

# Signature verification based on proposed fast hyper deep neural network

Zainab Hashim<sup>1</sup>, Hanaa Mohsin<sup>1</sup>, Ahmed Alkhayyat<sup>2</sup>

<sup>1</sup>Department of Computer Sciences, University of Technology, Baghdad, Iraq

<sup>2</sup>Department of Qaultiy Assurance, The Islamic University, Najaf, Iraq

## Article Info

### Article history:

Received Mar 27, 2023

Revised May 27, 2023

Accepted Jun 12, 2023

### Keywords:

Deep learning

Fast Fourier transform

Gray-level co-occurrence matrix

Principal component analysis signature verification

## ABSTRACT

Many industries have made widespread use of the handwritten signature verification system, including banking, education, legal proceedings, and criminal investigation, in which verification and identification are absolutely necessary. In this research, we have developed an accurate offline signature verification model that can be used in a writer-independent scenario. First, the handwritten signature images went through four preprocessing stages in order to be suitable for finding the unique features. Then, three different types of features namely principal component analysis (PCA) as appearance-based features, gray-level co-occurrence matrix (GLCM) as texture-features, and fast Fourier transform (FFT) as frequency-features are extracted from signature images in order to build a hybrid feature vector for each image. Finally, to classify signature features, we have designed a proposed fast hyper deep neural network (FHDNN) architecture. Two different datasets are used to evaluate our model these are SigComp2011, and CEDAR datasets. The results collected demonstrate that the suggested model can operate with accuracy equal to 100%, outperforming several of its predecessors. In the terms of (precision, recall, and F-score) it gives a very good results for both datasets and exceeds (1.00, 0.487, and 0.655 respectively) on Sigcomp2011 dataset and (1.00, 0.507, and 0.672 respectively) on CEDAR dataset.

*This is an open access article under the [CC BY-SA](#) license.*



## Corresponding Author:

Zainab Hashim

Department of Computer Sciences, University of Technology

Baghdad, Iraq

Email: cs.20.16@grad.uotechnology.edu.iq

## 1. INTRODUCTION

One of the key developments in modern scientific study is the advancement of biometric verification techniques [1]. Meanwhile, it is well recognized that the present approaches to biometry [2]–[5] have a number of issues that restrict the range of their applicability. Typically, when high levels of security and authentication are necessary, biological and behavioral features are utilized. Biological features include the face, fingerprint, palm, iris, and retina, while behavioral features include signature and voice. In many areas of our lives, such as banks, educational institutions, attendance monitoring systems, and official document verification, where the need for authenticity is paramount, handwritten signature verification has become an integral aspect [6]. Based on the technique of acquisition, signatures are divided into offline and online signatures. Online signatures are obtained using digital devices like tablets or electronic pens that can capture real-time features as opposed to traditional pen and paper signatures (off-line), which are afterwards scanned as a digital file [7]. The process of deciding whether a signature is real or fake is known as signature verification, which is why it is described as a two-class problem. Signature verification phases are similar to identification phases, except in the classification phase, where its legitimacy to that class will be verified and the tested signature class will

be known [8]. The two basic approaches employed in the verification process are model-based verification and distance-based verification. In the model-based approach, models such as the convolutional neural network (CNN), hidden Markov model (HMM), and support vector machine (SVM) are created to represent the distribution of data. While in the distance-based method, dynamic time wrapping (DTW) compares the test signature with the reference signature using distance measures [9]. The major aim of this work is the development of an accurate writer-independent off-line signature verification model that protects from unexpected forgery with a small error rate. Finding more effective and relevant hybrid features to represent the handwritten signature as well as building a robust classifier to train these features can help overcome this issue. The following succinct summary of contributions that try to address the research's primary issue and improve classification accuracy:

- The proposed work extracts meaningful and perfect signature features based on fusion of appearance-based features, texture-based features, and frequency-based features.
- Using a new fast hyper deep neural networks (FHDNN) architecture to classify the extracted features. The proposed deep model achieves improvement in results compared with the previous approaches based on accuracy and equal error rate (EER) measures.
- The system achieved great performance on it by using two types of datasets from the more popular challenge handwritten signature datasets without any data augmentation techniques.

This research has the following framework: section 2 introduces recent research on signature verification systems. Section 3, explains the proposed model stages in details. In section 4, the findings and comments are covered. Section 5 contains the conclusions and recommendations for future work.

## 2. RELATED WORKS

Many researchers have investigated the use of handwritten signatures to confirm identity and have developed processing techniques. One of the first research on signature verification was carried out in 1977. The research involved features that were taken from horizontal and vertical regions of each signature [10]. The researches then carried on.

Banerjee *et al.* [11] created a language-invariant off-line signature verification approach. The modified signal of the signature image was used to extract 4 various kinds of features, including statistical, shape-based, similarity-based, and frequency-based features. A unique wrapper feature selection method built on the red deer algorithm has also been developed to minimize the feature dimension. The authentication and verification procedures were carried out with the Nave Bayes classifier.

Ruiz *et al.* [12] used the Siamese neural networks to help solve the off-line handwritten signature verification. To enhance the number of samples and the diversity required for training deep neural networks, 3 different kinds of synthetic data have been examined: the global public dataset of synthetic (GPDS-synthetic) dataset, compositional synthetic signature generation from shape primitives, and augmented data samples from the GAVAB dataset. Combining real and fake signatures for training led to the best verification outcomes.

Tayeb *et al.* [13] suggested a method for validating written signatures on bank checks. In order to identify anomalies, this model used a convolutional neural network (CNN) to analyze the pixels in a signature image. The ability of a CNN to extract features helps speed up feature analysis and processing for signature verification.

Hadjadj *et al.* [14] offered a method for determining whether a signature is authentic by using the textural information in the image of the signature. The local ternary patterns (LTP) and the orientated basic image features (oBIFs) are two textural descriptors that were utilized to describe the signature images. The distances between pairs of real and fake signatures were utilized to train SVM classifiers. Signature images were projected in feature space (a unique SVM for every single descriptor).

Navid *et al.* [15] aimed to utilize convolutional neural networks (CNN) to automate the process of signature verification. This model was built on top of the VGG-19, a pre-trained convolutional neural network. Bhunia *et al.* [16] proposed a one-class support vector machines (SVMs) in order to create a writer-dependent signature verification method based on two distinct texture feature types, namely discrete wavelet and local quantized patterns (LQP) features, in order to generate two distinct authenticity scores for a given signature.

