Proceedings of the Federated Conference on Computer Science and Information Systems pp. 287–290
DOI: 10.15439/2019F28
ISSN 2300-5963 ACSIS, Vol. 18

Signature analysis system using a convolutional neural network

Alicja Winnicka, Karolina Kęsik
Dawid Połap
Institute of Mathematics
Silesian University of Technology
Kaszubska 23, 44-100 Gliwice, Poland
Email: Alicja.Lidia.Winnicka@gmail.com, Karola.Ksk@gmail.com
Dawid.Polap@polsl.pl

***Abstract*—Identity verification using biometric methods has been used for many years. A special case is a handwritten signature made on a digital device or piece of paper. For the digital analysis and verification of its authenticity, special methods are needed. Unfortunately, this is a rather complicated task that quite often requires complex processing techniques. In this paper, we propose a system of signatures verification consisting of two stages. In the first one, a signature pattern is created. Thanks to this, the first attempt to verify identity takes place. In the case of approval, the second stage is followed by the processing of a graphic sample containing a signature by the convolutional neural network. The proposed technique has been described, tested and discussed due to its practical use.**

I. Introduction

USING signature, we can confirm our identity. This is particularly important in the case of signing contracts or receiving parcels from couriers. In each of these situations, we confirm something with signature. Of course, such a signature may differ each time we use it. The reason is his elaboration, which means that the more we sign, the more stable it will be. Lawyers, politicians or office workers who often sign on different documents will have a stable and permanent signature very quickly. Consequently, such a signature is very important as an element of our identity.

Unfortunately, there are situations where such a signature is forged for fraud purposes. To prevent this, it is worth having a signature, which minimizes the possibility of counterfeiting. This is possible thanks to a much smoother writing process, pen pressure, or runtime. All these features are often used in the authentication process. In practice, such verification is not an easy thing, which is why the complicated methods of artificial intelligence are often used.

An important element of verification process is the classifier, which makes decisions with a certain probability. The most popular are artificial neural networks which are inspired by the mechanisms occurring in the brain. An important element of scientific research is the improvement and design of new solutions that may later be used in biometrics [1]–[6]. One of the last achievements in this field are papers on the interpretation of the signature in numerical form and the use of so-processed samples in the training process [7], [8]. Interesting approach was presented in [9], where the authors described a technique that uses a single record with a signature.

The cited works show great effectiveness, however, it is worth paying attention to the mechanism of such software. Quite a frequent mechanism is processing not on the device, but in the computing cloud [10]. Another thing is the front-end of the software and the ways in which the application is displayed to the user [11], [12]. Conducted research is not only related to the signature but also to other elements that can confirm our identity, an example of which are the fingerprint and iris of the eye [13]–[15].

In this paper, we present a system for signature verification based on two stages based on image processing and convolutional neural networks.

II. Signature processing

Each graphic sample containing a signature should be processed. The main reason is the inclination of the signature relative to the straight line. In order to analyze or to compare two signatures, both samples should be arranged on the same straight line without any inclination.

The slope of the signature can be calculated using the linear approximation for the given set in the discrete form. The signature is a graphic file that should be saved in numerical form. The first step is to binarize the image. Each pixel in the image is described in the RGB color system (*Red-Green-Blue*), so the binarization will mean replacing all colors with white or black one using the following equation

$$\begin{cases} \dfrac{\sum\limits_{i\in\{R,G,B\}} i(p)}{3} < \left[\dfrac{255}{2}\right] \text{ then } R(p)=G(p)=B(p)=0 \\ \dfrac{\sum\limits_{i\in\{R,G,B\}} i(p)}{3} > \left[\dfrac{255}{2}\right] \text{ then } R(p)=G(p)=B(p)=255 \end{cases} \tag{1}$$

where functions $R(\cdot)$, $G(\cdot)$ and $B(\cdot)$ are a color component of pixel p. The value on the right of the inequality means the total value of the center of the color range in the RGB model.

As already mentioned, the image after the binarization process consists of two colors – black and white, where

the black pixels represent the signature. Each pixel can be interpreted as a coordinate (x, y). Taking these points, we have a set of points on the Cartesian plane given discreetly. On its basis, it is possible to calculate the slope of the signature using the linear function equation in the following form

$$f(x) = y = a_0 + a_1 x \Rightarrow a_0 = \tan(\alpha) \Rightarrow \alpha = \arctan(a_0). \tag{2}$$

