Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 13, No. 3, August 2022), Pages: 31-49
www.jacr.iausari.ac.ir

Automatic offline identification of signature author based on deep learning and its evaluation in noisy conditions

Davood Keykhosravi [1], Seyed Naser Razavi ✉[1], Kambiz Majidzadeh [1], Amin Babazadeh Sangar [1]

1) Department of IT and Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, 57169-63896, Iran

D.keykhosravi@iaurmia.ac.ir; N.razavi@tabrizu.ac.ir; K.Majidzadeh@iaurmia.ac.ir; Bsamin2@liveutm.onmicrosoft.com

Abstract

Signature identification plays an important role in many areas such as banking, administrative and judicial systems. For this purpose, in this paper, an automatic intelligent framework is developed by combining a deep pre-trained network with a recurrent neural network. The results of the proposed model were evaluated on several valid datasets and collected datasets. Since there was no suitable Persian signature dataset, we collected a Persian signature dataset based on US ASTM guidelines and standards, which can be very effective and profound for deep approaches. Due to the very promising results of the proposed model in comparison with recent studies and conventional methods, to evaluate the resistance of the proposed model to different noises, we added Gaussian Noise, Salt and Pepper Noise, Speckle Noise, and Local var Noise in different SNRs to the raw data. The results show that the proposed model can still be resistant to a wide range of SNRs; So at 15 dB, the accuracy of the proposed method is still above 90%.

Keywords: *Automatic Identification of the Writer of the Signature, Pre-trained Network, Feature Learning, Convolutional Neural Network (CNN).*

1. Introduction

Biometrics refers to the recognition or confirmation of a person's identity by measuring his physiological or behavioral characteristics. Thus biometrics is a technology suite. Biometric systems have two very important features that enhance their reliability. First is that the person who wants to authenticate must be personally present during the process and secondly, identification does not require the person to keep or remind information or bring something with it [1]. Signature analysis is used as a popular, cost-effective authentication method and is preferred among various biometrics as it is the widely accepted way to identify an individual. A signature is a special case of the handwritten note which results in a complex process of writing a series, based on the motion curve muscle movements, associated with the idea of signing, which includes special characters with special art of writing [2]. The identification of humans by their signature can now be an identifiable factor in the process of recognition [3]. In the past, handwriting signatures were traditionally verified and recognized by signature recognition experts [4]. However, the verification and recognition of handwritten signatures in the traditional way is time-consuming, tedious, and unprincipled; the traditional method also reduces the accuracy of the identification and verification of signatures [5]. Today, with the advancement of technology in many modern countries, the verification and recognition of handwritten signatures are carried out automatically. Automatic verification and identification of handwritten signatures can be performed in two different modes, online and offline [6, 7]. In the offline signature mode, people put their signatures on paper, and after scanning these documents, the relevant signatures are entered into the system as an image that is finally identified and approved by the system [8]. While in online signature mode, the signatures are entered directly on the computer monitor using tablets and are immediately verified by the system [8]. However, the accuracy of offline signature verification is more difficult than the online signature mode due to the dynamic information of the signature images. In general, the systems for the recognition of signatures are divided into two categories: Signature Verification (SV) and signature recognition (SR) [8, 9]. Thus, in the case of SV, the attributes of the test signature are compared with the attributes associated with the reference signatures whose identities are known [10]. While in SR, the system recognizes the signature of each person and predicts their identity [11]. Genuine and forged signatures are therefore tested by the SV system. Whereas in SR systems, only genuine signatures are distinguished from each other [11-13]. SV methods can be divided into Writer-Dependent (WD) and Writer-Independent (WI) modes. In WD mode, a separate classifier is created for each signatory, and the classifier is trained from the start for each new signature. But for WI mode, the system can test the new signature without retraining [11-14]. Different methods and studies have been used to identify signatures authors. Figure 1 represents the scope of this lecture.

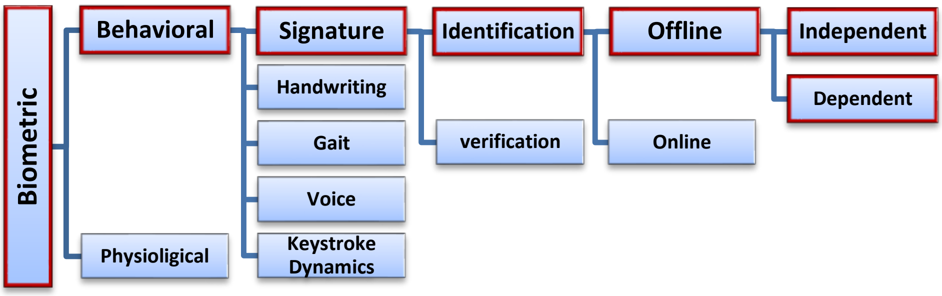


Figure 1. Specific Scope of Lecture

The majority of the above-mentioned studies are divided into three general categories: the first is Geometric methods [15], the second is statistical methods [16], and the third is machine learning-based methods [17]. Geometric methods such as the Hidden Markov Model (HMM) [18], Euclidean Distance (ED) [19], Mahalanobis Distance (MD) [20], and Discrete Radon Transform (DRT) [21] were used in the last research. Co-event Matrix [22], Texture and Binary Pattern (TBP) [23], and Gradient Structural Concavity (GSC) [24] were used as statistical methods to identify signatures authors, and also machine learning methods used in recent investigations were Multi-Layer Perceptron (MLP) [25], Support Vector Machine (SVM) [2], K-Nearest Neighbor (KNN) [27], and Deep Learning (DL) methods such as Convolutional Neural Networks (CNNs) [17]. Nowadays DL methods with higher accuracy and complexity are replaced with previous methods that can-do features engineering automatically. Recent research on the automatic recognition and verification of handwritten signatures in offline mode will be discussed below. Van et al. [28] used Viterbi Path Along (VPA) for signature verification, and

HMM was employed to classify signatures by extracting a likelihood score. Nanni et al. [29] presented the Linear Programming Descriptor classifier (LPD) in their study to extract various features of the signatures on which, the discrete 1-D Wavelet Transform (WT) and the Discrete Cosine Transform (DCT) were performed on. Vargas et al. [30] study is based on gray level information using the Co-occurrence Matrix to represent texture features and SVM used for classification. Karouni et al. [31] performed neural networks approach using a set of geometric features such as Skewness, Eccentricity, Area, and Center of gravity for validation and verification. Khalajzadeh et al. [32] used CNNs and MLP to verify Persian signatures which focused on feature extraction where a dataset with 176 genuine Persian signatures from 22 writers was examined. Verma et al. [33] studied Angle, Pixel Density, and combined features to classify with Neural Networks (NNs), where False Accept Rate (FAR) and False Reject Rate (FRR) were improved. Samonte et al. [34] used Deep Convolutional Neural Networks (DCNNs) for handwritten signature classification in two genuine and forged classes and four datasets. Okawa et al. [35] studied signature feature extraction and encoding in Two methods: Bag-of-Visual Words (BoVW) and a Vector of Locally Aggregated Descriptors (VLAD). Bouamra et al. [36] designed a classifier based on One-Class SVM (OC-SVM) to classify the GPDS960 dataset. Ghosh et al. [37] used Recurrent Neural Networks (RNNs) to verify and recognize offline handwritten signatures. [38] used DCNN to automatically detect and verify offline handwritten signatures. In their DNN architecture, the researchers used a combination of the KNN algorithm with CNN to classify feature vectors based on the neighborhood rule. The right-to-left signatures are not normally similar to the left-to-right signatures and do not have any text in their drawn style. This difference is clearly shown in Figure 2. Because there are no comprehensive datasets for right-to-left signature data available, recent studies have used the UTsig dataset for writer identification based on right-to-left signatures, which has few samples and is thus unsuitable for DL approaches due to sampling limitations. To address the existing problems in this paper:

• Collecting a new comprehensive dataset of Persian right-to-left signatures containing 6739 samples from 85 left-handed and right-handed people of different ages, genders, and educational levels. To collect this dataset, various environmental conditions such as light level, sitting position, standing position, etc. have been considered.