Arisoy [6] proposed a writer-independent signature verification system based on one-shot learning. Siamese neural network was performed in order to recognize authentic signatures from fake ones. All these works above are based mostly on deep learning neural networks to implement signature verification processes. In this paper, we present a new fast hyper deep neural network model based on hybrid features in order to verify handwritten signatures.

### 3. THE PROPOSED METHODOLOGY

The objective of this study is the development of an accurate writer-independent off-line signature verification model that protects from unexpected forgery with small error rate. Basically, signature verification is categorized into three stages, beginning with signature images pre-processing, continuing with feature extraction, and ending up with a classification process. The off-line signature images are obtained from two famous signature datasets: SigComp2011 (composed of 937 signature images) [17] and CEDAR (composed of 2640 signature images) [18], which are divided into two classes: genuine and forged. To prepare the signatures for the verification stage, we first pre-process all training and test images. The images are converted to gray-scale, histogram equalized, blurred, and resized. Then, three feature extraction methods are applied to the processed images; these are: principal component analysis (PCA), fast fourier transform (FFT), and gray-level co-occurrence matrix (GLCM) producing hybrid features. Lastly, the resultant hybrid feature vectors are entered into the new fast hyper deep neural network (FHDNN) architecture to be trained in order to use them later to classify the new features. Figure 1 shows the suggested procedure for verifying signatures.

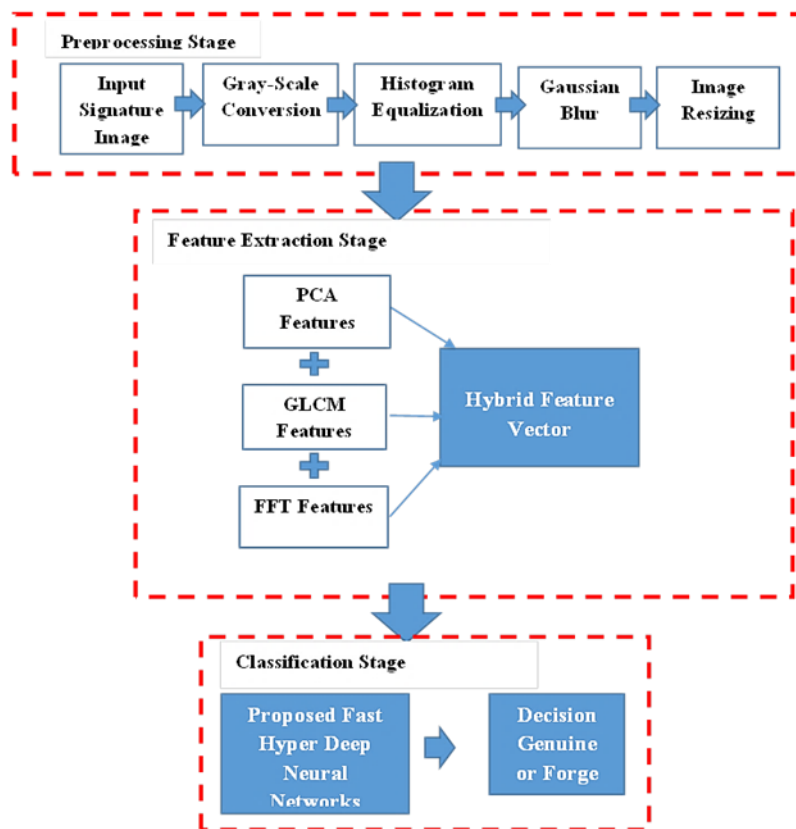


Figure 1. The proposed signature verification procedure

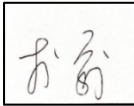
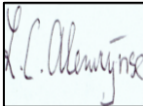
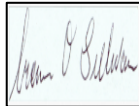
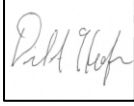
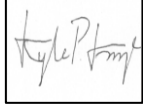
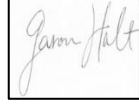
#### 3.1. Data acquisition stage

For data acquisition, two different datasets are used. First, is the Dutch and Chinese subset of signature verification championship 2011 dataset (SigComp2011) which are PNG images, scanned at 400 dpi, Red, Green and Blue (RGB) color with total number of signers of 74 for Dutch and Chinese and each signer has varying number of genuine and forged signatures creating 937 signatures. Second, is CEDAR dataset that consist of 55 signers each has 24 genuine signatures and 24 forged signatures creating 2640 signature in total. All signatures were scanned at 300 dpi in grayscale and binarized using a gray-scale histogram. Table 1 shows samples of these datasets.

#### 3.2. Data pre-processing stage

Data preparing is a crucial stage in signature verification. The objective of this stage is to raise the quality of the signature images and prepare them for easy extraction of the distinctive features. The following procedures are part of this pre-processing stage.

Table 1. Samples of handwritten signature images from SigComp2011 and CEDAR datasets

	Sample 1	Sample 2	Sample 3
SigComp2011 Chinese and Dutch			
CEDAR			

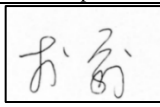
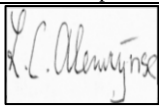
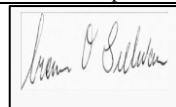
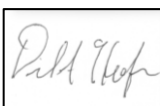
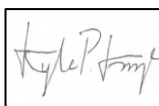
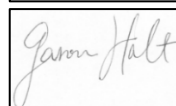
### 3.2.1. Gray-scale image

The first step in signature pre-processing is to convert signature images format from the traditional RGB color format and 24-bit gray scale format to 8-bit gray-scale images using luminosity method. According to (1).

$$Gray = (0.21 \times R) + (0.72 \times G) + (0.07 \times B) \quad (1)$$

Because each pixel requires less data to be delivered in a grayscale image, utilizing this format will simplify the signature extraction procedure and speed up processing. As indicated in Table 2, the 256 possible shades of gray, that vary from black to white, are stored as an 8-bit integer representing the grayscale level.

Table 2. Gray-scale conversion of signature images from SigComp2011 and CEDAR datasets

	Sample 1	Sample 2	Sample 3
SigComp2011 Chinese and Dutch			
CEDAR			

### 3.2.2. Histogram equalization (HE)

Low contrast, caused by issues with lighting or an irregular distribution of image illumination, may affect some iconic images. Because of this, after converting signature images to grayscale, histogram equalization (HE) is applied on small regions of image. HE's main goal is to enhance the histogram distribution of intensities and boost the overall contrast of the image. This makes it possible for regions with less local contrast to acquire more contrast. This can be accomplished via histogram equalization by spreading the majority of frequent values for intensity equally [19]. After applying HE a heavy noise is shown hidden in the background of signature images in both datasets as shown in Table 3. Due to the restricted dynamic range that imaging sensors have, signature images taken by digital cameras in low light circumstances display minimal contrast in dark or bright parts.

