IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 February 2023, accepted 23 February 2023, date of publication 3 March 2023, date of current version 8 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3252022

RESEARCH ARTICLE

A Hybrid Method of Feature Extraction for Signatures Verification Using CNN and HOG a Multi-Classification Approach

FADI MOHAMMAD ALSUHIMAT AND FATMA SUSILAWATI MOHAMAD
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Kuala Terengganu 21300, Malaysia
Corresponding author: Fadi Mohammad Alsuhimat (karaklove@yahoo.com)

ABSTRACT The offline signature verification system's feature extraction stage is regarded as crucial and has a significant impact on how well these systems perform because the quantity and calibration of the features that are extracted determine how well these systems can distinguish between authentic and fake signatures. In this study, we introduced a hybrid method for extracting features from signature images, wherein a Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG) were used, followed by the feature selection algorithm (Decision Trees) to identify the key features. Finally, the CNN and HOG methods were combined. Three classifiers were employed to evaluate the efficacy of the hybrid method (long short-term memory, support vector machine, and K-nearest Neighbor). The experimental findings indicated that our suggested model executed satisfactorily in terms of efficiency and predictive ability, with accuracies of (95.4%, 95.2%, and 92.7%) the UTSig dataset, and (93.7%, 94.1%, and 91.3%, respectively) with the CEDAR dataset. This accuracy is deemed to be of high significance, particularly given that we checked skilled forged signatures that are more difficult to recognize than other forms of forged signatures like (simple or opposite).

INDEX TERMS Offline signature verification, CNN, HOG, deep learning.

I. INTRODUCTION

Biometrics represents the most important technological method used to identify people and determine their power through the behavioral and physiological characteristics of individuals. Measurements of biological traits, such as ears, fingerprints, iris, and DNA, are used to make identifications in the physiological category, while expression, voice, gait, and signature are used to identify persons based on the behavioral category. The handwritten signature is one of the most accepted methods of biometric verification in the world [1].

Banks, credit cards, passports, check processing, and financial documents use handwritten signatures as unique behavioral biometrics. It is difficult to verify these signatures, particularly when they are unclear. Therefore, a system that can distinguish between a genuine signature and a fake signature is required to lower the chance of theft or fraud. In the past thirty years, several studies have been conducted in this field, from traditional verification based on expert opinions to machine learning algorithms, then deep learning algorithms today, despite all these studies, offline signature verification systems still need a lot of development and improvement [2].

There are two methods for automating signature verification: online [3, 4, 5, 6, 7] and offline [8, 9, 10, 11, 12, 13]. According to previous studies [1, 2, 8, 10, 11], offline signature verification is regarded as more challenging than online verification because variables such as pen-tip pressure, velocity, and acceleration are not available when employing offline signature images. Moreover, the unique procedures for obtaining signatures render the online technique inappropriate in practice in several situations.

Although signature verification is considered the most widely accepted and least extreme biometric method in

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

society compared to other biometric methods, many previous studies [12], [13], [14], [15] have indicated that signature verification is not easy, given that handwriting signatures contain special letters and symbols, which are often unreadable and signer behaviors are dissimilar. Therefore, it is important to analyze the signature as one image without analyzing it as letters or words independently, and focus on building an effective signature system that relies on a real-life situation.

Offline signature verification seeks to discover forged signatures to reduce the danger of hacking and crime [14]. In addition, the techniques for checking signatures help automatically distinguish between real and fake signatures by assessing whether the signature used in the inquiry is real or fraudulent. Although there are many systems for verifying signatures offline, these systems have difficulty distinguishing between real and forged signatures because of the diversity in the degree of forged signatures, as there are three types of forged signatures: random forgery, unskilled forgery/simple forgery, and skilled forgery [15].

For the first two forms, the forged signature is made either haphazardly or without knowledge of the name or signature design. The imitator is presumed to be an expert in recreating the form and style of the signature, and is aware of the genuine signature style in the case of competent forgeries. Without dynamic components, skilled forgery detection is undoubtedly more difficult [16].

The offline signature verification system includes two main phases: feature extraction and classification. Many studies [17], [18] have indicated that the feature extraction phase affects the signature verification system performance, given its importance and its main role in facilitating the work of works to distinguish between original signatures and forged signatures, where providing important features contributes to improving and developing the ability of signature verification systems and increasing their degree of accuracy.

Despite this, the feature extraction stage still faces many issues, such as the quality and number of features extracted from signature images, which has been confirmed by previous studies [17], [19], [20], [21], indicating the need to improve the feature extraction stage to reach a set of features capable of assisting classifiers in verifying signatures.

Based on the discussion, the current study seeks to improve the feature extraction stage in offline signature verification by developing a hybrid method for extracting features from signature images and classifying them for authentic and forged signatures using deep learning and machine learning classifiers to ensure that the hybrid method can improve the performance of various classifiers.

II. CONTRIBUTIONS AND NOVELTY

The use of offline handwritten signatures as a popular technique for biometric human identification has increased over the past 10 years. Despite the significance of this technology, it is not a simple undertaking; the difficulty in such a system arises from the inability of any person to sign the same signature every time. In this work, we are interested in the dataset's features that could affect the model's performance by identifying the most important features in the signature image using Histogram Orientation Gradient (HOG) and Convolutional Neural Network (CNN). Thus, the output of the feature extraction process includes a specific set of features capable of contributing to improving the performance of the signature classification process and identifying the forger among them.

The key contributions and novelty of this work include:

1. Develop a hybrid method for feature extraction by combining HOG and CNN algorithms for signature verification. To overcome and go beyond the feature extraction limitations, this study proposes the use of the HOG feature extraction algorithm with a specific cell size for a number of extracted features, which can be suitable for the classification process. In this hybrid model, we believe that we can further improve the verification precision while avoiding adjusting the system parameters for each writer. Therefore, (HOG-CNN) allows more flexible training and, thus, a verification with many writers.
2. The hybrid model has a robust feature set and may work in conjunction with a low-complexity classifier to improve performance.
3. The use of three classifiers from machine learning and deep learning will contribute to confirming the importance of the hybrid method adopted in extracting features, given that the evolution of the performance of the three classifiers will ensure that the hybrid method prepared in this study can improve the performance of different machine learning and deep learning classifiers.