• Presenting a developed model for writer identification under various experimental conditions that is independent of right-handed or left-handed people.

• Designing a developed DCNN model based on a pre-trained network (Inception_V3) with a recurrent neural network (LSTM).

• Evaluation of the proposed model under different noises (Gaussian noise, salt and pepper noise, point noise, and Local var noise) to consider the effect of environmental conditions on documents.

The rest of the paper is organized as follows: The second section describes the methodology of this research. The third section presents a proposed method. The fourth section presents the simulation results based on the proposed network for automatic offline identification of the signature author and, finally, the fifth section deals with the conclusion.

Figure 2. Comparison between right-to-left and left-to-right signatures.

2. METHODOLOGY

In this section, first, the dataset prepared based on the right-to-left Persian signatures is described, then the mathematical theory of CNN and LSTM is examined.

2.1. Dataset collection

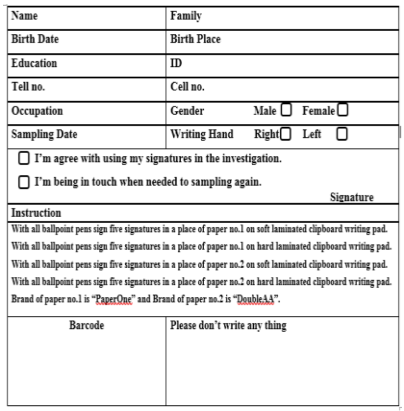
To overcome the limitations of previous datasets, we collected a standard right-to-left signature dataset called DANASIG. In addition, the UTSig, ICDAR-Chinese, ICDAR-Dutch, and MCYT datasets were used in this study to evaluate the proposed algorithm's performance. The datasets used in this study have been extensively used in previous studies and are among the most popular and widely used datasets. The proposed dataset, as well as other datasets used in this study, are described briefly below.

2.1.1. DANASIG

To obtain a standard data set based on the right-to-left Persian handwritten signature, 6739 original signatures were collected from 85 candidates by the ASTM standard. The ASTM standard form is shown in Figure 3. Accordingly, of the 85 volunteers participating in this test, 57 were male and 28 were female with an average age of 20 to 80 years. Also, 11 of these volunteers were left-handed and 74 were right-handed. The samples collected include 80 signatures of each participant on sheet A4. As a result, each participant registered their signatures on four sheets (each sheet contains 20 signatures). There will be 6800 (85 volunteers × 80 signatures from each person) signatures for the dataset collected, of which 61 samples have been left out of the processing process due to corrupt signatures and only 6739 samples have been processed. An example of the signatures collected for one of the study subjects is shown in Figure 4. Various types of pens and paper have been used to consider the impact of the signing conditions. The different types of pens and papers used in this study are shown in Table 1. According to Table 1, four different brands of pens and two different brands of paper with different specifications have been used in this study, so that the data set collected is not based on just one type of paper and pen. In the collection of signature samples, we also considered different environmental conditions. Different environmental conditions are: signing in a standing position, signing in a sitting position, signing in a semi-standing position, and signing in different lighting conditions. As mentioned earlier, the conditions considered for this study have not been met in the collection of any of the previous datasets, which indicates the importance of the dataset being collected. All data set samples were scanned with a resolution of 300 dpi in RGB mode. Also, the size of each scanned signature sample is 591 × 472 pixels. The collection data collected under the name of "DANASIG" on the GitHub platform is available to all researchers .(***https://github.com/davoodsig/Persian-signature-DANASIG***)

Table 1. Various papers and pens are intended for the collection of the proposed dataset.

Equipment	Brand	Specification
Paper	Paper One	A4 – Weight 80 g/m2 – Brightness 99% - Thickness 110 µm
	Double A	A4 – Weight 80 g/m2 – Brightness 102.5% - Thickness 106.5 µm
Pen	Schneider	Type Ballpoint – Color Blue– Width 0.4mm
		Type Ballpoint – Color Black – Width 0.4mm
	Faber-Castell	Type Ballpoint – Color Blue– Width 0.7mm
		Type Ballpoint – Color Black – Width 0.7mm
	Panter	Type Ballpoint – Color Blue– Width 1.0mm
		Type Ballpoint – Color Black – Width 1.0mm
	Bic	Type Ballpoint – Color Blue– Width 1.2mm
		Type Ballpoint – Color Black – Width 1.2mm



Name	Family
Birth Date	Birth Place
Education	ID
Tell no.	Cell no.
Occupation	Gender Male ☐ Female ☐
Sampling Date	Writing Hand Right ☐ Left ☐

☐ I'm agree with using my signatures in the investigation.
☐ I'm being in touch when needed to sampling again.
Signature

Instruction
With all ballpoint pens sign five signatures in a place of paper no.1 on soft laminated clipboard writing pad.
With all ballpoint pens sign five signatures in a place of paper no.1 on hard laminated clipboard writing pad.
With all ballpoint pens sign five signatures in a place of paper no.2 on soft laminated clipboard writing pad.
With all ballpoint pens sign five signatures in a place of paper no.2 on hard laminated clipboard writing pad.
Brand of paper no.1 is "PaperOne" and Brand of paper no.2 is "DoubleAA".

Barcode	Please don't write any thing

Figure 3. English Version of Instruction Form.

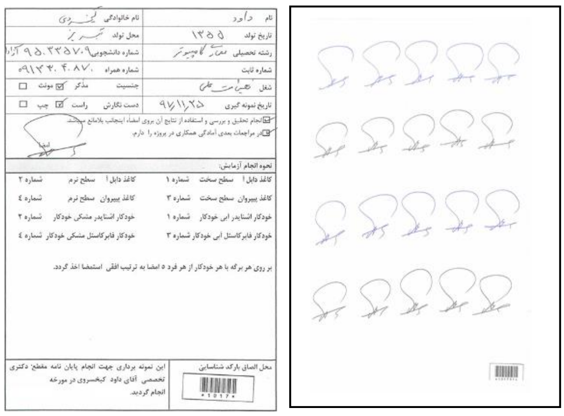


Figure 4. Samples of the signatures collected by one of the participants in the experiment.

2.1.2. MCYT-75

The MCYT-75 dataset includes offline hand-written signatures from 75 volunteers. According to this dataset, 15 samples of genuine signatures and 15 samples of forged signatures were collected for each candidate. In total, this dataset contains 1125 samples of genuine signatures and 1125 samples of forged signatures [39, 42].

2.1.3. UTSIG

The UTSIG dataset consists of the right-to-left offline signatures of the Persian language. The dataset contains 27 samples of genuine signatures and 45 samples of forged signatures. In total, this dataset contains 8235 signature samples of 116 volunteers [43].

2.1.4. ICDAR-Chinese and ICDAR-Dutch

Dataset ICDAR contains two sets of hand-written offline signatures. The first dataset contains 601 samples of Chinese signatures from 10 volunteers. The second dataset also includes 1933 samples of Dutch signatures from 54 participants [44, 45].