### 3.2.3. Gaussian blurring

In order to clarify the signature from the noisy background, Gaussian blur filter of size 7\*7 is used to blur the image. This filter will blur some details that were in the original image that were clear, enhancing the completed image in which the signature will stand out more than the background as presented in Table 4. Gaussian blur is a type of visual blurring that averages the pixels using the weight idea. According to the normal distribution, weights are supplied to each pixel [20], the following definition of (2) of Gaussian blur is provided.

$$GB_p = \sum_{q \in S} G_{\sigma} ||p - q|| I_q \quad (2)$$

### 3.2.4. Image resizing

Finally, because some of the signature images in both datasets (SigComp2011 and CEDAR) are significantly larger or smaller than others, they were resized to 50\*50 as shown in Table 5. Our deep learning model architecture and some of the feature extraction algorithms we used require signature images to be the

same size, whereas our raw collected images can be of varying sizes. We use bicubic interpolation as a resizing method, the bicubic interpolation estimates the pixels in the (i, j) positions using a sampling distance of 16 adjacent pixels (4×4) [21] as shown in (3).

$$f_{i,j} = [W_{-1}(S_Y) W_0(S_Y) W_1(S_Y) W_2(S_Y)] \begin{pmatrix} f_{i-1,j-1} & f_{i,j-1} & f_{i+1,j-1} & f_{i+2,j-1} \\ f_{i-1,j} & f_{i,j} & f_{i+1,j} & f_{i+2,j} \\ f_{i-1,j+1} & f_{i,j+1} & f_{i+1,j+1} & f_{i+2,j+1} \\ f_{i-1,j+2} & f_{i,j+2} & f_{i+1,j+2} & f_{i+2,j+2} \end{pmatrix} \begin{bmatrix} W_{-1} & (S_X) \\ W_0 & (S_X) \\ W_1 & (S_X) \\ W_2 & (S_X) \end{bmatrix} \quad (3)$$

Where:

$$W_{-1}(S_X) = \frac{-S^3 + 2S^2 - S}{2}$$

$$W_0(S_X) = \frac{-3S^3 + 4S^2 + 2}{2}$$

$$W_1(S_X) = \frac{S^3 - S^2}{2}$$

$$W_2(S_X) = \frac{S^3 - S^2}{2}$$

Table 3. Histogram equalized signature images from SigComp2011 and CEDAR datasets

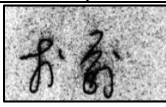
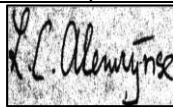

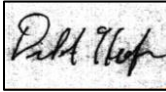
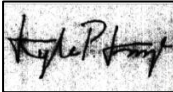
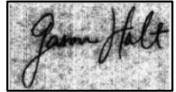
	Sample 1	Sample 2	Sample 3
SigComp2011 Chinese and Dutch			
CEDAR			

Table 4. Gaussian blurred images from SigComp2011 and CEDAR datasets

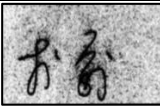
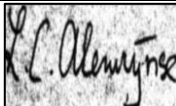
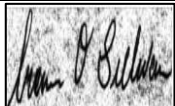
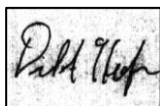
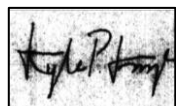
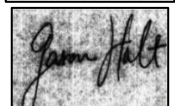




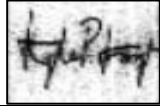

	Sample 1	Sample 2	Sample 3
SigComp2011 Chinese and Dutch			
CEDAR			

Table 5. Resized signature images from SigComp2011 and CEDAR datasets

	Sample 1	Sample 2	Sample 3
SigComp2011 Chinese and Dutch			
CEDAR			

### 3.3. Feature extraction stage

The input signature image is first processed by hybrid feature extraction technique that combines features extracted from three techniques these are PCA, FFT, and GLCM in order to find distinctive signature features. The input image is then transformed to a vector of geometric features. The shape of output feature vector is (106\*1).

#### 3.3.1. Principal component analysis (PCA)

A simple, non-parametric technique called PCA can be used to eliminate relevant information from large datasets and reduce the number of dimensions without sacrificing information [22]. It calculates the distance between two objects using the Euclidean distance concept. To discover the collection of projection vectors, the Eigensignature approach, based on principal component analysis (PCA), is employed [23]. In this

paper, we will use PCA as feature extractor to extract appearance-based signature features from each image based on Eigensignature method producing (50\*1) feature vector which later will be combined with FFT and GLCM feature vectors. The signature image matrices in two dimensions will be converted into one dimension row vector. The Eigenvector, a signature code set, is created using the Eigensignature method, which extracts important information from a signature image. The signature database, which contains signature codes from earlier training, is then used to compare this signature code. The principal component is calculated as follows: First, the resized signature images are read and a training set of total M images is created in order to use them in computing the average mean. The input image is then subtracted from the average mean, as indicated in (4) [24].

$$\phi_i = \bar{X}_i - X \quad (4)$$

After that, the covariance matrix is calculated according to (5) [23].

$$C = 1/M \sum_{n=1}^m \phi_n \phi_n \quad (5)$$

Most effective Eigen values are sorted and selected after computing the covariance matrix's Eigenvalues and Eigenvectors. A collection of Eigenvectors' highest Eigenvalues are picked. The training samples are projected onto Eigenimages to obtain feature space.

### 3.3.2. Gray-level co-occurrence matrix (GLCM)

An excellent technique for extracting texture information is the gray level co-occurrence matrix (GLCM). In 1973, Haralick proposed this strategy with the aid of a research team [25]. The GLCM is organized as a two-dimensional, dimensionless histogram with pairs of pixels separated by their spatial relationship. Analyzing a collection of co-occurrence matrices is often how the issue with texture discrimination is approached. It uses a statistical method that includes a row and column index. The values of P (i, j) at specific positions describe how frequently gray levels i and j occur at a particular distance and in a particular direction [26]. The indexed data correspond to the specified range of image data. Utilizing the vector d, which is displaced by the radius  $\delta$  and direction  $\theta$ , the GLCM is computed. Based on image representation, Haralick used GLCM to extract 13 texture features [27].