III. RELATED WORK AND MOTIVATION

The collection of features used for the verification model determines its performance. There has been a significant amount of work on offline signature verification, which uses several feature sets to operate the model. Topology, geometric data, gradient, structural data, and concavity bases are the features found in the majority of the works [9], [11], [22], [23], [24]. For instance, an approach using a collection of geometric properties described in the specification of the signature envelope and the patterns of strokes was presented by Ferrer et al. [9]. The hidden Markov model, support vector machines, and Euclidean distance classifier are then used in the verification process.

With the aim of improving signature verification offline a variety of systems have been developed, A technique for authenticating signatures was recently devised by ZulNarnain et al. [25] based on the side, angle, and perimeter of triangles that are formed after triangulating a signature image. They used a voting-based classifier in addition to a Euclidean classifier to categorize the data. Studies based on characteristics related to curvature [10], directional run of pixels [12], [26], [27], pixel surroundings [28], gray value distribution [29], [30], [31], and pixel surroundings have been

published. Further publications in the literature use graphometric features [32]. To examine upper and lower signature envelopes, the authors of [33] introduced a shape property termed the chord moment. Chord moment-based characteristics were combined with a support vector machine (SVM) for signature verification.

Multiple features have frequently been combined to increase the classification accuracy of the models. For instance, minute information and a grey value distribution were employed in [12], together with the directional characteristic. The distribution of pixels in the thinned signature strokes was utilized by the authors to create a 16-directional feature. The feature extraction process is expensive because many distinct types of characteristics are involved. With the model being utilized for real-time applications, it is evident that computing moment information coupled with the 16-directional feature is computationally expensive.

Serdouk et al. [13] addressed feature extraction methods outside directed distribution. The longest runs were considered in the horizontal, vertical, and two primary diagonal directions. In this case, a directional distribution-based longest-run feature is paired with gradient local binary patterns to expand the feature set (GLBP). Consequently, they combined topological and gradient properties. The topological property chosen was the length of the longest run of the pixels. Using a Local Binary Pattern (GLBP) in the neighborhood, the gradient information is collected. It can be expensive to compute the GLBP for each pixel in the signature image. They proposed an Artificial Immune Recognition System-based Verification System. A template-based verification method was also proposed [34]. They proposed a method based on encoding the geometric structures of signatures using grid templates.

Subramaniam et al. [35] used a CNN to improve signature forgery detection, indicating that CNN is more accurate and faster in the detection of forged signatures. Kumar et al. [36] used CNN to enhance signature verification and mentioned that using CNN obtained leading performance with a 3.56 average error rate (AER) on the GPDS synthetic, 4.15 on CEDAR, and 3.51 on MCYT-75 datasets, respectively.

Jindal et al. [37] used two machine learning algorithms, support vector machine and decision tree, to verify the signature and indicated that both algorithms achieved good results compared to previous works. In addition, Jagtap et al. [38] used a CNN to improve signature verification and forgery detection and confirmed the effectiveness of CNN in detecting forged signatures and developing an offline signature verification system.

Ajij et al. [2] conducted a study to enhance offline signature verification using simple combinations of border pixel directional codes, using SVM as a classifier, and found that the proposed feature set has very good results, which are supported by experimental data, and may be useful in real-time applications.

Zhou et al. [39] aimed to enhance the feature extraction phase in offline signature verification by extracting two types of features (static and dynamic), combining a gray-level co-occurrence matrix (GLCM) with HOG, and classifying the output with a support vector machine (SVM) and dynamic time warping (DTW), which indicated that improving the feature extraction phase positively affected the classification process.

Additionally, Arisoy [40] used deep learning algorithms CNN and Siamese Network to improve signature verification, and the proposed method was implemented on four datasets (4NSigComp2012, SigComp2011, 4NSigComp2010, and BHsig260) and achieved accuracies of 93.23, 90.11, 89.99, and 92.35, respectively. In addition, Tahir et al. [16] used artificial neural networks to improve offline signature verification systems, and the proposed method used many features, including Baseline Slant Angle (BSA), Aspect Ratio (AR), and Normalized Area (NA), and achieved an accuracy of 82.5. Table 1 lists some methods used in existing studies on offline signature verification.

Based on previous research, there is still a need to develop an offline signature verification system, as there is a need to improve the dataset quantity and quality to train the model, use more types of algorithms when training the model while including features extracted in different scenarios, and enhance the feature extraction phase considering its significant impact on the performance of the classification stage.

Despite the development of several techniques and recognition models, the outcomes of these techniques confirm that there is still much room for improvement in accuracy and robustness. It also has the potential to provide a strong feature set that can enhance the performance of a classifier with minimal complexity. It would be advantageous if the feature set could be readily determined from signature images.

Through the integration of two feature extraction methods, a Convolutional Neural Network (CNN) and Histogram of Oriented Gradients (HOG), we have suggested a fresh set of features in this study. This new vector has a robust feature set, and may work in conjunction with a low-complexity classifier to become a better performer.

IV. PROPOSED METHOD

The feature extraction method and classification algorithms utilized for the signature verification system are briefly described in this section. The following are the two feature extraction techniques and three classifiers that constitute the recommended signature classification algorithm.

A. HOG ALGORITHM

In this study, features from the signature images were extracted using the HOG approach. Trait shape representation, first discussed by Dalal and Triggs [45] at the CVPR conference in 2005, was implemented using HOG. HOG, or Histograms of Oriented Gradients, are mostly employed as person detectors. In this study, HOG was used both alone and in conjunction with the CNN method as a feature extraction approach to detect and recognize signature pictures.

The HOG descriptor approach hypothetically records events of angle introduction in specific areas of a photograph

TABLE 1. Features and classifiers for the existing methods.