2.2. Deep Convolutional Neural Network

CNN is a better alternative to a conventional neural network that provides classification methods for machine vision. There are two stages of learning at CNN, including the Feed-Forward (FF) and the Back-Propagation (BP) phases [46, 47]. Three main layers are included in CNN: convolutional layers, pooling layers, and fully connected layers (FC) [48, 49, 50]. The convolution layer consists of kernels that slide over scanned signature images. The output of the convolution layer is called the feature map. The pooling layer is known as the reducing layer, which reduces the size of the neuron output from the convolution layer and reduces the computational load. The max-pooling layer, which selects only the maximum values for each feature map, has been used for this study. The fully connected layer has a direct link to all of the previous layer activations. In this research, the drop-out technique is used to prevent an over-fitting problem. In this way, at each stage of training, each neuron is thrown out of the network according to the probability that the network will be reduced [51]. In this study, the batch normalization (BN) layer is also used to normalize the data within the network [52, 53]. To reduce internal covariance change, the BN technique can increase network training speed. The transformation of the BN layer is described in Equation 1.

$$\hat{y}^{(l-1)} = \frac{y^{*(l-1)} - \mu_B}{\sqrt{(\sigma_B^2 + \varepsilon}} \qquad z^{*(l)} = \gamma^{(l)} \hat{y}^{(l-1)} + \beta^{(l)} \tag{1}$$

where the input vector to the BN layer is $y^{*(l-1)}$, $z^{*(l)}$ is the output response to the neurons in layer l, $\mu_B = E[y^{*(l-1)}]$, $\sigma_B^2 = \text{var}[y^{*(l-1)}]$, ε is a small constant corresponding to the numerical stability. $\gamma^{(l)}$ and $\beta^{(l)}$ are the parameters of scale and shift obtained by learning, respectively. An activation function is added after each layer. Relu and Softmax are used as two types of activation functions in this study. Relu, as defined in (5), is used in convolutional layers and can add nonlinearity and sparseness to the structure of the network.

$$R(d) = \begin{cases} d & if\ d > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The probability distribution of the output groups can be estimated by using the softmax activation function. The softmax is used in the last FC layer and is defined as follows:

$$\sigma(\mathrm{F})_{\mathrm{i}} = \frac{\mathrm{e}^{\mathrm{f_i}}}{\sum_{\mathrm{j}=1}^{\mathrm{k}} \mathrm{e}^{\mathrm{f_j}}} \quad \text{for } \mathrm{i} = 1, \ldots \mathrm{k} \ \textit{and} \ \mathrm{F} = (\mathrm{f}_1, \ldots, \mathrm{f}_\mathrm{k}) \in \mathrm{R}^\mathrm{k} \tag{3}$$

where F is the input vector and the output values of $\sigma(F)$ are between 0 and 1 and their sum is equal to 1 [54]. The loss function used in this investigation is based on the standard Cross-Entropy loss [55, 56], and Equation 4 shows how the loss function ratio is calculated and updated to reduce its amount.

$$J = -\sum_j y_{ij} \log P\left(y_j \,|S_i\right) \tag{4}$$

where J indicates a loss function ratio, S is a sample of the signature in the dataset, y is a volunteer to whom the signature belongs, and P is a prediction that estimates whether or not the signature belongs to a volunteer who is computed by Equation 5, i is the counter in range of the classes, and j equals class number.

$$P = \left(\exp\left(M(S_i) - M(S_j)\right) \Big/ 1 + \exp\left(M(S_i) - M(S_j)\right) \right) \tag{5}$$

M defines the model to be used, and S is the different samples of the signature dataset that are separated by index i and j.

2.3. Long Short Term Memory(LSTM)

Recurrent neural networks (RNNs) are an important branch of deep neural networks that are used to analyze complex systems. These networks can reduce the computational load by reducing the dimensions of the input data X_t and also improve the training performance. In addition, these networks provide the possibility of combining information among different inputs to obtain features that cannot be extracted using traditional feature extraction methods. The short-term long memory network is one of the recurrent neural networks, which are used to solve the weaknesses of the recurrent networks, such as solving the Exploding Gradient problems [55,56]. Unlike the recurrent neural network that simply calculates the balanced sum of the input signals and then passes through an activation function, each LSTM unit uses a memory Ct at time t. A memory cell consists of four main elements: an input gate or update gate Γ_u , a neuron with self-feedback connection, a forget gate Γ_f and an output gate Γ_o. The activation of the LSTM unit is described in Equation 6.

$$h_t = \Gamma_o . \tanh(C_t) \tag{6}$$

where Γ_o is the output gate and controls the amount of content that is provided through the memory. The output gate is calculated through the following relationship:

$$\Gamma_o = \sigma(W_o.[h_{t-1}, X_t] + b_0) \tag{7}$$

where σ is the softmax function, W_o and b_0 are the weight matrix and the initial bias vector. The memory cell C_t is also updated by partially forgetting the current memory and adding new memory content $\tilde{C_t}$ in the form of relationship 8, where the new memory content is obtained from Equation 9.

$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, X_t] + b_C) \tag{8}$$

$$C_t = \Gamma_f.C_{t-1} + \Gamma_u.\hat{C}_t \tag{9}$$

The amount of current memory to be forgotten is controlled by the forget gate $\Gamma_f$and the amount of new memory to be added to the memory cell is controlled by the update gate (input gate) Γ_u. This operation is shown in Equations 10 and 11.

$$\Gamma_f = \sigma(W_f.[h_{t-1}, X_t] + b_f) \tag{10}$$

$$\Gamma_u = \sigma(W_u.[h_{t-1}, X_t] + bu) \tag{11}$$

Figure 5 shows the structure of an LSTM recurrent neural network. This network, which has one input X_t, two outputs are produced: C_t and another output is h_t, which h_t itself is divided into two parts. A part is transferred to the next time step and another part is used in the current time step if needed to produce output. The forget gate Γ_f has the task of controlling the flow of information from the previous time step. This gate specifies whether memory information from the previous time step should be used or not, and if so, how much should be imported from the previous time step. The update gateway $\Gamma_u$is responsible for controlling the flow of new information. This gate specifies whether new information should be used in the current time step and, if so, to what extent. The output gate Γ_o also determines how much of the information from the previous time step is transferred to the next time step with the information of the current time step.

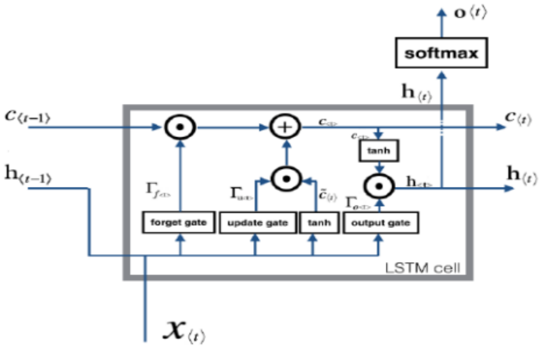


Figure 5. Structure of the LSTM Recurrent Neural Network [56]

The proposed deep network model in this study is created by combining a deep pre-trained network Inception_V3 with a proposed block (including two layers of LSTM, three layers of batch normalization, three layers of dropout, and two fully connected layers). By combining the Inception_V3 network with the LSTM network, the advantages of both networks can be used simultaneously. In many studies, the combination of LSTM networks with deep convolutional networks has been used to reduce feature dimensions, increase stability, reduce fluctuations, improve the training process, and increase recognition accuracy.

