAMI, 27(6), 993–997 (2005)

[26] F. Vargas, M. Ferrer, C. Travieso, and J. Alonso, “Off-line handwritten signature gpds-960 corpus,” Proc. ICDAR, 764–768 (2007)

[27] A. Soleimani, B.N. Araabi, and K. Fouladi, “Deep multitask metric learning for offline signature verification,” Pattern Recognition Letters, 80, 84–90 (2016)