Source	Features	Classifiers
Kovari & Charaf [23]	Height, width, area, etc.	Probabilistic model
Jiang et al. [30]	GLBP and Improved GLBP	SVM
Kumar & Puhan [33]	Upper and lower Envelope; chord moments	SVM
Guerbai et al. [10]	Curvelet transform	RBF-SVM, MLP
Pham et al. [24]	Geometry-based features.	Likelihood ratio
Serdouk et al. [13]	Gradient Local Binary Patterns (GLBP) and LRF	k-NN
Pal et al. [41]	Uniform Local Binary Patterns (ULBP)	Nearest Neighbor
Loka et al. [42]	Long range correlation (LRC)	(LRC) SVM
Zois et al. [43]	Lattice arrangements Pixel distribution	Decision tree
Muhammad et al. [19]	Local pixel distribution GA	SVM
Batool et al. [14]	GLCM, geometric features	SVM
Ajij et al. [2]	Quasi-straight line segments	SVM
Tahir et al. [16]	Baseline Slant Angle, Aspect Ratio, Normalized Area, Center of Gravity, and line's Slope.	Artificial Neural Network
Lopes et al. [44]	Straight lines, edges, and objects.	CNN
Proposed	Magnitude, angle of the gradient, orientations of the gradient, edges, colour data, parts of shapes and lines	LSTM, SVM, and KNN

or a location of interest (ROI). The following is the main use of the HOG descriptor, as shown in Figure 1 A histogram of the angle directions or edge orientations for the pixels inside each area is generated after the image has been segmented into tiny, related regions (cells). The resulting gradient orientation was then employed. After discretizing each cell into precise containers, nearby cells were gathered into groups inside the spatial region. At this point, each cell's pixel provides a weighted angle to its precise canister. Finally, the normalized gather of histograms talks to the piece histogram, and the set of these square histograms speaks to the descriptor, which forms the basis for histogram gathering and normalizing [46].

This study defines HOG as having a [4 × 4] pixel block size. The result is that each signature picture sample's feature

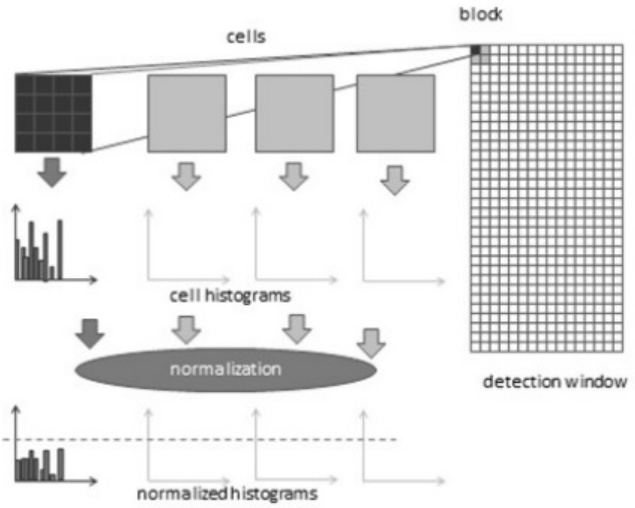

FIGURE 1. Demonstrates the HOG algorithm implementation.

vector is 34596 bytes long overall. To better elucidate the HOG implementation of offline signatures, When the cell size is small, the number of shown gradients and orientations often exists more clearly than when the cell size is large, according to Singh et al. [47], becoming noticeable compared with the large cell size. When the number of cells in the HOG parameter increased, the directions and gradient gradually decreased. The effects of HOG on offline signature images are shown in Figure 2, with cells ranging in size from 2 to 16.

B. CNN ALGORITHM

In this study, offline signature verification was performed using a deep learning technique. A Convolutional Nesural Network (CNN) ad hoc model was employed as a deep learning technique. Krizhevsky et al. [48] originally presented a convolutional neural network as a technique for processing images, which included two key components: spatial pooling and spatially shared weights.

CNNs are currently considered the most widely used deep learning architecture in feature learning because of their numerous successful applications in a variety of fields, including autonomous vehicles [49], character recognition [50], video processing [51], medical image processing, and object recognition [45]. Figure 3 shows the core structure of the CNN.

Figure 3 illustrates the three basic layers of a CNN: the convolutional layer, subsampling layer (sometimes called the pooling layer), and fully connected layer. CNN use convolutional and pooling approaches to recognize the distinguishing features of images. While the characteristics obtained in the latter layers depict portions of forms and objects, those obtained in the early stages are identified as edges or color information [42].

The convolution process is performed at the convolution layer by shifting the filter data matrix onto the input data matrix and adding a bias to the multiplication of these

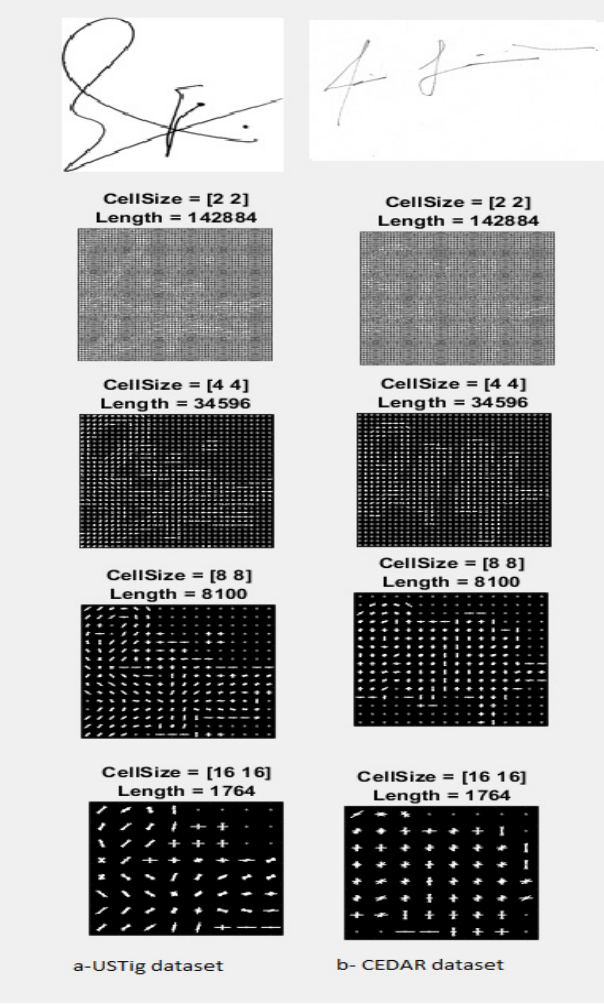

FIGURE 2. HOG implementation on signature images with different cell size.

matrices. Figure 4 shows the basic convolution approach. Equation provides the fundamental formulation of convolution process (1). The pixels from the input picture, filter (kernel), bias term, and output image are represented by y, x, w, and b, respectively.

$$y_n = \sum_{n=1}^{9} (x_n.w_n + b_0) \tag{1}$$

Pooling is another tool used by the CNNs. By propagating the highest activation of the preceding neuron groups, the pooling tool [50] is utilized to spatially down-sample the activation of the prior layer. The primary goal of pooling layers is to gradually reduce the dimensionality of the representation, which lowers the computational complexity [49]. At the end of each layer, normalization can be performed upon request using a rectified linear unit (ReLU) activation function. Equation (2) represents the fundamental (ReLU) method.

$$\text{ReLU(x)} = f\,(x) = \begin{cases} 0 & if\ x < 0 \\ x & if\ x \geq 0 \end{cases} \tag{2}$$

Fully Connected Layers (FC), the foundational components of conventional neural networks, constitute the last layer of

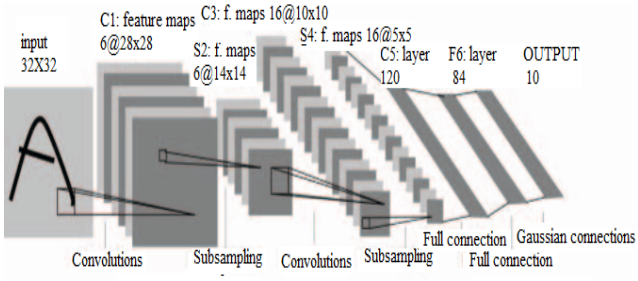

FIGURE 3. Basic structure of CNN.