3. proposed method

In this section, the proposed method of this study, based on deep pre-trained network Inception_V3, will be presented. Figure 7 shows the block diagram of the proposed method for the network architecture.

3.1. Data preprocessing

In this section, data preprocessing on the collected dataset is described. For this purpose, all images are resized to 299x299 pixels. This resizing is because the Inception_V3 network accepts input of size 299x299. Then, to avoid the overfitting problem, the Data Augmentation (DA) technique [57] is used. For this purpose, the following operations are performed on the data: shape change, random color change, and random rotation between ±45. Based on the data augmentation technique, the training data is increased by 50%. Figure 6 shows the technique of adding data during the test.

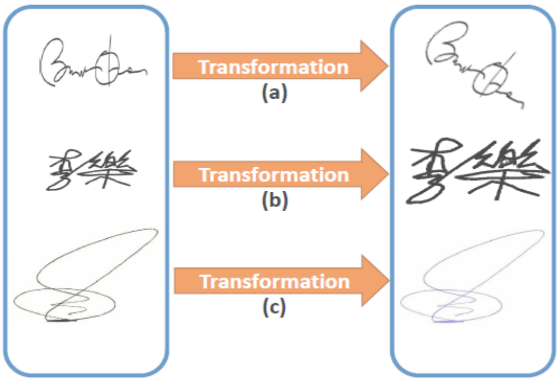


Figure 6. The DA technique was performed on the data.

Figure 7 shows how to divide the data for the training, validation, and test sets. According to this figure, the number of selected samples for training, validation, and test sets are 6064, 2022, and 674 respectively. Also, all samples were randomly selected for training and test sets.

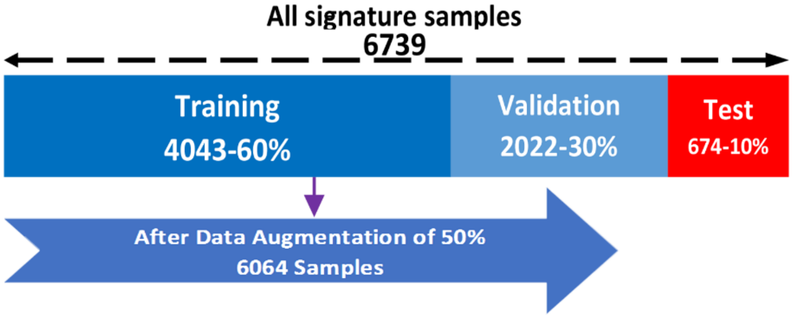


Figure 7. How to split the data for training, validation, and testing sets.

3.2. Proposed deep network architecture

The network architecture consists of a deep pre-trained network (Inception_V3) [58] with a proposed block including two LSTM layers, three batch normalization layers, three dropout layers, and two fully connected layers. The Inception_V3 pre-trained network consists of multiple layers, each layer learning specific features. The first layers of this network learn basic and low-level features, and the next layers have the task of learning complex and high-level features. In the mentioned process, the weight matrix is formed and adjusted by the network training process. The proposed block architecture is organized as follows (see Figure 8): (a). An FC layer with a linear function, a batch normalizer layer, Leaky-Relu function and followed by a dropout layer. (b). LSTM layer with Leaky-Relu function followed by batch normalization and dropout layers. (J). The architecture of the previous stage is repeated once more. (d). An FC layer with the nonlinear Softmax function is used to access the output layer. According to the network architecture, the output of the pre-trained network is a feature vector with a size of 256x512. In the first layer of the proposed block (FC), the linear function is applied to the learnable weights of the obtained features (w). To rearrange the dimensions of the feature vector into 256×1, $u(-\sqrt{\frac{1}{w}}, \sqrt{\frac{1}{w}})$ is used to evaluate the biased expected values. As can be seen from Figure 6, the dimensionality reduction in hidden layers continues from

112 × 112 (input size) to 112 (selected feature vector). In the end, the selected feature vector is connected to an FC layer with the nonlinear Softmax function.

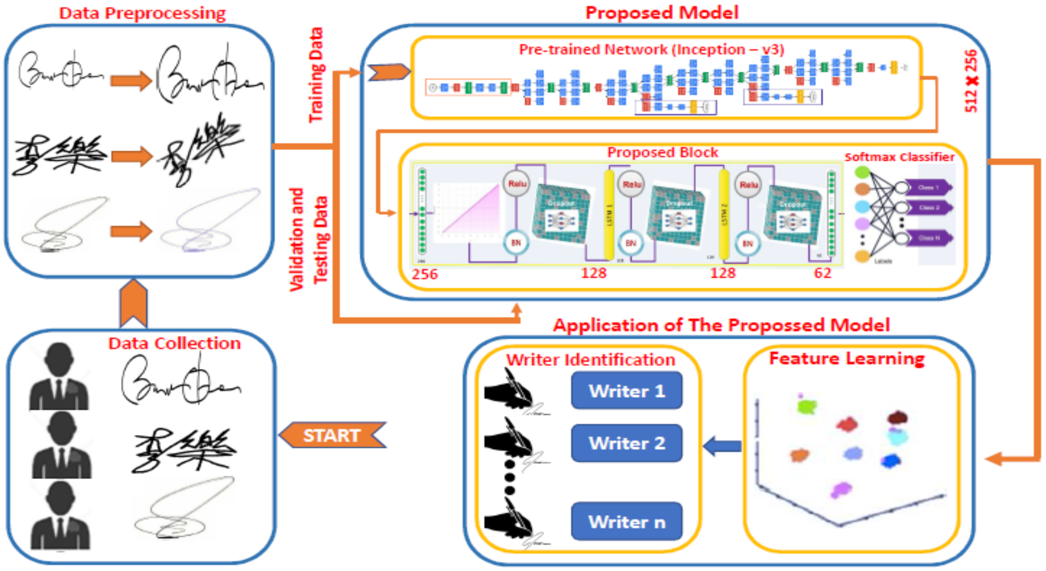


Figure 8. Block diagram of the Proposed Model (P-M).

3.3. Optimal hyperparameters

In this study, all the hyperparameters are carefully adjusted to achieve the best convergence rate; Finally, the cross-entropy error function and the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 have been selected. In an SGD, convergence is much faster than the Standard (or Batch) Gradient Descent [59]. Also, the conventional error backpropagation method with a batch size of 100 has been used to train the network. The trial and error method has been used to select hyperparameters. The optimal parameters selected for the proposed model after trial and error are shown in Table 2.

Table 2. The hyperparameters used in the proposed model.

Optimal value	Search space	Hyperparameters
SGD	RMSProp, Adam, Adamax, SGD, Adadelta	Optimizer
Cross-entropy	MSE, Cross-entropy	Loss function
0.2	0, 0.2, 0.3, 0.4, 0.5	Dropout rate
100	4, 8, 10, 16, 32, 64, 100	Batch size
0.01	0.01, 0.001, 0.0001	Learning rate
256	128, 256, 512	The number of neurons in the first FC layer of the proposed block
Linear	Leaky-Relu, Sigmoid, Relu, Linear	Activator function in the first FC layer of the proposed block
2	1, 2, 3, 4, 5	Number of LSTM layers
128	128, 256, 512	Number of recurrent neurons in LSTM layers
Relu	Leaky-Relu, Sigmoid, Relu, Linear	Activator function in the hidden layers of the proposed block
Softmax	Softmax, Sigmoid	The activation function in the last layer

4. Results and Discussion

In this section, simulation results will be presented based on the proposed network for automatic offline identification of the signature author. The specifications of the computer used in this study include an Intel Core i7-6700l CPU, 64GB RAM, and Geforce GTX TITAN X 12 graphics processor. The experimental results of the proposed model (Inception_V3 pre-trained network with the proposed block) and the Inception_V3 pre-trained network without using the proposed block are shown in Table 3.