It is easy to see that having a coefficient a_0, the slope can be calculated. However, the value of a_0 must be calculated. Suppose that the set of points has n elements, then we are looking for such a function $f(x)$, for which the following condition will occur

$$f(x_i) = y_i, \quad i = 0, \dots n-1. \tag{3}$$

Assume that for a set of points $\{(x_i, y_i)\}$ where $i \in \{0, n-1\}$, a function $S(a_0, a_1)$ will be presented as

$$S(a_0, a_1) = \sum_{i=0}^{n-1} (y_i - a_0 - a_1 x_i)^2 \tag{4}$$

Therefore, the system of normal equations has the following form

$$\frac{\partial S(a_0, a_1)}{\partial a_0} = \sum_{i=0}^{n-1} (y_i - a_0 - a_1 x_i)(-1) = 0 \tag{5}$$

$$\frac{\partial S(a_0, a_1)}{\partial a_1} = \sum_{i=0}^{n-1} (y_i - a_0 - a_1 x_i)(-x_i) = 0 \tag{6}$$

By grouping the above two equations, we get

$$a_0 n + a_1 \sum_{i=0}^{n-1} x_i = \sum_{i=0}^{n-1} y_i \tag{7}$$

$$a_0 \sum_{i=0}^{n-1} x_i + a_1 \sum_{i=0}^{n-1} + a_1 \sum_{i=0}^{n-1} x_i^2 = \sum_{i=0}^{n-1} x_i y_i. \tag{8}$$

The above system of equations is linear, so it can be saved in a simpler form as

$$X \cdot A = Y, \tag{9}$$

where A, X, Y are a matrices defined as

$$A = \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}, X = \begin{pmatrix} \sum_{i=0}^{n-1} 1 & \sum_{i=0}^{n-1} x_i \\ \sum_{i=0}^{n-1} x_i & \sum_{i=0}^{n-1} x_i^2 \end{pmatrix}, Y = \begin{pmatrix} \sum_{i=0}^{n-1} y_i \\ \sum_{i=0}^{n-1} x_i y_i \end{pmatrix} \tag{10}$$

Finally, searched coefficients can be obtained by

$$A = X^{-1} \cdot Y. \tag{11}$$

In this way, we obtain the coefficients of the approximated linear function, and thus using the equation (2), it is possible to find the slope of the signature for which the image with the signature should be rotated.

Having a processed and rotated image, we put together several signatures belonging to the same person. The imposition of images consists in creating matrix with a dimension adequate to the samples (if the samples are of different sizes, they should be normalized to the same one). This matrix should be filled with 0 (which is understood as a white pixel). Then, for each image, pixels are checked. If there is a black pixel at a given position in the image, then the value in this matrix is increased by 1.

Such a matrix allows us to create patterns. The higher the value of a given matrix element, the more often the pixel appears and can be treated as a feature.

The smallest values should be replaced with zeros. The selection of this value depends on the number of samples used in the process of its creation. As part of the experiments to be carried out, the optimal value was determined as $\frac{n}{2}$ or $\frac{n-1}{2}$.

Such a matrix can be applied to a new, processed signature. In this case, we calculate the number of black pixels of the signature that are on the positions in the matrix (where the elements are different from zero). This allows us to calculate the percentage coverage of features.

III. Convolutional neural network

The previous stage allowed the creation of a technique that gives the percentage quality of coverage of the main features. Unfortunately, it does not allow the verification process itself. For this purpose, we use Convolutional Neural Network (CNN) [16], which are a mathematical model of action having place in the primary cortex. These structure take the image at the entrance, and at the output they return the class to which the input belongs with a certain probability. Structure construction can be described using three layers. The first two layers are used to feature extraction and are called convolutional and pooling. The first type processes the image using a certain filter ω defined as a matrix and a step S through which this matrix will be moved. The next layer is called pooling which reduces the size of the image using the selected function. If the image is to be reduced to t times, a matrix of size $t \times t$ is created, in which only one pixel is selected and this matrix is shifted in the image, resulting in a reduced image. Next, a classic neural network is created that forms the last layer of the network.

The structure itself is quite simple in its model, however, it must be added that the layers are connected to each other thanks to synapses burdened with weights. At the initial stage of creating networks, they are generated in a random way. Using the training algorithm, they are modified due to input data. The most commonly used algorithm is Adaptive Moment Estimation (*Adam*).The algorithm consists in calculating the mean values m of the gradient and the second momentum (variance) v in each iteration t. Let us assume that $w^(t)$ will be understood as a parameters and $L^{(t)}(\cdot)$ as a loss function. Formally, the equations for these values are as follows

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{12}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{13}$$