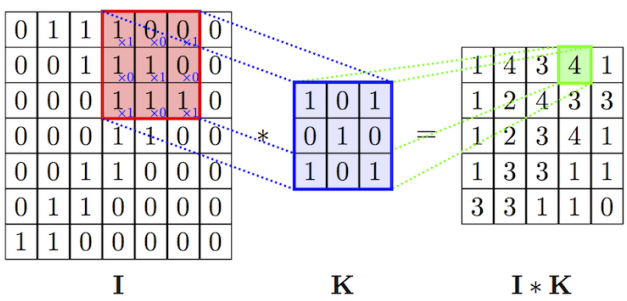

FIGURE 4. Basic convolution operation.

the CNN. Fully linked layers are shaped by how the neurons in one layer interact with each other. At that point, it is normalized to the probability dispersion using a softmax layer. Additionally, this implies that to convert the photos into votes, high-level filters are required. Weights or connections between each vote and category are used to express these votes [49], [51].

C. FEATURES SELECTION AND INTEGRATION PROCESS

As previously mentioned, the number and quality of features extracted from the signature image have a significant impact on the performance of signature recognition systems. For example, if there are insufficient features or the quality of the features is low, the accuracy of the signature recognition system will decrease. Therefore, reducing the number of features used in a statistical analysis may have various advantages, including increased model explainability, less risk of overfitting, faster training, greater data visualization, and accuracy gains.

In reality, it has been demonstrated statistically that there are an ideal number of features that should be employed for each individual job while performing a machine learning assignment (Figure 4). The model performance will simply decline if more features than are actually necessary are introduced (because of the added noise). Finding the right number of features to employ is a true issue; this depends on the amount of data we have at our disposal and the difficulty of the task we are attempting to complete.

Several approaches can be used for feature selection. The most significant is [52].

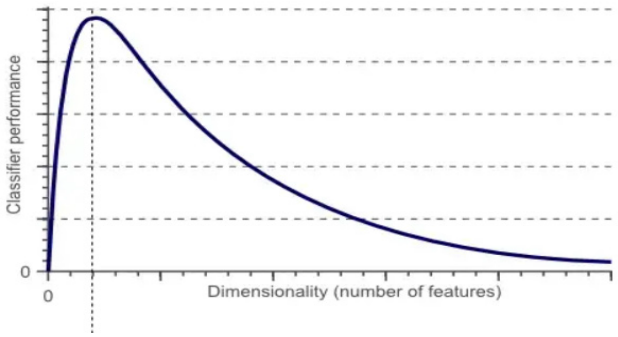

FIGURE 5. Dimensionality and classifier performance relationship.

Filtering our dataset to only include a portion of it that includes all the necessary features (for example, correlation matrix using Pearson's correlation coefficient).

Wrapper Technique: employs a machine learning model as its criterion for assessment while adhering to the same goal as the filter method (such as Forward, Backward, Bidirectional, and Recursive Feature Elimination).

The embedded technique uses a machine-learning model similar to the filter method. The Embedded Method analyzes several training iterations of our ML model and rates the significance of each feature depending on how much each feature contributes to the ML model training, which is the difference between the two approaches (for example,. LASSO Regularization).

We recognized the critical features, made use of them, and dropped optional ones to carry out the feature selection procedure. A decision tree was used to identify the key attributes. To rate the significance of the various attributes, decision tree models that are based on ensembles (such as extra trees and random forests) might be utilized. Understanding how our model makes predictions and how to explain them better depends on understanding which attributes our model values most highly. The characteristics that do not aid our model in any way or cause it to make a mistake may be removed at the same time.

Detailed process of decision trees is described as follows (Algorithm 1).

Algorithm 1 Features Selection by Decision Trees

Input: Set of Features
Output: Importance Features
Step1: Calculate Time
Step2: trained forest by trained random forest classifier
Step3: create a feature importance plot
Step4: make predictions
Step5: draw features figure
Step6: determine the most important
Step7: use the importance features

These steps were implemented on the results of the (HOG) method and the results of the (CNN) method. The following table lists the number of features extracted according to each method and the number of important features.

TABLE 2. Number of features with and without feature selection.

Method	Number of Features
HOG method	34596
CNN method	512
Feature Selection / HOG	22916
Features Selection / CNN	402

After completing the process of identifying the important features, merging between the HOG and CNN methods was performed, and the algorithm explained the steps of the merging process.

Algorithm 2 HOG Combined With CNN

Input: HOG Features, CNN Features
Output: Integrated Features
Step1: Getting the number and type for each features set
Step2: Convert the features matrix to the same type (N ∗ Features value), where N is features number
Step3: create two empty matrixes
Step4: save features result for each method in different matrix
Step5: combine both matrixes

The following table shows the final number of extracted features.

In the HOG algorithm the feature vector for signature images, where each image has labeled as genuine or forgery, and each image have these types of features, magnitude, angle of the gradient, and orientations of the gradient, features total in this section were (34596) features. In the CNN algorithm, the feature vector for signature images, where each image is labeled as genuine or forgery, and each image has these types of features, edges, color data, parts of shapes and lines, and features in this section are (512) features. In the proposed method, the feature vector for signature images, where each image is labeled as genuine or forgery, and each image has these types of features: magnitude, angle of the gradient, orientations of the gradient, edges, color data, parts of shapes and lines, and features total in this section were (23318) features.

Based on the proposed method, the number of significant features was improved for each signature image by selecting the most important features extracted from the HOG and CNN. Therefore, in this case, the feature vector will contain only the significant features that positively affect the performance of the classification process.