Table 3. Experimental results of the P-M based on the DANASIG dataset

Examination Time (ms)		Examination Result	Pre-Trained Network
Test Time	Train Time	Accuracy	
1.65	16.54	99.70%	Inception_V3 (with the proposed block)
0.93	10.24	96.22%	Inception_V3 (Without the proposed block)

According to this table, as can be seen, the accuracy of the test related to the proposed model is around 98%, while for Inception_V3 without using the proposed block, this accuracy is around 96%. The training time for the proposed model is longer than the case where Inception_V3 is used. However, the time required to test the proposed model is promising. Figure 9 shows the test accuracy of the proposed model compared to the Inception_V3 network in 300 iterations.

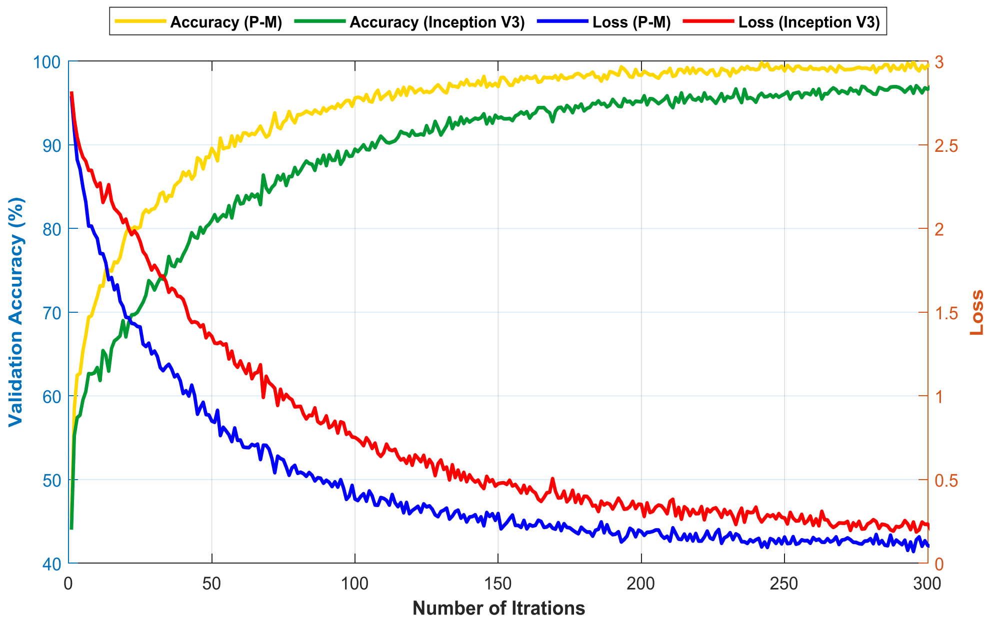


Figure 9. Accuracy and error obtained for the Inception_V3 with proposed block (P-M) compared to Inception_V3 network.

According to this figure, the proposed model has converged to 170 iterations. Also, the network Loss Ratio has reached its lowest value of 0.2. Table 4 shows the performance of the proposed model and the pre-trained Inception_V3 network without the proposed block for other datasets.

Table 4. Test accuracy of the P-M for datasets that have been widely used in recent research

Examination Result	**Dataset**				**Pre-Trained Network**
Accuracy(%)	**Sig#**	**Signer**	**Lang.**	**Name**	
99.10	8235	116	Persian	UTSig	**Inception_V3 (with the proposed block)**
97.20	601	10	Chinese	ICDAR	
97.50	863	20	Japanese	ICDAR	
99.76	1933	54	Dutch	ICDAR	
99.95	1125	75	Spanish	MCYT	
97.92	8235	116	Persian	UTSig	**Inception_V3 (Without the proposed block)**
86.17	601	10	Chinese	ICDAR	
85.19	863	20	Japanese	ICDAR	
96.43	1933	54	Dutch	ICDAR	
98.21	1125	75	Spanish	MCYT	

According to Table 4, the test accuracy of the proposed P-M model for Persian, Chinese, Japanese, Dutch, and Spanish datasets is 99.10, 97.20, 97.50, 99.76, and 99.95 respectively. These datasets have been widely used in recent research by other researchers. To evaluate the proposed model, the Accuracy criterion has been used, and Equation 12 shows how to calculate it. The Recall and Precision criteria were calculated and checked according to relations 13 and 14. However, because True Positive (Tp), True Negative (TN), False positive (FP), and False negative (FN) results are not used together in their calculations, it has not been used in related studies.

$$\text{Accuracy} = \frac{T_P + T_N}{T_P + F_P + T_N + F_N} \quad (12)$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \quad (13)$$

$$\text{Precision} = \frac{T_P}{T_P + F_P} \quad (14)$$

Based on the obtained results, it can be concluded that the performance of the proposed P-M model is very promising compared to the unmodified Inception_V3 network. In addition, the performance of the proposed P-M model is compared with recent studies of automatic signature author identification in Table 5.

According to Table 5, the accuracy of the proposed P-M model is higher than the compared studies. Furthermore, to demonstrate the utility of the proposed model, it is compared and evaluated against other popular methods (SVM [60] and BPNN [61]) using the DANASIG dataset. The SVM kernel is the Gaussian Radial Basis Function (RBF). A hidden layer with a sigmoid activation function is also used in BPNN. Five standard features that cannot be affected by temporal changes are selected as engineering features: Area, Centroid Coordinates, Eccentricity, Kurtosis, and Skewness [62, 63]. The accuracy of the test obtained from two methods of learning from raw data and learning using engineering features is shown in Table 6.

Table 5. Comparing the accuracy of the P-M with recent studies

Dataset				Examination			
Name	Lang.	Signer	Sig#	Reference	Year	Approach	Accuracy
UTSig	Persian	116	8235	Jampour et al.	2019	Chaos Game Theory	71.63%
				Masoudnia et al.	2019	CNN	94.68%
				Ghanim et al.	2018	SVM	94%
				Shariatmadari et al.	2019	Hierarchical CNN	87.12%
				P-M	2021	Modified Inception	99.10%
ICDAR	Chinese	10	601	Hadjadj et al.	2019	SVM	75.36%
				P-M	2021	Modified Inception	97.20%
	Japanese	20	863	Rantzsch et al.	2016	DNN	93.39%
				P-M	2021	Modified Inception	97.50%
	Dutch	54	1933	Hadjadj et al.	2019	SVM	92.31%
				Rantzsch et al.	2016	DNN	81.76%
				P-M	2021	Modified Inception	99.76%
MCYT	Spanish	75	1125	Masoudnia et al.	2019	CNN	98.01%
				Stauffer et al.	2019	Graph Embedding	98.38%
				Shariatmadari et al.	2019	Hierarchical CNN	94.54%
				P-M	2021	Modified Inception	98.21%

Table 6. Comparison of different learning methods.

Methods	Feature learning from raw data	engineering features
SVM	86.26	88.48
BPNN	83.18	85.73
P-M	99.70	85.45

According to Table 6, it is evident that feature learning based on raw data according to the proposed P-M model provides better results compared to engineering features. These desirable results are fully consistent with the proposed architecture, which can extract desirable features from the raw data layer by layer and recognize the author of the signature automatically. Moreover, unlike engineering features, learning based on raw data does not require prior knowledge of the problem/topic, which can improve the speed of network training. According to the same table, as it is clear, the results of all MLP networks, SVM, and the proposed P-M model are almost the same for engineering features; This shows that the proposed P-M model cannot make significant progress in identifying the signature author without learning the feature from the raw data.