We have used 6 GLCM features namely contrast, energy, homogeneity, entropy, mean, and inverse producing (6\*1) GLCM feature vector. And the GLCM matrix was created with 256 levels, radius=1 and in the horizontal direction.

- Contrast: is a measurement of local differences or intensity at the grayscale level. Over the entire image, it measures the variations between the pixel point and its neighbors. According to one theory, a high-contrast image has more tones at either end of the spectrum than a low-contrast image, which has a smoother range of gray tones (black and white). In (6) displays the key formula utilized in contrast calculations:

$$\text{Contrast} = \sum_{i,j=0}^{N-1} \widetilde{P}_{ij} (i - j)^2 \quad (6)$$

where  $\widetilde{P}$  is the estimated probability of the groups of pairs of surrounding gray-levels in the image and N is the overall number of gray-levels employed (the GLCM dimensions) [28].

- Energy: is a metric of similarity or angular second momentum (ASM), which evaluates the consistency of the textural representation, or the repeating of pixel pairs, as illustrated in (7). It is in charge of identifying texture disorders. Energy can reach a maximum value of 1 [29].

$$\text{Energy} = \sum_{i,j=0}^{N-1} (\widetilde{p}_{ij})^2 \quad (7)$$

- Entropy: this is typically categorized as an initial measure of the level of chaos in an image, is another crucial GLCM property to distinguish an image texture. In (8) can be used to quickly calculate the GLCM derived entropy from the GLCM elements [26], which is inversely proportional to GLCM energy.

$$\text{Entropy} = - \sum_{i,j=0}^{N-1} P_{ij} \log P_{ij} \quad (8)$$

- Homogeneity: is also called inverse difference moment (IDM). The GLCM's diagonal elements with high values indicate that the visual texture is highly homogeneous. The homogeneity is greatest when the image pixel values are same [29]. Due to the GLCM's large yet adverse relationship between contrast and

homogeneity, homogeneity decreases as contrast increases while being constant in energy [29]. The IDM is shown in (9).

$$IDM = \sum_{i,j=0}^{N-1} \frac{\bar{p}_{ij}}{1+(i-j)^2} \quad (9)$$

- Mean: compared to other GLCM textural features, it seems to be the best way to measure GLCM texture. The GLCM Mean is not simply the total of all the original values of pixels in the image window; instead, it is numerically equal to the GLCM dissimilarity, in which each pixel is valued by its frequency of appearance and a specific neighboring pixels [30] as shown in (10).

$$u_i = \sum_{i,j=0}^{N-1} i \bar{p}_{ij} \quad u_j = \sum_{i,j=0}^{N-1} j \bar{p}_{ij} \quad (10)$$

- Inverse: the last feature we use is inverse and is shown in (11).

$$Inverse = \sum_{i,j=0}^{N-1} \frac{\bar{p}_{ij}}{(i-j)^2} \quad (11)$$



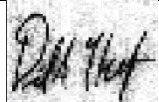
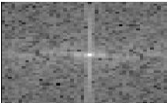
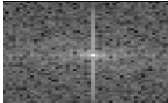
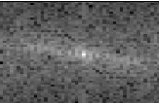
### 3.3.3. Fast Fourier transform (FFT)

The FFT method, considered an efficient illustration of the discrete Fourier transform (DFT), works by transforming information from the time domain to the frequency domain. The spatial frequency of each object in the remote sensing image is unique. The shape, structure, texture, and other properties of various things can be efficiently reflected in their frequency spectrum energy [31]. In this paper, we will use FFT as feature extractor to extract frequency features from signature images producing (50\*1) feature vector. First, the FFT of a 2D signature image is calculated using (12) and (13), and is shown in Table 6.

$$f(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-\left(i \times x \times \pi \left(x \frac{m}{M} + \frac{n}{N}\right)\right)} \quad (12)$$

$$f(x,y) = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-\left(i \times x \times \pi \left(x \frac{m}{M} + \frac{n}{N}\right)\right)} \quad (13)$$

Table 6. The frequency spectral of the signature images

	SigComp2011 Chinese	SigComp2011 Chinese	CEDAR
Signature image			
Frequency signature image			

The pixel at location (m, n) is represented by f(m, n), while F(x, y) is the function to represent the image in the frequency domain with respect to position x and y, M × N indicates the image's dimension, and i is sqrt (-1). Then, we apply vector quantization on spectral signature images to convert it to feature vector. Finally, the PCA (50\*1) feature vector, GLCM (6\*1) feature vector and FFT (50\*1) feature vector are combined producing (106\*1) signature feature vector.

### 3.4. Classification stage

In this stage, classification processes are implemented by the proposed FHDNN. The proposed model's primary goal is to categorize the hybrid features that were derived from the earlier stage to establish if the signature is authentic or not. The proposed model built with 31 layers, nine of them were convolutional layers (with filters equal to 16, 32, 64, 128, 256, 512, and 35 respectively), eight max-pooling 1D layers, nine Leaky-ReLU 1D layers, four dense 1D layers and one flatten layer. Figure 2 shows the proposed model architecture. Since the input layer is 1D feature vector of size (106\*1), all layers with their parameters are build as 1D instead of 2D layers. Because of the straightforward and small arrangement of one-dimension layers, these layers have a low computing need, making real-time and inexpensive hardware implementation possible.



The proposed model is trained with 64 batches, 100 epochs, and a learning rate of 0.001. Our model design uses the adaptive momentum (Adam) optimizer, which has a learning rate of 0.001.