D. SIGNATURE CLASSIFICATION

In this study, we classified data using a variety of techniques, including long short-term memory, support vector machines, and K-nearest Neighbor.

This method of obtaining parameters is based on the most accurate assessments of a variety of inner characteristics [53].

TABLE 3. Number of features for each method.

Method	Number of Features
HOG method	34596
CNN method	512
Integrated method	23318

One of the most common and straightforward classification algorithms is KNN. The learning strategy coupled internal gathering activities while sparing distinctive vectors and markers from the learning pictures.

This unmarked location might be assigned the designation of its k-nearest neighbors. By using overwhelming part surveys, this entity is often characterized based on the characteristics of its k-nearest neighbors. According to the power of the parameter closest to it, the parameters are grouped as k=1. Only two segments are required in this case; therefore, k must be an odd number. K can appear in a multiclass configuration as odd numbers. As a related point separation capability for KNN, this step uses the renowned distance equation (3), Euclidean distance, after converting each image to a vector and dropping the fixed length for true numbers claim:

$$\boldsymbol{d(x, y) = (\sum_{i=1}^{m} ((x_i - y_i)^2))^{1/2}} \tag{3}$$

Support Vector Machine (SVM): This is used to evaluate various signature characteristics [54]. During the training process, we built a signature classifier using all the preparation data by applying a classification algorithm to specific characteristics for signature photos. A signature prediction outline uses training processes and an SVM algorithm to categorize the input signature picture. The characteristic vectors are the inputs xi [55]: We used the Gaussian kernel K to set the SVM parameters.

$$f(x) = \sum_{i=1}^{N_s} a_i y_i K(s_i, x) + b$$
$$K(x_i, x_j) = e^{\frac{1}{2\sigma^2}|x_i - x_j|^2} \tag{4}$$

Long Short-Term Memory (LSTM): A deep learning artificial recurrent neural network (RNN) structure consists of long short-term memory (LSTM). In addition, LSTM integrates feedback connections in contrast to standard feed-forward neural networks, allowing it to generate both discrete informational units (such as pictures) and complete informative arrangements (such as speech or video) [56].

The Long Short-Term Memory (LSTM) structure for recurrent neural networks has been advocated since it was first introduced in 1995. These methods have been developed into cutting-edge models for several machine learning problems over time. This has sparked attention once again in determining the value and function of different computational elements in typical LSTM versions [57].

The long-term conditions are intended to be preserved while designing RNN nodes with LSTMs. They consist of a self-connected memory cell that resembles a conventional RNN node and three gates that control the hub yield and input. An LSTM hub input may be a sigmoid function for each gate. An input door is the main door that controls whether fresh input for the hub is available. The hub can modify the memory cell's activation values by using the moment door as a disregard entry. The last entrance, known as the yield entryway, controls which parts of the cell yield are made available to the other nodes [58]. Figure 6 depicts the organization of Long Short-Term Memory.

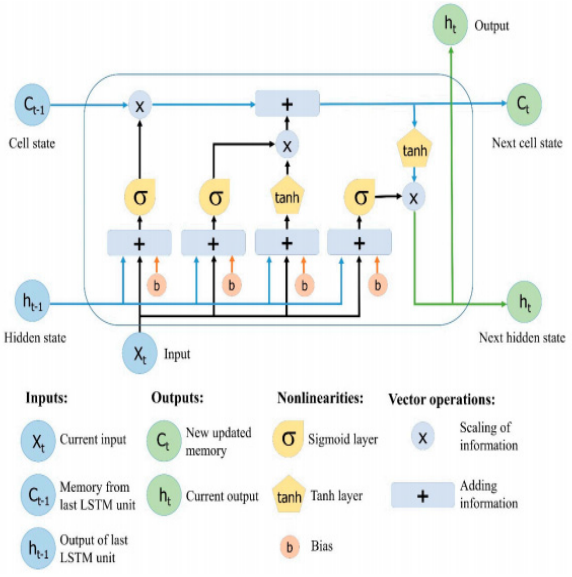

FIGURE 6. The structure of the long short-term memory (LSTM) neural network. Reproduced from [59].

E. DATASET

The (UTSig) and CEDAR datasets were used in the experimental procedure. The UTSig dataset has (115) classifications, including (27) authentic signatures, (3) opposite-hand forgeries, (36) easy forgeries, and (6) skill forgeries, as illustrated in Figure 7. Each class has a specific real person assigned to it. Students from the University of Tehran and Sharif University of Technology who signed up for UTSig had their signatures scanned at a resolution of 600 dpi and saved as 8-bit Tiff files [60].

A total of (1350) signature photos from the UTSig dataset were utilized in this study to train a set that included 50 real signatures and each of the six expertly forged signatures. We favor expert forgeries because they are more challenging to detect than other forgery categories, and we tested our classification method using (300) signature photos.

The signatures of 55 signers from various professional and cultural backgrounds were included in the CEDAR data collection. Each of these signers verified 24 documents at intervals of 20 minutes. To create 24 fake signatures for each real signer, each forger made an eight-time effort to copy the signatures of three different signers. Consequently, there were 1,320 legitimate signatures and 1,320 fake signatures in the dataset (55 24) [20].

In this study, we trained our classification algorithms on 1200 signature shots from the CEDAR dataset and tested

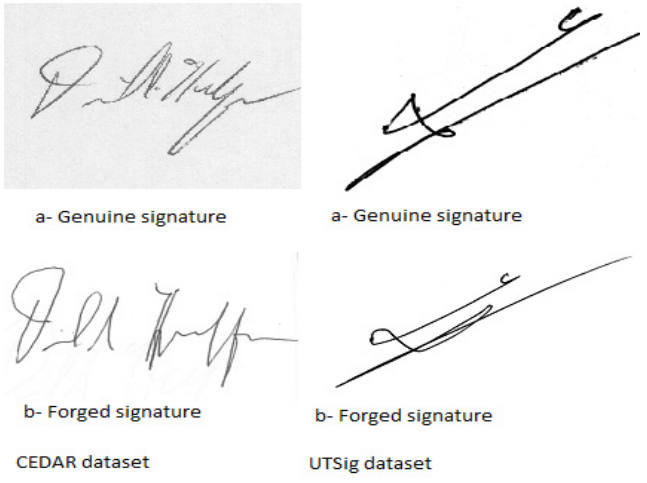

FIGURE 7. Forger and Genuine signature examples from UTSig and CEDAR dataset.