As we know, handwritten documents such as signatures lose their originality over time due to the spreading and drying of the ink. To simulate the effect of environmental conditions on documents, multiple noises have been applied to the samples to create an inappropriate visual quality on them, which is not suitable for analysis and understanding by the user. In addition, after applying various noises, many common image processing such as edge detection, segmentation, etc. suffer from disturbances that should be considered. For this purpose, a realistic investigation should be adopted regarding whether the proposed network can perform well in a noisy environment. To answer this question, we have evaluated our proposed model in different noise environments according to Table 7.

Table 7. different noise environments

Type of Noise	Description
Gaussian	Gaussian white noise with constant mean and variance
Local var	Zero-mean Gaussian white noise with an intensity-dependent variance
Salt & Pepper	On and off pixels
Speckle	Multiplicative noise

Gaussian noise is noise whose probability density function is equal to the normal distribution, which is also called Gaussian distribution or white Gaussian noise [64]. As Equation 15 shows, the basic assumption of the noise model is that the image is corrupted by white Gaussian noise with an incremental zero mean [65].

$$l(x, y) = f(x, y) + n(x, y) \tag{15}$$

(x, y) represents the coordinates of a considered pixel, l(x, y) is the observed image, f(x, y) is the healthy image, and n(x, y) is Gaussian noise whose probability density function can be written as follows:

$$P(g) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(g-m)^2 / 2\sigma^2} \tag{16}$$

g represents the gray level, m is the mean of the function, and σ is the standard deviation of the noise in relation 16. Figure 10 shows the effect of Gaussian noise on the original image at two different SNRs.

SNR= 40 dB	SNR= -5 dB
	

Figure 10. The effect of Gaussian noise on the original image.

Local var noise is Gaussian white noise with zero means [66]. Figure 11 shows the effect of this noise on the original image at two different SNRs.

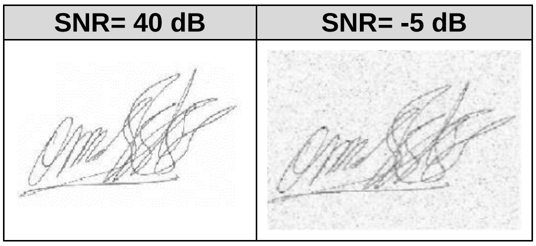


Figure 11. The effect of Local var noise on the original image.

Salt and pepper noise is known as the impulse or diffuse noise [67]. The image destroyed by this noise has bright pixels in the dark part and dark pixels in the bright part of the image. The probability density function of salt and pepper noise is shown by Equation 17 [68].

$$\boldsymbol{P(g)} = \begin{cases} P(a) \;\; for \;\; \text{g=a} \\ P(b) \;\; for \;\; \text{g=b} \\ 0 \;\; for \;\; \text{otherwise} \end{cases} \tag{17}$$

In figure 12, the black and white dots are shown as a result of this noise.

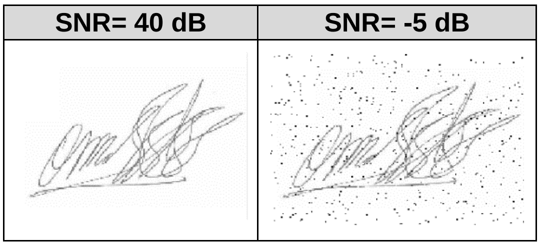


Figure 12. The effect of Salt and pepper noise on the original image.

Speckle noise spreads the average gray near a region, which is caused by the interference between light returning from uneven surfaces and the aperture. This noise is a type of multiplication that reduces the clarity and contrast of the image. It is assumed that the speckle noise has a generalized gamma distribution, so its probability density function follows the gamma distribution, which is shown in Equation 17 [69-71]. Figure 13 shows the effect of this noise on the image.

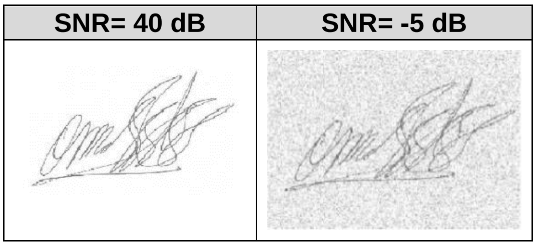


Figure 13. The effect of Speckle noise on the original image.

Figure 14 examines the performance of SVM, BPNN, and the proposed method in noisy environments at different SNRs.

According to this figure, as it is clear, the proposed model for removing noise and learning signature features, for grayscale images, performs better in the face of different noises.

Also, the proposed model has performed better in learning the features of images corrupted by local var and salt and pepper noises than speckle and Gaussian noises.

Nevertheless, the proposed model can be robust in a wide range of SNR. So that at 15 dB, the accuracy of the proposed method is still above 90% for all four added noises.

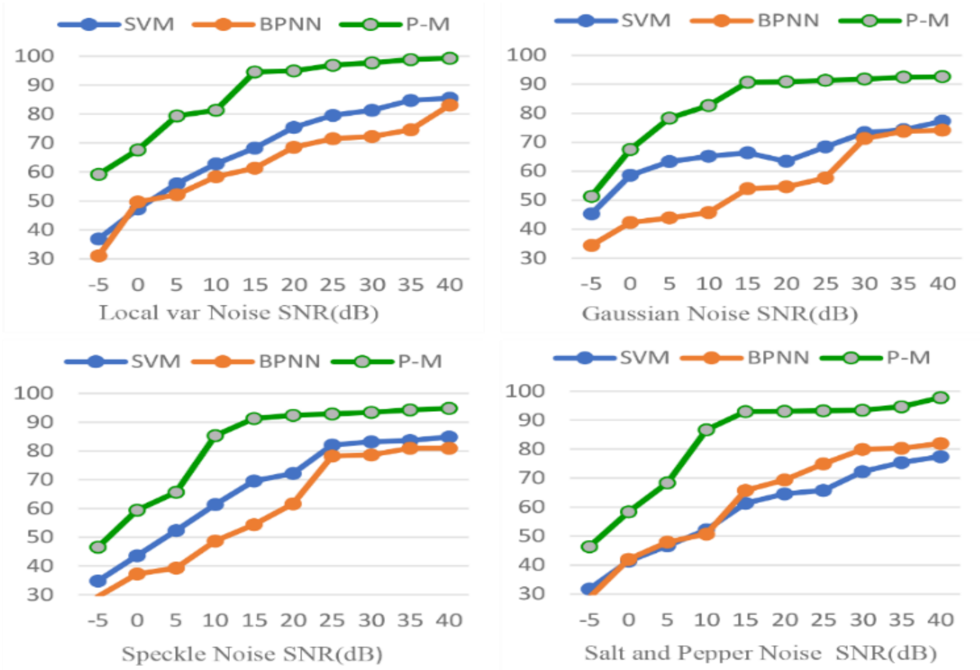


Figure 14. Classification accuracy results for SVM, BPNN, and the proposed model for noisy data at various SNR levels.