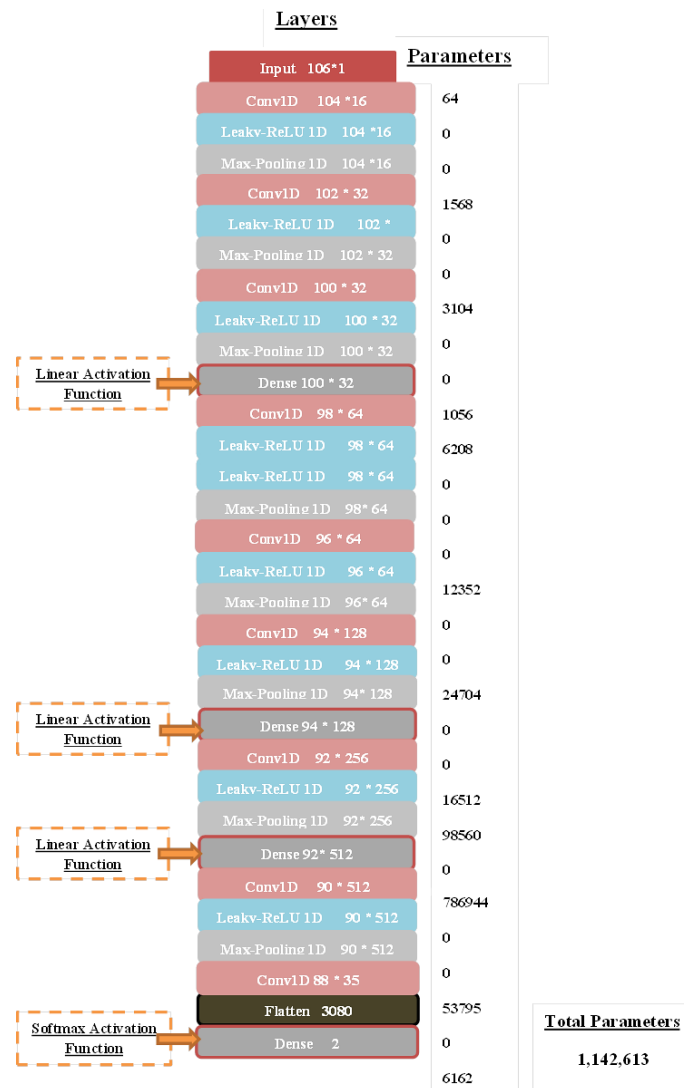


Figure 2. The proposed fast hyper model (FHDNN) architecture

A customized form of two-dimensional CNNs called one-dimensional convolutional neural networks (1D CNNs) is used with kernel of size 3, padding value equal to 1 (to give the kernel extra space to cover the vector, padding is applied to the feature vector's frame), and stride of 1 that modifies the amount of movement over the feature vector in which the filter will move one unit, at a time. Leaky rectified linear units (Leaky ReLU) function, a non-linear activation function, is applied after the CNN layers. When compared to conventional activation functions, the Leaky ReLU function has the ability to speed up the training of deep neural networks. Another reason for applying Leaky ReLU is because we have a large number of features with high negative values and the Leaky ReLU has small slop for negative values. A new pooling layer is added after the Leaky ReLU layers. The Max-Pooling method is used; the maximum output can be obtained. For 1D temporal data, the maximum value over the window determined by the pool size is used to downsample the input representation. The window is moved a little distance. When the "valid" padding option is used, the output has the shape: output shape= (input shape-pool size+1)/strides. In order to take the output of the preceding layers, "flatten" them, and create a single vector that can be an input for the following stage, one flattens layer is added. Three dense layers are used before the flatten layer with linear activation function and works as feature collector, and one dense layer used after the flatten layer with softmax activation function and



used as classifier of the resulted vectors. The dense layer's stated number of neurons will have an impact on the output shape. Dense layer carries out the action: activation (dot (input, kernel) +bias) is equal to output, Table 7 summarized all the layers of the proposed hyper deep model.

Table 7. Layers Summary of the proposed fast hyper deep model (FHDNN)

Layer	Filters	Parameters
Conv1D	16	Stride=1; Kernal-size=3
Max-pooling	16	Stride=1; Pool size=1
Conv1D	32	Stride=1; Kernal-size=3
Max-pooling	32	Stride=1; Pool size=1
Conv1D	32	Stride=1; Kernal-size=3
Max-pooling	32	Stride=1; Pool size=1
Dense	32	Linear Activation Function
Conv1D	64	Stride=1; Kernal-size=3
Max-pooling	64	Stride=1; Pool size=1
Conv1D	64	Stride=1; Kernal-size=3
Max-pooling	64	Stride=1; Pool size=1
Conv1D	128	Stride=1; Kernal-size=3
Max-pooling	128	Stride=1; Pool size=1
Dense	128	Linear activation function
Conv1D	256	Stride=1; Kernal-size=3
Max-pooling	256	Stride=1; Pool size=1
Dense	512	Linear activation function
Conv1D	512	Stride=1; Kernal-size=3
Max-pooling	512	Stride=1; Pool size=1
Conv1D	35	Stride=1; Kernal-size=3
Flatten		None
Dense		SoftMax activation function

#### 4. RESULTS AND DISCUSSION

The experimental results are presented in this section from using our technique on two different datasets of handwritten signatures. Additionally, the system's outcomes are compared with state-of-art methods that utilize and incorporate the same datasets. A learning rate of 0.001 is used to train the suggested hyper model, epochs of 100, and 64 batch sizes. The overall number of parameters obtained from our model is equal to (1,142,613).

The two datasets are used separately to compare the effectiveness and performance of our suggested approach with different approaches, these are: SigComp2011, which is considered a more challenging as it contains Chinese and Dutch signatures, and the popular CEDAR dataset. We divide the datasets randomly into training (70%) and test (30%) partitions. Five performance metrics-accuracy, precision, recall, F-score, and loss metrics-are used to evaluate the efficacy of our proposed methodology. According to (14), the percentage of true positive and true negative categorized points to all total points is known as accuracy.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

Where TP, TN, FP, and FN stand for the corresponding true positive, true negative, false positive, and false negative results. In (15) defines F1-score as the harmonic mean of precision and recall. In (16) and (17) illustrate precision and recall.

$$F1Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (15)$$

$$Precision = \frac{TP}{TP+FP} \quad (16)$$

$$Recall = \frac{TP}{TP+FN} \quad (17)$$

In case of the SigComp2011 dataset, when our method is compared to other state-of-the-art methods utilizing accuracy and EER on the Dutch and Chinese SigComp2011 dataset, it is evident that our method outperforms them, with accuracy value of 100% and EER of 0.0 as presented in Table 8. The suggested strategy has improved accuracy metrics with a growing range of (24.02%) compared to (Ismail Hadjadj (2019) Chinese [13]) that achieves an accuracy of 75.98%.