TABLE 4. The differences between training and testing sets.

Sets	UTSig	CEDAR
Training	1350	1200
Test	300	400
Total	1650	1600

TABLE 5. The values of Run-Time and accuracy for each method.

Method	Run-Time		Accuracy	
	UTSig dataset	CEDAR dataset	UTSig dataset	CEDAR dataset
LSTM-HOG	1.67	2.98	92.4%	87.7%
LSTM-CNN	1.41	2.67	93.1%	89.2%
LSTM-proposed	**1.54**	**2.73**	**95.4%**	**93.7%**
SVM-HOG	266.9	172.8	91.8%	92.5%
SVM-CNN	70.1	62.3	93.4%	93.7%
SVM-proposed	**173.1**	**109.6**	**95.2%**	**94.1%**
KNN-HOG	7.47	5.32	86.7%	87.2%
KNN-CNN	1.89	1.72	88.4%	89.1%
KNN-proposed	**3.51**	**2.34**	**92.7%**	**91.3%**

them on 400 signature images. Figure 7 depicts examples of both phoney and authentic signatures.

In this study, the original and forged signatures of the first 50 participants were selected from the UTSig database, and the original signatures and eight fake signatures for each participant were selected from the CEDAR database. table displays the many signature photos taken from each dataset.

TABLE 6. Results of our proposed model compare to other models.

Methods	Algorithms used	Accuracy
[61]	Convolution Neural Network, SURF, and Harris	89%
[62]	K-nearest neighbour (KNN), Support vector machine (SVM)	78.5%
[63]	Gaussian empirical rule	91.2%
[64]	probabilistic neural network	92.06%
[65]	multilayer perceptron and SVN	91.67%
Proposed System	LSTM-proposed	95.4%
	SVM-proposed	95.2%
	KNN-proposed	92.7%

TABLE 7. Results of our proposed model comparing to other models.

Methods	FAR %	FRR %	ERR %
[66]	15.08	22.76	20.94
[67]	16.1	16.2	16.5
[68]	17.25	17.26	17.25
LSTM-proposed	12.68	10.12	11.40
SVM-proposed	13.42	14.51	13.43
KNN-proposed	12.50	13.75	12.50

V. RESULTS AND DISCUSSION

The accuracy attained after each method was performed on (300) signature pictures from the UTSig dataset, and (400) signature images from the CEDAR dataset were used to measure the efficiency of each algorithm. The effectiveness of each approach was assessed based on the accuracy achieved after each technique was applied to the (300) signature images from the UTSig dataset and (400) signature images from the CEDAR dataset. Table 5 lists the accuracy and runtime of the categorization algorithms.

Table 5 summarizes the experimental findings along with the runtime and classification accuracy of each classifier. With an LSTM accuracy of 92% and a run-time of 1.67 seconds for the USTig dataset and 76% and a run-time of 20.3 seconds for the CEDAR dataset, respectively, we found that our proposed model performs well on both datasets.

Based on the accuracy results for each approach, Table 6 displays the results of the compression process between the proposed method and a few other methods for offline signature verification.

Additionally, we assessed the effectiveness of the proposed approach using the following metrics: False Acceptance Rate (FAR), False Rejection Rate (FRR), and Equal Error Rate (EER). The values at which the FAR and FRR are equivalent are used to compute the EER. The EER is the most effective and widely recognized explanation of the error rate of the verification algorithm, and the lower the EER, the less frequently the algorithm makes mistakes. It is believed that the most accurate strategy is that with the lowest ERR. Therefore, the results in Table 7 show that our approach was the most effective strategy for precisely confirming the signature attributes of offline handwritten signatures.

VI. CONCLUSION

In this work, we presented a new method for extracting features from signature images by selecting the important features in both the HOG and CNN methods, then merging the output of the two methods together, and testing the extracted features. Three classifiers (LSTM, SVM, and KNN) were used.

With an accuracy of (95.4%, 95.2%, and 92.7%, respectively) with the UTSig dataset and (93.7%, 94.1%, and 91.3%) with the CEDAR dataset, the testing findings showed that our suggested model worked well in terms of performance and predictive capacity, which is regarded as a high value, especially considering that we evaluated sophisticated forgeries, which are more difficult to spot than other kinds of forgeries, such as basic or opposite-hand forgeries, because skillful forgeries are usually very close to the original signatures.

Future signature verification performance and prediction capability are anticipated to be improved by refining the feature-extraction process.

REFERENCES

[1] F. M. Alsuhimat and F. S. Mohamad, "Offline signature verification using long short-term memory and histogram orientation gradient," *Bull. Elect. Eng. Inform.*, vol. 12, no. 1, pp. 283–292, 2023.

[2] M. Ajij, S. Pratihar, S. R. Nayak, T. Hanne, and D. S. Roy, "Off-line signature verification using elementary combinations of directional codes from boundary pixels," *Neural Comput. Appl.*, vol. 35, pp. 4939–4956, Mar. 2021, doi: 10.1007/s00521-021-05854-6.

[3] A. Q. Ansari, M. Hanmandlu, J. Kour, and A. K. Singh, "Online signature verification using segment-level fuzzy modelling," *IET Biometrics*, vol. 3, no. 3, pp. 113–127, 2014.

[4] K. Cpałka and M. Zalasiński, "On-line signature verification using vertical signature partitioning," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4170–4180, 2014.

[5] K. Cpałka, M. Zalasiński, and L. Rutkowski, "A new algorithm for identity verification based on the analysis of a handwritten dynamic signature," *Appl. Soft Comput.*, vol. 43, no. 1, pp. 47–56, Jun. 2016.

[6] E. Griechisch, M. I. Malik, and M. Liwicki, "Online signature verification based on Kolmogorov–Smirnov distribution distance," in *Proc. 14th Int. Conf. Frontiers Handwriting Recognit.*, Sep. 2014, pp. 738–742.

[7] N. Sae-Bae and N. Memon, "Online signature verification on mobile devices," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 933–947, Jun. 2014.

[8] S. Chen and S. Srihari, "A new off-line signature verification method based on graph matching," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 869–872.

[9] M. A. Ferrer, J. B. Alonso, and C. M. Travieso, "Offline geometric parameters for automatic signature verification using fixed-point arithmetic," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 6, pp. 993–997, Jun. 2005.

[10] Y. Guerbai, Y. Chibani, and B. Hadjadji, "The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters," *Pattern Recognit.*, vol. 48, no. 1, pp. 103–113, 2015.