5. Conclusion

The handwritten signature can be considered the easiest way to confirm the identity of people. Considering the simplicity of handwritten signatures, it is necessary to take the necessary measures to secure the identity of the signature authors. Ensuring the identity of the signature authors traditionally is controlled and checked by visual inspection of the signature authors. In this study, the authors of handwritten signatures are automatically identified, based on deep learning networks, in offline mode. For this purpose, in this study, a comprehensive data set has been designed based on international standards. A deep convolutional network model based on pre-trained network Inception_V3 is designed and developed to extract features hierarchically from raw handwritten signatures. In addition to the collected dataset, the proposed model has been evaluated with other index datasets, including UTSig, ICDAR, ICDAR, and MCYT. Also, the proposed model has been compared with other methods and previous studies, and based on this, the results of the proposed model have been very promising. Moreover, the performance of the proposed model is satisfactory in noisy environments and can be robust in a wide range of different SNRs.

References

[1] Rafizadeh Shotorbani Fard, N., Hosseinzadeh, H., Bekravi, M. Accessibility Evaluation in Biometric Hybrid Architecture for Protecting Social Networks Using Colored Petri Nets. Journal of Advances in Computer Research, 2019; 10(1): 43-57.

[2] Mohammadzade, M., Ghonodi, A. Persian off-line signature recognition with structural and rotation invariant features using by the one-against-all SVM classifier. Journal of Advances in Computer Research, 2013; 4(2): 87-96.

[3] Masoudnia, S.; Mersa, O.; Araabi, B. N.; Vahabie, A.-H.; Sadeghi, M. A.; Ahmadabadi, M. N., Multi-representational learning for offline signature verification using multi-loss snapshot ensemble of CNNs. Expert Systems with Applications 2019, 133, 317-330.
[4] Deng, P. S.; Liao, H.-Y. M.; Ho, C. W.; Tyan, H.-R., Wavelet-based off-line handwritten signature verification. Computer vision and image understanding 1999, 76 (3), 173-190.
[5] Ng, A. Y.; Jordan, M. I. In On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes, Advances in neural information processing systems, 2002; pp 841-848.
[6] Peng, F.; Schuurmans, D.; Wang, S., Augmenting naive Bayes classifiers with statistical language models. Information Retrieval 2004, 7 (3), 317-345.
[7] Liao, Y.; Vemuri, V. R., Use of k-nearest neighbor classifier for intrusion detection. Computers & Security 2002, 21 (5), 439-448.
[8] Ghosh, R.; Kumar, P.; Roy, P. P., A dempster–Shafer theory based classifier combination for online signature recognition and verification systems. International Journal of Machine Learning and Cybernetics 2019, 10 (9), 2467-2482.
[9] López-García, M.; Ramos-Lara, R.; Miguel-Hurtado, O.; Cantó-Navarro, E., Embedded system for biometric online signature verification. IEEE Transactions on industrial informatics 2013, 10 (1), 491-501.
[10] Malik, M. I.; Liwicki, M. In From terminology to evaluation: Performance assessment of automatic signature verification systems, 2012 International Conference on Frontiers in Handwriting Recognition, IEEE: 2012; pp 613-618.
[11] Bhunia, A. K.; Alaei, A.; Roy, P. P., Signature verification approach using a fusion of hybrid texture features. Neural Computing and Applications 2019, 31 (12), 8737-8748.
[12] Hafemann, L. G.; Oliveira, L. S.; Sabourin, R., Fixed-sized representation learning from offline handwritten signatures of different sizes. International Journal on Document Analysis and Recognition (IJDAR) 2018, 21 (3), 219-232.
[13] 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 Systems with Applications 2018, 107, 182-195.
[14] Wan, Q.; Zou, Q. In Learning Metric Features for Writer-Independent Signature Verification using Dual Triplet Loss, 2020 25th International Conference on Pattern Recognition (ICPR), IEEE: 2021; pp 3853-3859.
[15] He, L.; Yi, C.; Wang, D.; Wang, F.; Lin, J.-h., Optimized minimum generalized Lp/Lq deconvolution for recovering repetitive impacts from a vibration mixture. Measurement 2021, 168, 108329.
[16] Chandra, S., Classification of Static Signature Based on Distance Measure Using Feature Selection. In Advances in Communication and Computational Technology, Springer: 2021; pp 707-717.
[17] Choi, S.-H.; Jung, S. H., Performance improvement of fake discrimination using time information in CNN-based signature recognition. Journal of Digital Contents Society 2018, 19 (1), 205-212.
[18] Eddy, S. R., What is a hidden Markov model? Nature biotechnology 2004, 22 (10), 1315-1316.
[19] Danielsson, P.-E., Euclidean distance mapping. Computer Graphics and image processing 1980, 14 (3), 227-248.
[20] De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D. L., The mahalanobis distance. Chemometrics and intelligent laboratory systems 2000, 50 (1), 1-18.
[21] Beylkin, G., Discrete radon transform. IEEE transactions on acoustics, speech, and signal processing 1987, 35 (2), 162-172.
[22] Reddy, U. J.; Ramana Reddy, B. V.; Reddy, B. E., Categorization & recognition of lung tumor using machine learning representations. Current Medical Imaging 2019, 15 (4), 405-413.
[23] Guo, Z.; Zhang, L.; Zhang, D., A completed modeling of local binary pattern operator for texture classification. IEEE transactions on image processing 2010, 19 (6), 1657-1663.
[24] Lin, T.; Jin, C.; Jordan, M. In On gradient descent ascent for nonconvex-concave minimax problems, International Conference on Machine Learning, PMLR: 2020; pp 6083-6093.
[25] Gardner, M. W.; Dorling, S., Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment 1998, 32 (14-15), 2627-2636.
[26] Noble, W. S., What is a support vector machine? Nature biotechnology 2006, 24 (12), 1565-1567.
[27] Peterson, L. E., K-nearest neighbor. Scholarpedia 2009, 4 (2), 1883.