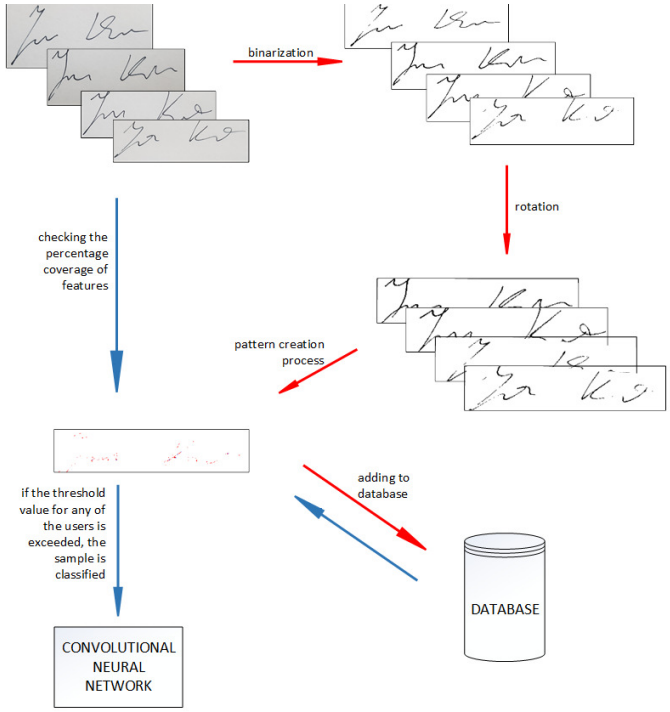

Figure 1: Red arrows indicate the preparation process and blue arrows verification process.

Layer	Output Shape
Convolutional	(None,148,148,32)
Activation	(None,148,148,32)
MaxPooling	(None,74,74,32)
Convolutional	(None,72,72,32)
Activation	(None,72,72,32)
MaxPooling	(None,36,36,32)
Convolutional	(None,34,34,64)
Activation	(None,34,34,64)
MaxPooling	(None,17,17,64)
Flatten	(None,18496)
Dense	(None,64)
Activation	(None,64)
Dropout	(None,64)
Dense	(None,2)
Activation	(None,2)

Table I: Convolutional neural network architecture.

where β_1 and β_2 are decay coefficients, whcich values are close to 1. The correction for a given moment is defined as

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \tag{14}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}, \tag{15}$$

which are used to update the value

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t}+\epsilon}\hat{m}_t, \tag{16}$$

where ϵ is a constant, small value used to prevent dividing by 0 and η is the learning rate.

IV. Signature Verification Model

The proposed system consists of image processing or normalization of the sample, and then using it in two stages. The first one is to create a matrix or use it to check the percentage of coverage of features. If the obtained value exceeds the threshold value, then this sample is classified by the convolutional neural network. The graphical illustration of the model is shown in Fig. 1. This action reduces the number of operations performed by possibly rejecting the sample due to the feature matrix.

Assuming that the system should enable the identity verification of several people, the signature is analyzed in relation to all matrices in the database. If for any of them, the percentage threshold is exceeded, it is classified by the network. In case the matrix and network return different results, the system will not be able to clearly identify the owner of the sample.

V. Experiments

For testing purposes, a small signature database of two people was created, consisting of 50 samples (25 per person). In addition, 20 samples (10 for each person) of fake signatures were created (tried to imitate other's signatures). In the classifier learning process, all samples were normalized to the dimension 150×150 pixels. Used architecture of CNN is presented in Tab. I.

The features matrix was tested for different values, and the best efficiency (in an empirical way) was obtained for a value equal to $\frac{n-1}{2}$. Classifier was trained 10 epochs using 70% : 30% of samples (training to validation number of samples). The history of training is shown in Fig. 2

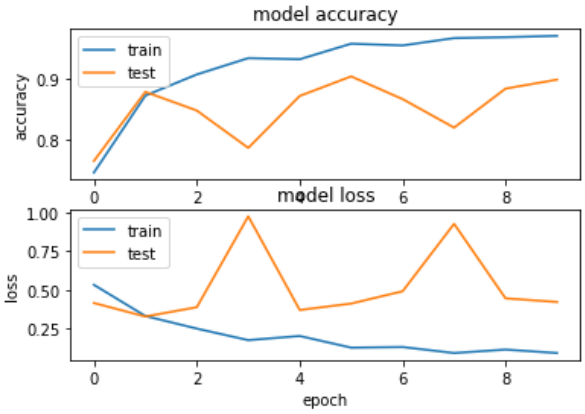

Figure 2: Graphs of training history using 50 samples (35:15 training : validation).