[11] R. Larkins and M. Mayo, "Adaptive feature thresholding for off-line signature verification," in *Proc. 23rd Int. Conf. Image Vis. Comput. New Zealand*, Nov. 2008, pp. 1–6.

[12] H. Lv, W. Wang, C. Wang, and Q. Zhuo, "Off-line Chinese signature verification based on support vector machines," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2390–2399, Nov. 2005.

[13] Y. Serdouk, H. Nemmour, and Y. Chibani, "New off-line handwritten signature verification method based on artificial immune recognition system," *Expert Syst. Appl.*, vol. 51, pp. 186–194, Jun. 2016.

[14] F. E. Batool, M. Attique, M. Sharif, K. Javed, M. Nazir, A. A. Abbasi, Z. Iqbal, and N. Riaz, "Offline signature verification system: A novel technique of fusion of GLCM and geometric features using SVM," *Multimedia Tools Appl.*, pp. 1–20, Apr. 2020, doi: 10.1007/s11042-020-08851-4.

[15] F. M. Alsuhimat and F. S. Mohamad, "Histogram orientation gradient for offline signature verification via multiple classifiers," *Nveo-Natural Volatiles Essential OILS J.*, vol. 8, no. 6, pp. 3895–3903, 2021.

[16] N. M. Tahir, N. Adam, U. I. Bature, K. A. Abubakar, and I. Gambo, "Off-line handwritten signature verification system: Artificial neural network approach," *Int. J. Intell. Syst. Appl.*, vol. 1, no. 1, pp. 45–57, 2021.

[17] A. B. Jagtap, D. D. Sawat, R. S. Hegadi, and R. S. Hegadi, "Verification of genuine and forged offline signatures using Siamese neural network (SNN)," *Multimedia Tools Appl.*, vol. 79, nos. 47–48, pp. 35109–35123, Dec. 2020.

[18] B. Kiran, S. Naz, and A. Rehman, "Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities," *Multimedia Tools Appl.*, vol. 79, no. 1, pp. 289–340, 2020.

[19] M. Sharif, M. A. Khan, M. Faisal, M. Yasmin, and S. L. Fernandes, "A framework for offline signature verification system: Best features selection approach," *Pattern Recognit. Lett.*, vol. 139, pp. 50–59, Nov. 2020.

[20] N. Sharma, S. Gupta, and P. Metha, "A comprehensive study on offline signature verification," in *Proc. J. Phys., Conf.*, 2021, Art. no. 012044, doi: 10.1088/1742-6596/1969/1/012044.

[21] 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," *Appl. Sci.*, vol. 10, no. 1, p. 3716, 2020.

[22] M. K. Kalera, S. Srihari, and A. Xu, "Offline signature verification and identification using distance statistics," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 7, pp. 1339–1360, 2004.

[23] B. Kovari and H. Charaf, "A study on the consistency and significance of local features in off-line signature verification," *Pattern Recognit. Lett.*, vol. 34, no. 3, pp. 247–255, 2013.

[24] T.-A. Pham, H.-H. Le, and N.-T. Do, "Offline handwritten signature verification using local and global features," *Ann. Math. Artif. Intell.*, vol. 75, nos. 1–2, pp. 231–247, Oct. 2015.

[25] Z. ZulNarnain, M. S. Rahim, N. F. Ismail, and M. M. Arsad, "Triangular geometric feature for offline signature verification," *Int. J. Comput. Inf. Eng.*, vol. 10, no. 3, pp. 485–488, 2016.

[26] R. K. Bharathi and B. H. Shekar, "Off-line signature verification based on chain code histogram and support vector machine," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Aug. 2013, pp. 2063–2068.

[27] V. Nguyen, Y. Kawazoe, T. Wakabayashi, U. Pal, and M. Blumenstein, "Performance analysis of the gradient feature and the modified direction feature for off-line signature verification," in *Proc. 12th Int. Conf. Frontiers Handwriting Recognit.*, Nov. 2010, pp. 303–307.

[28] R. Kumar, J. D. Sharma, and B. Chanda, "Writer-independent off-line signature verification using surroundedness feature," *Pattern Recognit. Lett.*, vol. 33, no. 3, pp. 301–308, Feb. 2012.

[29] M. Hanmandlu, M. H. M. Yusof, and V. K. Madasu, "Off-line signature verification and forgery detection using fuzzy modeling," *Pattern Recognit.*, vol. 38, no. 3, pp. 341–356, 2005.

[30] N. Jiang, J. Xu, W. Yu, and S. Goto, "Gradient local binary patterns for human detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 978–981.

[31] J. Vargas, M. Ferrer, C. Travieso, and J. Alonso, "Off-line signature verification based on high pressure polar distribution," in *Proc. 11th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, 2008, pp. 373–378.

[32] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers," *Pattern Recognit.*, vol. 43, no. 1, pp. 387–396, Jan. 2010.