[28] Van, B. L.; Garcia-Salicetti, S.; Dorizzi, B., On using the Viterbi path along with HMM likelihood information for online signature verification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 2007, 37 (5), 1237-1247.
[29] Nanni, L.; Lumini, A., A novel local online signature verification system. Pattern Recognition Letters 2008, 29 (5), 559-568.
[30] Vargas, J.; Travieso, C. M.; Alonso, J. B.; Ferrer, M. A. In Off-line signature verification based on gray level information using wavelet transform and texture features, 2010 12Th international conference on frontiers in handwriting recognition, IEEE: 2010; pp 587-592.
[31] Karouni, A.; Daya, B.; Bahlak, S., Offline signature recognition using neural networks approach. Procedia Computer Science 2011, 3, 155-161.
[32] Khalajzadeh, H.; Mansouri, M.; Teshnehlab, M., Persian signature verification using convolutional neural networks. International Journal of Engineering Research and Technology 2012, 1 (2), 7-12.
[33] Verma, R.; Rao, D., Offline signature verification and identification using angle feature and pixel density feature and both methods together. International Journal of Soft Computing and Engineering 2013, 2 (4), 740-46.
[34] Hafemann, L. G.; Sabourin, R.; Oliveira, L. S., Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognition 2017, 70, 163-176.
[35] Okawa, M., From BoVW to VLAD with KAZE features: Offline signature verification considering cognitive processes of forensic experts. Pattern Recognition Letters 2018, 113, 75-82.
[36] 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 Systems with Applications 2018, 107, 182-195.
[37] Ghosh, R., A Recurrent Neural Network based deep learning model for offline signature verification and recognition system. Expert Systems with Applications 2021, 168, 114249.
[38] Calik, N.; Kurban, O. C.; Yilmaz, A. R.; Yildirim, T.; Ata, L. D., Large-scale offline signature recognition via deep neural networks and feature embedding. Neurocomputing 2019, 359, 1-14.
[39] Yapıcı, M. M.; Tekerek, A.; Topaloğlu, N., Deep learning-based data augmentation method and signature verification system for an offline handwritten signature. Pattern Analysis and Applications 2021, 24 (1), 165-179.
[40] Shariatmadari, S.; Emadi, S.; Akbari, Y., Patch-based offline signature verification using one-class hierarchical deep learning. International Journal on Document Analysis and Recognition (IJDAR) 2019, 22 (4), 375-385.
[41] Fierrez-Aguilar, J.; Nanni, L.; Lopez-Penalba, J.; Ortega-Garcia, J.; Maltoni, D. In An on-line signature verification system based on the fusion of local and global information, International Conference on Audio-and Video-Based Biometric Person Authentication, Springer: 2005; pp 523-532.
[42] Ortega-Garcia, J.; Fierrez-Aguilar, J.; Simon, D.; Gonzalez, J.; Faundez-Zanuy, M.; Espinosa, V.; Satue, A.; Hernaez, I.; Igarza, J.-J.; Vivaracho, C., MCYT baseline corpus: a bimodal biometric database. IEE Proceedings-Vision, Image and Signal Processing 2003, 150 (6), 395-401.
[43] Soleimani, A.; Fouladi, K.; Araabi, B. N., UTSig: A Persian offline signature dataset. IET Biometrics 2016, 6 (1), 1-8.
[44] Blankers, V. L.; van den Heuvel, C. E.; Franke, K. Y.; Vuurpijl, L. G. In Icdar 2009 signature verification competition, 2009 10th International Conference on Document Analysis and Recognition, IEEE: 2009; pp 1403-1407.
[45] Malik, M. I.; Liwicki, M.; Alewijnse, L.; Ohyama, W.; Blumenstein, M.; Found, B. In ICDAR 2013 competitions on signature verification and writer identification for on-and offline skilled forgeries (SigWiComp 2013), 2013 12th international conference on document analysis and recognition, IEEE: 2013; pp 1477-1483.
[46] Sheykhivand, S.; Mousavi, Z.; Rezaii, T. Y.; Farzamnia, A., Recognizing emotions evoked by music using CNN-LSTM networks on EEG signals. IEEE Access 2020, 8, 139332-139345.
[47] Mousavi, Z.; Rezaii, T. Y.; Sheykhivand, S.; Farzamnia, A.; Razavi, S., Deep convolutional neural network for classification of sleep stages from single-channel EEG signals. Journal of neuroscience methods 2019, 324, 108312.
[48] LeCun, Y.; Bengio, Y.; Hinton, G., Deep learning. nature 521 (7553), 436-444. Google Scholar CrossRef 2015.

[49] Xing L, Qiao Y (2016) Deepwriter: a multi-stream deep CNN for text-independent writer identification. In: 2016 15th international conference on frontiers in handwriting recognition (ICFHR), IEEE, pp 584–589.
[50] Hasibuan, Z. A. In Towards Using Universal Big Data in Artificial Intelligence Research and Development to Gain Meaningful Insights and Automation Systems, 2020 International Workshop on Big Data and Information Security (IWBIS), IEEE: 2020; pp 9-18.
[51] Mousavi, Z.; Varahram, S.; Ettefagh, M. M.; Sadeghi, M. H.; Razavi, S. N., Deep neural networks–based damage detection using vibration signals of finite element model and real intact state: An evaluation via a lab-scale offshore jacket structure. Structural Health Monitoring 2021, 20 (1), 379-405.
[52] Sheykhivand, S.; Rezaii, T. Y.; Mousavi, Z.; Delpak, A.; Farzamnia, A., Automatic identification of epileptic seizures from EEG signals using sparse representation-based classification. IEEE Access 2020, 8, 138834-138845.
[53] Sheykhivand, S.; Rezaii, T. Y.; Farzamnia, A.; Vazifehkhahi, M. In Sleep stage scoring of single-channel EEG signal based on RUSBoost classifier, 2018 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), IEEE: 2018; pp 1-6.
[54] Sheykhivand, S.; Mousavi, Z.; Mojtahedi, S.; Rezaii, T. Y.; Farzamnia, A.; Meshgini, S.; Saad, I., Developing an efficient deep neural network for automatic detection of COVID-19 using chest X-ray images. Alexandria Engineering Journal 2021, 60 (3), 2885-2903.
[55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.
[56] Huang, Z., Xu, W. and Yu, K., 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.
[57] Van Dyk, D.A. and Meng, X.L., 2001. The art of data augmentation. Journal of Computational and Graphical Statistics, 10(1), pp.1-50.
[58] Huang, J., Gong, W. and Chen, H., 2019, December. Menfish Classification Based on Inception_V3 Convolutional Neural Network. In IOP Conference Series: Materials Science and Engineering (Vol. 677, No. 5, p. 052099). IOP Publishing.
[59] Hoseini, F., Shahbahrami, A., Bayat, P. A Hybrid Optimization Algorithm for Learning Deep Models. Journal of Advances in Computer Research, 2018; 9(4), 59-71.
[60] Xia, X.; Xu, C.; Nan, B. In Inception-v3 for flower classification, 2017 2nd International Conference on Image, Vision, and Computing (ICIVC), IEEE: 2017; pp 783-787.
[61] Huang, G.; Liu, S.; Van der Maaten, L.; Weinberger, K. Q. In Condensenet: An efficient densenet using learned group convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018; pp 2752-2761.
[62] LeCun, Y. A.; Bottou, L.; Orr, G. B.; Müller, K.-R., Efficient backprop. In Neural networks: Tricks of the trade, Springer: 2012; pp 9-48.
[63] Hossain, MD Moinul., 2019. IMPROVING GLOBAL NEIGHBORHOOD STRUCTURE MAP DENOISING APPROACH FOR DIGITAL IMAGES. on International
[64] Narayan P. Bhosale and Ramesh R. Manza., 2013. Analysis of Effect of Noise Removal Filters on Noisy Remote Sensing Images on International Journal of Scientific & Engineering Research (IJSER), vol. 4, issue 10, page 1151, ISSN: 2229-5518.
[65] V. Saradhadevi and V. Sundaram., 2012. An Enhanced Two-stage Impulse Noise Removal Technique for SAR Images Based on Fast ANFIS and Fuzzy Decision on European Journal of Scientific Research, X, vol. 68, no. 4 pp. 506–522.
[66] Hossain, MD Moinul., 2019. IMPROVING GLOBAL NEIGHBORHOOD STRUCTURE MAP DENOISING APPROACH FOR DIGITAL IMAGES. on International Conference on Interfaces and Human-Computer Interaction.
[67] Dhruv B, Mittal N, Modi M., 2017. Analysis of different filters for noise reduction in images. On recent developments in control, automation & power engineering, Noida, pp 410–415.
[68] Koli M. & Balaji S., 2013. Literature survey on impulse noise reduction, Signal & Image Processing on International Journal (SIPIJ) Vol.4, No.5
[69] Benzarti F. & Amiri H., 2013. Speckle Noise Reduction on Medical Ultrasound Images Signal.
[70] Kaur T, Sandhu M., Performance Comparison of Transform Domain for Speckl Reduction in Ultrasound Image on International Journal of Engineering Research and Application, Vol.2, Issue 1, pp.184-188.
[71] Salivahanan S., Vallavaraj A. & Gnanapriya C., 2008. Digital Signal Processing. Tata Mcgraw-