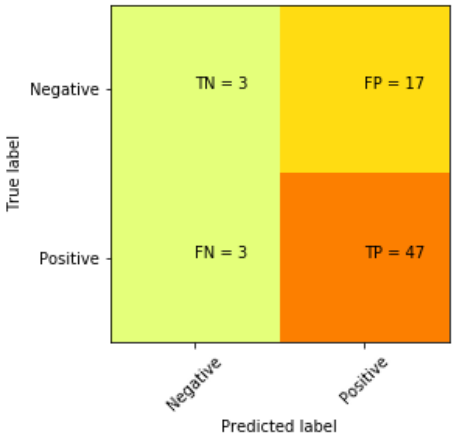

Figure 3: Confusion matrix for average results contained from 10 attemps.

The effectiveness of the classification for two people was achieved at 94%, which is a very good result considering the number of samples in the training process. It is worth noting that after training the classifier, fake samples were used in order to verify the operation of the proposed system. 11 signatures were rejected at the level of the features matrix and the remaining ones were classified by the classifier. The average probability of belonging to these people was in the range $33-90\%$, what is a good result. The optimistic approach is the result of the fact that the original samples were classified above 83%, what allows to reject counterfeit signatures except for selected images. The network was trained ten times in order to obtain an average classification value, which was achieved at 91%, the average results of classification was presented in confusion matrix in Fig. 3.

VI. Conclusion

The described method of identity verification based on the signature indicates high efficiency. The experiments were carried out on the basis of a total sum of samples equal to 50, which is quite a small amount. It is worth noting that adding verification using the matrix of features allowed to reduce the operations performed using CNN because it rejected over half of suspicious samples before the verification stage. This solution indicates the possibility of obtaining better efficiency using more extensive techniques for creating matrix pattern using other features.

Acknowledgments

Authors acknowledge contribution to this project to the Diamond Grant No. 0080/DIA/2016/45 funded by the Polish Ministry of Science and Higher Education.

References

[1] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[2] V. Nourani, S. Mousavi, D. Dabrowska, and F. Sadikoglu, "Conjunction of radial basis function interpolator and artificial intelligence models for time-space modeling of contaminant transport in porous media," *Journal of hydrology*, vol. 548, pp. 569–587, 2017.

[3] D. Dąbrowska, R. Kucharski, and A. J. Witkowski, "The representativity index of a simple monitoring network with regular theoretical shapes and its practical application for the existing groundwater monitoring network of the tychy-urbanowice landfills, poland," *Environmental Earth Sciences*, vol. 75, no. 9, p. 749, 2016.

[4] A. Venčkauskas, R. Damaševičius, R. Marcinkevičius, and A. Karpavičius, "Problems of authorship identification of the national language electronic discourse," in *International Conference on Information and Software Technologies*. Springer, 2015, pp. 415–432.

[5] R. Damaševičius, R. Maskeliūnas, E. Kazanavičius, and M. Woźniak, "Combining cryptography with eeg biometrics," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[6] R. Damaševičius, R. Maskeliūnas, A. Venčkauskas, and M. Woźniak, "Smartphone user identity verification using gait characteristics," *Symmetry*, vol. 8, no. 10, p. 100, 2016.

[7] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Exploring recurrent neural networks for on-line handwritten signature biometrics," *IEEE Access*, vol. 6, no. 5128-5138, pp. 1–7, 2018.

[8] M. Elhoseny, A. Nabil, A. E. Hassanien, and D. Oliva, "Hybrid rough neural network model for signature recognition," in *Advances in Soft Computing and Machine Learning in Image Processing*. Springer, 2018, pp. 295–318.

[9] M. Diaz, A. Fischer, M. A. Ferrer, and R. Plamondon, "Dynamic signature verification system based on one real signature," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 228–239, 2018.

[10] G. L. Masala, P. Ruiu, and E. Grosso, "Biometric authentication and data security in cloud computing," in *Computer and Network Security Essentials*. Springer, 2018, pp. 337–353.

[11] Z. Sroczyński, "Actiontracking for multi-platform mobile applications," in *Computer Science On-line Conference*. Springer, 2017, pp. 339–348.

[12] A. Bier and Z. Sroczynski, "Towards semantic search for mathematical notation," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2018, pp. 465–469.

[13] N. Merhav, "Ensemble performance of biometric authentication systems based on secret key generation," *IEEE Transactions on Information Theory*, 2018.

[14] K. Zhou and J. Ren, "Passbio: Privacy-preserving user-centric biometric authentication," *IEEE Transactions on Information Forensics and Security*, 2018.

[15] P. Gupta and P. Gupta, "Multibiometric authentication system using slap fingerprints, palm dorsal vein, and hand geometry," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9777–9784, 2018.

[16] H. Huang, C. Wang, and B. Dong, "Nostalgic adam: Weighing more of the past gradients when designing the adaptive learning rate," *arXiv preprint arXiv:1805.07557*, 2018.