[33] M. V. M. Kumar and N. B. Puhan, "Off-line signature verification: Upper and lower envelope shape analysis using chord moments," *IET Biometrics*, vol. 3, no. 4, pp. 347–354, 2014.
[34] E. N. Zois, L. Alewijnse, and G. Economou, "Offline signature verification and quality characterization using poset-oriented grid features," *Pattern Recognit.*, vol. 54, pp. 162–177, Jun. 2016.
[35] M. Subramaniam, E. Teja, and A. Mathew, "Signature forgery detection using machine learning," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 4, no. 2, pp. 479–483, 2022.
[36] R. Kumar, M. Saraswat, D. Ather, M. N. Mumtaz Bhutta, S. Basheer, and R. N. Thakur, "Deformation adjustment with single real signature image for biometric verification using CNN," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–12, Jun. 2022, doi: 10.1155/2022/4406101.
[37] U. Jindal, S. Dalal, G. Rajesh, N. U. Sama, and N. Z. Jhanjhi, "An integrated approach on verification of signatures using multiple classifiers (SVM and decision Tree): A multi-classification approach," *Int. J. Adv. Appl. Sci.*, vol. 9, no. 1, pp. 99–109, Jan. 2022.
[38] S. Jagtap, S. Kalyankar, T. Jadhav, and A. Jarali, "Signature Verification and detection system," *Int. J. Recent Sci. Res.*, vol. 13, no. 6, pp. 1412–1418, 2022.
[39] Y. Zhou, J. Zheng, H. Hu, and Y. Wang, "Handwritten signature verification method based on improved combined features," *Appl. Sci.*, vol. 11, no. 13, p. 5867, 2021.
[40] M. Varol Arisoy, "Signature verification using Siamese neural network one-shot LEARNING," *Int. J. Eng. Innov. Res.*, pp. 248–260, Aug. 2021.
[41] S. Pal, A. Alaei, U. Pal, and M. Blumenstein, "Performance of an off-line signature verification method based on texture features on a large indic-script signature dataset," in *Proc. 12th IAPR Workshop Document Anal. Syst. (DAS)*, Apr. 2016, pp. 72–77.
[42] H. Loka, E. Zois, and G. Economou, "Long range correlation of preceded pixels relations and application to off-line signature verification," *IET Biometrics*, vol. 6, no. 2, pp. 70–78, 2017.
[43] E. N. Zois, A. Alexandridis, and G. Economou, "Writer independent offline signature verification based on asymmetric pixel relations and unrelated training-testing datasets," *Expert Syst. Appl.*, vol. 125, pp. 14–32, Jul. 2019.
[44] J. Lopes, B. Baptista, N. Lavado, and M. Mendes, "Offline handwritten signature verification using deep neural networks," *Energies*, vol. 15, p. 7611, 2022.
[45] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, United States, Jun. 2005, pp. 886–893.
[46] N. Abbas, K. Yasen, K. H. Faraj, L. Razak, and F. Malaliah, "Offline handwritten signature recognition using histogram orientation gradient and support vector machine," *J. Theor. Appl. Inf. Technol.*, vol. 96, pp. 2048–2075, 2018.
[47] S. Singh, M. Gogate, and S. Jagdale, "Signature verification using LDP & LBP with SVM classifiers," *Int. J. Sci. Eng. Sci.*, vol. 1, no. 11, pp. 95–98, 2017.
[48] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Images classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[49] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.
[50] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[51] M. F. Yahya and M. R. Arshad, "Detection of markers using deep learning for docking of autonomous underwater vehicle," in *Proc. IEEE 2nd Int. Conf. Autom. Control Intell. Syst. (I2CACIS)*, Oct. 2017, pp. 179–184.
[52] C. Boufenar, A. Kerboua, and M. Batouche, "Investigation on deep learning for off-line handwritten Arabic character recognition," *Cognit. Syst. Res.*, vol. 50, pp. 180–195, Aug. 2018.
[53] M. Li, H. Wang, L. Yang, Y. Liang, Z. Shang, and H. Wan, "Fast hybrid dimensionality reduction method for classification based on feature selection and grouped feature extraction," *Expert Syst. Appl.*, vol. 151, Jul. 2020, Art. no. 113277.
[54] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," *Neurocomputing*, vol. 275, pp. 66–72, Jan. 2018.
[55] V. D. Nguyen, H. Van Nguyen, D. T. Tran, S. J. Lee, and J. W. Jeon, "Learning framework for robust obstacle detection, recognition, and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1633–1646, Jun. 2017.
[56] F. S. Mohamad, M. Iqtait, and F. Alsuhimat, "Age prediction on face features via multiple classifiers," in *Proc. 4th Int. Conf. Comput. Technol. Appl. (ICCTA)*, May 2018, pp. 161–166.
[57] X. H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water*, vol. 11, no. 7, p. 1387, 2019.
[58] K. Greff, R. K. Srivastava, J. Koutnìk, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[59] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.
[60] S. Yan. *Understanding LSTM and Its Diagrams*. Accessed: Jan. 10, 2023. [Online]. Available: https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714
[61] A. Soleimani, K. Fouladi, and B. N. Araabi, "UTSig: A Persian offline signature dataset," *IET Biometrics*, vol. 6, no. 1, pp. 1–8, Jan. 2017.
[62] S. Deya, A. Dutta, I. Toledoa, S. Ghosha, and J. Liados, "SigNet: Convolutional Siamese network for writer independent offline signature verification," *Pattern Recognit. Lett.*, pp. 1–7, Sep. 2017, doi: 10.48550/arXiv.1707.02131.
[63] J. Poddara, V. Parikha, and S. Bharti, "Offline signature recognition and forgery detection using deep learning," in *Proc. 3rd Int. Conf. Emerg. Data Ind. 4.0 (EDI40)*, Warsaw, Poland, 2020, pp. 6–9.
[64] S. Rana, A. Sharma, and K. Kumari, "Performance analysis of off-line signature verification," in *Proc. Int. Conf. Innov. Comput. Commun.* (Advances in Intelligent Systems and Computing). Singapore: Springer, 2020, pp. 161–171.
[65] D. Kisku, P. Gupta, and J. Sing, "Fusion of multiple matchers using SVM for offline signature identification," in *Proc. Int. Conf. Secur. Technol.* Berlin, Germany: Springer, 2009, pp. 201–208.
[66] K. Daqrouq, H. Sweidan, A. Balamesh, and M. Ajour, "Off-line handwritten signature recognition by wavelet entropy and neural network," *Entropy*, vol. 19, no. 6, p. 252, 2017.
[67] A. Soleimani, B. N. Araabi, and K. Fouladi, "Deep multitask metric learning for offline signature verification," *Pattern Recognit. Lett.*, vol. 80, no. 1, pp. 84–90, Sep. 2016.
[68] G. Sulong, A. Ebrahim, and M. Jehanzeb, "Offline handwritten signature identification using adaptive window positioning techniques," *Signal Image Process. Int. J.*, vol. 5, no. 3, pp. 1–14, 2014.

FADI MOHAMMAD ALSUHIMAT received the B.S. degree in computer information systems from Al-Hussien Bin Talal University, Ma'an, Jordan, in 2007, and the M.S. degree in computer science from Universiti Utara Malaysia (UUM), Kedah, Malaysia. He is currently pursuing the Ph.D. degree in pattern recognition and deep learning with Universiti Sultan Zainal Abidin (UniSZA), Kuala Terengganu. His research interests include machine and deep learning, data science, and artificial intelligence.

FATMA SUSILAWATI MOHAMAD received the B.Sc. degree in information system management from Oklahoma City University, Oklahoma City, OK, USA, the master's degree in computer science from Universiti Kebangsaan Malaysia, and the Ph.D. degree in computer science from Universiti Teknologi Malaysia. She is currently an Associate Professor with the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA), Kuala Terengganu, Malaysia. Her current research interest includes statistical and biometric pattern recognition.