Table 8. Compares the outcomes of our model using the SigComp2011 dataset with those of other systems

Method	Accuracy	EER
Arisoy (2021) [6]	90.11%	-
Banerjee <i>et al.</i> (2021) Dutch [11]	99.28%	0.03
Banerjee <i>et al.</i> (2021) Chinese [11]	99.12%	0.06
Ruiz <i>et al.</i> (2020) [12]	99.44%	3.93
Tayeb <i>et al.</i> (2017) [13]	98.8%	-
Hadjadj <i>et al.</i> (2019) Dutch [14]	97.74%	-
Hadjadj <i>et al.</i> (2019) Chinese [14]	75.98%	-
Cozzens <i>et al.</i> (2018) Dutch [32]	84.74%	-
Solar <i>et al.</i> (2020) Dutch [33]	99.44%	0.03
Alvarez <i>et al.</i> (2016) Dutch [34]	97.00%	-
Alvarez <i>et al.</i> (2016) Chinese [34]	95.00%	-
Kao and Wen (2020) Dutch [35]	98.96%	-
Proposed method	100%	0.0

In addition to the two public datasets that we used, our model was also tested with a private dataset we collected from 31 signers, with 12 signature images for each signer as shown in Table 9. Every image has a signature made with a red, blue, or black pen. The results show an accuracy of 99.33% on this dataset.

Table 9. Samples of collected handwritten signatures for testing our model




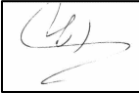




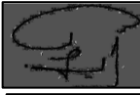
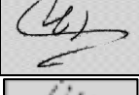

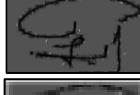



	Sample 1	Sample 2	Sample 3
Collected handwritten signatures			
Gray-scale signatures			
Histogram equilization signatures			
Gaussian blurred signatures			
Resized signatures			

Table 10 compares the outcomes from our model's use of the CEDAR dataset with several novel techniques. Our model gets better results and exceeded the pre-trained models when it comes to both accuracy and EER as shown in Figure 3. The small EER value indicates better performance. However, in our case, EER is equal to 0.0, which means that the proportion of misclassified genuine signatures and the proportion of misclassified fake signatures are almost the same as shown in (18), and this is due to the datasets we used. In practical applications EER should be near to zero.

$$EER = \frac{FAR + FRR}{2} \quad (18)$$

Where FAR is false accept rate and FRR is false reject rate.

The proposed model is used with features extracted from PCA, GLCM, and FFT which lead to finding the best results of (precision, recall, F-measure), Table 11 shows the proposed system results on SigComp2011 and CEDAR datasets. In term of speed, our model achieves great results in training each dataset on hp laptop with processor (11th Gen Intel(R) Core (TM) i7-1195G7 @ 2.90GHz 2.92 GHz), RAM size 16.0 GB, the screen card is NVIDIA GeForce MX350 and the operating system is windows10. For SigComp2011, each epoch in the proposed model executed within 6 sec that make the total training time is approximately 10 minutes, while for CEDAR dataset each epoch is executed in about 18 sec and that make the total training time 30 minute, and when we compare these results with the VGG16 and VGG19 models that were applied to the same datasets, it appeared that the suggested model outperformed the previous models in relation to speed as shown in Table 12. We also compared the accuracy and loss of our model with VGG16 model as shown in Table13. Our model overcomes the pre-trained model.

Table 10. Compares the outcomes of our model using the CEDAR dataset with those of other systems

Method	Accuracy	EER
Banerjee <i>et al.</i> (2021) [11]	99.36%	0.01
Hadjadj <i>et al.</i> (2017) [13]	97.99%	-
Navid <i>et al.</i> (2019) [15]	88%	-
Bhunia <i>et al.</i> (2019) [16]	-	7.59
Solar <i>et al.</i> (2020) [33]	99.15%	4.91
Nurullah Çalik <i>et al.</i> (2019) [36]	98.49%	-
Jampour and Naserasadi (2019) [37]	98.76%	-
Li Liu <i>et al.</i> (2021) [38]	-	6.74
Maruyama <i>et al.</i> (2021) [39]	-	0.82
Alsuhiat and Mohamad [40]	87.7%	11.40
Kadhm <i>et al.</i> [41]	99.7%	-
Proposed Method	100%	0.0

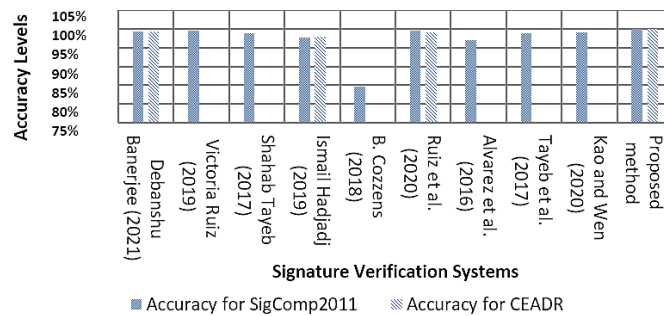


Figure 3. The accuracy obtained from our model on SigComp2011 and CEDAR dataset compared to other systems

Table 11. The proposed model results of SigComp2011 and CEDAR datasets

Dataset	Precision	Recall	F-score	Loss
SigComp2011	1.00	0.487	0.655	0.00001011
CEDAR	1.00	0.507	0.672	0.0000060849

Table 12. Comparison between the proposed model and VGG16 and VGG19 in term of speed

Method	SigComp2011 (speed for each epoch)	CEDAR (speed for each epoch)
VGG16	53 sec	179 sec
VGG19	51 sec	91 sec
The proposed model	6 sec	18 sec

Table 13. The proposed model results of SigComp2011 and CEDAR compared with pre-trained model

Method	Accuracy	Loss	Dataset
VGG16	49.85	8.17	SigComp2011
	50.71	7.94	CEDAR
The proposed model	100.00	0.00001011	SigComp2011
	100.00	0.0000060849	CEDAR

## 5. CONCLUSION

This study provides a hybrid feature-based method for handwritten signature verification and a proposed fast hyper deep neural network (FHDNN) that is applicable for writer-independent scenario. So as to assess the efficiency and performance of our suggested model, we employ two well-known datasets; SigComp2011 and CEDAR. As initial stage we perform four pre-processing stages on them. Then, we use PCA, GLCM, and FFT as feature extraction methods to build hybrid feature vector for each image. After that, these features are inputted into a proposed model, the proposed model's primary goal is to categorize the hybrid features that were derived from the earlier stage to identify a false signature from an authentic one. This model was built with 31 layers; nine of them were convolutional layers, eight Max-pooling layers, nine Leaky-ReLU layers, four Dense layers and one Flatten layer.

The suggested technique enhances the verification accuracy and outperforms the other previous modern techniques, with an accuracy value of 100% for both datasets and a speeding up the training time to about 10 minutes for the SigComp2011 dataset and 30 minutes for the CEDAR dataset. Additionally, it has a

high precision rate, which can be attributed to the model's architecture and the choice of effective signature features.

In future work, we will collect handwritten signatures in Arabic language to evaluate the verification performance of the proposed model on different languages. Also, we will build a model that depends on both writer-dependent and writer-independent approaches. Meanwhile, we'll keep investigating signature features to have the best verification effect with the least amount of training data.




## REFERENCES

- [1] Z. Hashim, H. M. Ahmed, and A. H. Alkhayyat, "A comparative study among handwritten signature verification methods using machine learning techniques," *Scientific Programming*, vol. 2022, pp. 1–17, Oct. 2022, doi: 10.1155/2022/8170424.
- [2] M. A. Taha and H. M. Ahmed, "Iris features extraction and recognition based on the local binary pattern technique," in *2021 International Conference on Advanced Computer Applications (ACA)*, IEEE, Jul. 2021, pp. 16–21. doi: 10.1109/ACA52198.2021.9626827.
- [3] M. A. Taha and H. M. Ahmed, "A fuzzy vault development based on iris images," *EUREKA: Physics and Engineering*, no. 5, pp. 3–12, Sep. 2021, doi: 10.21303/2461-4262.2021.001997.
- [4] H. M. A. Salman and S. Hameed, "Eye detection using Helmholtz principle," *Baghdad Science Journal*, vol. 16, no. 4(Suppl.), p. 1087, Dec. 2019, doi: 10.21123/bsj.2019.16.4(Suppl.).1087.
- [5] H. M. Ahmad and S. R. Hameed, "Eye diseases classification using hierarchical multilabel artificial neural network," in *2020 1st. Information Technology To Enhance e-learning and Other Application (IT-ELA)*, IEEE, Jul. 2020, pp. 93–98. doi: 10.1109/IT-ELA50150.2020.9253120.
- [6] M. V. Arisoy, "Signature verification using Siamese neural network one-shot learning," *International Journal of Engineering and Innovative Research*, vol. 3, no. 3, pp. 248–260, Sep. 2021, doi: 10.47933/ijeir.972796.
- [7] M. Saleem and B. Kovari, "Online signature verification based on signer dependent sampling frequency and dynamic time warping," in *2020 7th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, IEEE, Nov. 2020, pp. 182–186. doi: 10.1109/ISCMI51676.2020.9311604.
- [8] N. H. Al-banhawy, H. Mohsen, and N. Ghali, "Signature identification and verification systems: a comparative study on the online and offline techniques," *Future Computing and Informatics Journal*, vol. 5, no. 1, pp. 28–45, Dec. 2020, doi: 10.54623/fue.fcij.5.1.3.
- [9] Y. Zhou, J. Zheng, H. Hu, and Y. Wang, "Handwritten signature verification method based on improved combined features," *Applied Sciences*, vol. 11, no. 13, p. 5867, Jun. 2021, doi: 10.3390/app11135867.
- [10] H. Nemmour and Y. Chibani, "Off-line signature verification using artificial immune recognition system," in *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, Nov. 2013, pp. 164–167. doi: 10.1109/ICECCO.2013.6718254.
- [11] D. Banerjee, B. Chatterjee, P. Bhowal, T. Bhattacharyya, S. Malakar, and R. Sarkar, "A new wrapper feature selection method for language-invariant offline signature verification," *Expert Systems with Applications*, vol. 186, p. 115756, Dec. 2021, doi: 10.1016/j.eswa.2021.115756.
- [12] V. Ruiz, I. Linares, A. Sanchez, and J. F. Velez, "Off-line handwritten signature verification using compositional synthetic generation of signatures and Siamese neural networks," *Neurocomputing*, vol. 374, pp. 30–41, Jan. 2020, doi: 10.1016/j.neucom.2019.09.041.
- [13] S. Tayeb *et al.*, "Toward data quality analytics in signature verification using a convolutional neural network," in *2017 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2017, pp. 2644–2651. doi: 10.1109/BigData.2017.8258225.
- [14] I. Hadjadj, A. Gattal, C. Djeddi, M. Ayad, I. Siddiqi, and F. Abass, "Offline signature verification using textural descriptors," 2019, pp. 177–188. doi: 10.1007/978-3-030-31321-0\_16.
- [15] S. M. A. Navid, S. H. Priya, N. H. Khandakar, Z. Ferdous, and A. B. Haque, "Signature verification using convolutional neural network," in *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, IEEE, Nov. 2019, pp. 35–39. doi: 10.1109/RAAICON48939.2019.19.
- [16] A. K. Bhunia, A. Alaei, and P. P. Roy, "Signature verification approach using fusion of hybrid texture features," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8737–8748, Dec. 2019, doi: 10.1007/s00521-019-04220-x.
- [17] M. Liwicki *et al.*, "Signature verification competition for online and offline skilled forgeries (SigComp2011)," in *2011 International Conference on Document Analysis and Recognition*, IEEE, Sep. 2011, pp. 1480–1484. doi: 10.1109/ICDAR.2011.294.
- [18] M. A. Shaikh, T. Duan, M. Chauhan, and S. N. Srihari, "Attention based writer independent verification," in *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, IEEE, Sep. 2020, pp. 373–379. doi: 10.1109/ICFHR2020.2020.00074.
- [19] S. Rajendran, R. Dorothy, R. M. Joany, R. J. Rathish, S. Santhana Prabha, and S. Rajendran, "Image enhancement by Histogram equalization," *Int. J. Nano. Corr. Sci. Engg.*, vol. 2, no. 4, pp. 21–30, 2015, [Online]. Available: <https://www.researchgate.net/publication/283727396>
- [20] P. Singhal, A. Verma, and A. Garg, "A study in finding effectiveness of Gaussian blur filter over bilateral filter in natural scenes for graph based image segmentation," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, Jan. 2017, pp. 1–6. doi: 10.1109/ICACCS.2017.8014612.
- [21] B. K. Triwijoyo and A. Adil, "Analysis of medical image resizing using bicubic interpolation algorithm," *Jurnal Ilmu Komputer*, vol. 14, no. 1, p. 20, Apr. 2021, doi: 10.24843/JIK.2021.v14.i01.p03.
- [22] M. Malaisamy, "Principal component analysis based feature vector extraction," *Indian Journal of Science and Technology*, vol. 8, no. 35, Dec. 2015, doi: 10.17485/ijst/2015/v8i35/77760.
- [23] A. Wirdiani, T. Lattifia, I. K. Supadma, B. J. K. Mahar, D. A. N. Taradhita, and A. Fahmi, "Real-time face recognition with eigenface method," *International Journal of Image, Graphics and Signal Processing*, vol. 11, no. 11, pp. 1–9, Nov. 2019, doi: 10.5815/ijgisp.2019.11.01.
- [24] E. A. Khorsheed and Z. A. Nayef, "Face recognition algorithms: a review," *Academic Journal of Nawroz University*, vol. 11, no. 3, pp. 202–207, Aug. 2022, doi: 10.25007/ajnu.v11n3a1432.
- [25] B. Sebastian, A. Unnikrishnan, and K. Balakrishnan, "Gray level co-occurrence matrices: generalisation and some new features," May 2012, [Online]. Available: <http://arxiv.org/abs/1205.4831>
- [26] S. Bakheet and A. Al-Hamadi, "Automatic detection of COVID-19 using pruned GLCM-Based texture features and LDCRF classification," *Computers in Biology and Medicine*, vol. 137, p. 104781, Oct. 2021, doi: 10.1016/j.compbiomed.2021.104781.
- [27] K. Harrar, K. Messaoudene, and M. Ammar, "Combining GLCM with LBP features for knee osteoarthritis prediction: data from the




- Osteoarthritis initiative," *ICST Transactions on Scalable Information Systems*, p. 171550, Jul. 2018, doi: 10.4108/eai.20-10-2021.171550.
- [28] H. A. Dwaich and H. A. Abdulbaqi, "Signature texture features extraction using GLCM approach in Android studio," *Journal of Physics: Conference Series*, vol. 1804, no. 1, p. 012043, Feb. 2021, doi: 10.1088/1742-6596/1804/1/012043.
- [29] N. Iqbal, R. Mumtaz, U. Shafi, and S. M. H. Zaidi, "Gray level co-occurrence matrix (GLCM) texture based crop classification using low altitude remote sensing platforms," *PeerJ Computer Science*, vol. 7, p. e536, May 2021, doi: 10.7717/peerj-cs.536.
- [30] A. K. Aggarwal, "Learning texture features from GLCM for classification of brain tumor MRI images using random forest classifier," *WSEAS Transactions on Signal Processing*, vol. 18, pp. 60–63, Apr. 2022, doi: 10.37394/232014.2022.18.8.
- [31] D. Yanqing, Y. Guoqing, and Z. Yanjie, "Remote sensing image content retrieval based on frequency spectral energy," *Procedia Computer Science*, vol. 107, pp. 448–453, 2017, doi: 10.1016/j.procs.2017.03.088.
- [32] B. Cozzens *et al.*, "Signature verification using a convolutional neural network," 2018.
- [33] J. Ruiz-del-Solar, C. Devia, P. Loncomilla, and F. Concha, "Offline signature verification using local interest points and descriptors," 2008, pp. 22–29. doi: 10.1007/978-3-540-85920-8\_3.
- [34] G. Alvarez, M. Bryant, B. Sheffer, and M. Bryant, "Offline signature verification with convolutional neural networks," *Technical report, Stanford University*, p. 8, 2016.
- [35] H.-H. Kao and C.-Y. Wen, "An offline signature verification and forgery detection method based on a single known sample and an explainable deep learning approach," *Applied Sciences*, vol. 10, no. 11, p. 3716, May 2020, doi: 10.3390/app10113716.
- [36] N. Çalik, O. C. Kurban, A. R. Yilmaz, T. Yildirim, and L. Durak Ata, "Large-scale offline signature recognition via deep neural networks and feature embedding," *Neurocomputing*, vol. 359, pp. 1–14, Sep. 2019, doi: 10.1016/j.neucom.2019.03.027.
- [37] M. Jampour and A. Naserasadi, "Chaos game theory and its application for offline signature identification," *IET Biometrics*, vol. 8, no. 5, pp. 316–324, Sep. 2019, doi: 10.1049/iet-bmt.2018.5188.
- [38] L. Liu, L. Huang, F. Yin, and Y. Chen, "Offline signature verification using a region based deep metric learning network," *Pattern Recognition*, vol. 118, p. 108009, Oct. 2021, doi: 10.1016/j.patcog.2021.108009.
- [39] T. M. Maruyama, L. S. Oliveira, A. S. Britto, and R. Sabourin, "Intrapersonal parameter optimization for offline handwritten signature augmentation," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1335–1350, 2021, doi: 10.1109/TIFS.2020.3033442.
- [40] F. M. Alsuhimat and F. S. Mohamad, "Offline signature verification using long short-term memory and histogram orientation gradient," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 283–292, Feb. 2023, doi: 10.11591/eei.v12i1.4024.
- [41] M. S. Kadhmi, M. J. Mohammed, and H. Ayad, "An accurate signature verification system based on proposed HSC approach and ANN architecture," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, p. 215, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp215-223.

## BIOGRAPHIES OF AUTHORS






**Zainab Hashim**    received the B.S.c degree in computer science from University of Technology, Baghdad, Iraq, in 2016, the M.Sc. degree in computer/artificial intelligence from University of Technology, Baghdad, Iraq, in 2019. She is currently a PhD student in computer science from University of Technology. Her research interests include artificial intelligence, machine learning, and biometrics. She can be contacted at email: cs.20.16@grad.uotechnology.edu.iq.



**Hanaa Mohsin**    Assistant Professor Dr. Hanaa Moshin Ahmed Salman. Obtained her M.Sc. and Ph.D. from the University of Technology, Baghdad, Iraq, in 2002 and 2006, respectively. Currently she is a lecturer in Computer Science and a member of the Scientific Committee and Promotion Committee in the Department of Computer Science. Dr. Hanaa has more than 23 years of experience and she has supervised graduate students. Her primary research interests include cryptography, computer security, biometrics, image processing, and computer graphics. She can be contacted at email: 110113@uotechnology.edu.iq.



**Ahmed Alkhayyat**    received the B.Sc. degree in electrical engineering from AL KUFA University, Najaf, Iraq, in 2007, the M.Sc. degree from the Dehradun Institute of Technology, Dehradun, India, in 2010, and Ph.D. from Cankaya University, Ankara, Turkey, in 2015. He contributed in organizing a several IEEE conferences, workshop, and special sessions. He is currently a dean of international relationship and manager of the word ranking in the Islamic university, Najaf, Iraq. To serve my community, I acted as a reviewer for several journals and conferences. His research interests include IoT in the health-care system, SDN, network coding, cognitive radio, efficient-energy routing algorithms and efficient-energy MAC protocol in cooperative wireless networks and wireless body area network, as well as cross-layer designing for self-organized network. He can be contacted at email: ahmedalkhayyat85@gmail.com.