

# Off-line Handwritten Signature Verification System: Artificial Neural Network Approach

**N. M. Tahir, Adam N. Ausat**

Department of Mechatronics and System Engineering, Abubakar Tafawa Balewa University Bauchi, Nigeria  
E-mail: nuratahir85@gmail.com, mtnura@atbu.edu.ng

**Usman I. Bature**

Department of Computer and Communications Engineering, Abubakar Tafawa Balewa University Bauchi, Nigeria  
E-mail: 4070usa@gmail.com

**Kamal A. Abubakar**

Department of Materials and Metallurgical Engineering, Nigerian Army University Biu, Borno, Nigeria  
E-mail: kamalnice1@gmail.com

**Ibrahim Gambo**

Department of Mathematical Sciences, Faculty of Science, Universiti Teknologi Malaysia  
E-mail: igambo@utm.my

Received: 07 April 2020; Accepted: 14 June 2020; Published: 08 February 2021

**Abstract:** Nowadays, it is evident that signature is commonly used for personal verification, this justifies the necessity for an Automatic Verification System (AVS). Based on the application, verification could either be achieved Offline or Online. An online system uses the signature's dynamic information; such information is captured at the instant the signature is generated. An offline system, on the other hand, uses an image (the signature is scanned). In this paper, some set of simple shaped geometric features are used in achieving offline Verification of signatures. These features include Baseline Slant Angle (BSA), Aspect Ratio (AR), and Normalized Area (NA), Center of Gravity as well as the line's Slope that joins the Center of Gravities of the signature's image two splits. Before the features extraction, a signature preprocessing is necessary to segregate its parts as well as to eliminate any available spurious noise. Primarily, System training is achieved via a signature record which was acquired from personalities whose signatures had to be validated through the system. An average signature is acquired for each subject as a result of incorporating the aforementioned features which were derived from a sample set of the subject's true signatures. Therefore, a signature functions as the prototype for authentication against a requested test signature. The similarity measure within the feature space between the two signatures is determined by Euclidian distance. If the Euclidian distance is lower than a set threshold (i.e. analogous to the minimum acceptable degree of similarity), the test signature is certified as that of the claiming subject otherwise detected as a forgery. Details on the stated features, pre-processing, implementation, and the results are presented in this work.

**Index Terms:** Neural Network, Signature, Verification, Handwritten, Forgery, Genuine, Online and Offline.

## 1. Introduction

Biometric is knowledge of examining the characteristics of an individual for reliable authentication and identification. It is almost impossible for two individuals to have the same biometric information; like the same fingerprint, DNA, handwriting or signature. Biometric is classified as physical and behavioural, the behavioural is the signature, voice and handwriting, while the physical consists of DNA, palm, fingerprint etc. [1-3]. Generally, a signature is conventionally accepted as a biometric for identification of an individual, it represents some behavioural properties of a person, thus widely accepted in schools, banks, organisations hospitals as a means for verification and identification [4].

Today, signature verification is one of the most widely used methods of authentication especially in customer identification in banks. The task of verifying the authenticity of signatures is a difficult one to even we humans, especially when a signature is skilfully forged. Another aspect of the difficulty in signature verification is that even the owner of the signature cannot always write it same in all respects-there would be variations in the size of the parts of the signature and the signature as a whole, variations in orientation among the components and the whole, variations in thickness and lots more of other aspects. Artificial neural network (ANN) tends to produce more accurate results as the

level of data becomes more available, it is flexible in the prediction of applications, ANN can be model to forecast both non-linear and linear processes [5]. Usually, ANN components are the layer, activation function, neuron and weight [6]. Multi-layer perceptron (MLP) is often referred to the networks composing of several layers of the perceptron, it is feed-forward ANN that comprises of three main layers, namely: the input layer, the output layer and the hidden layer [7]. On the reception of a certain weight, every input nodule in the input layer will be moved to the neurons [8]. Therefore, the capability of neural networks to learn patterns and form generalizations about virtually any kind of data make them appropriate for this task. Offline signature verification, as opposed to online signature verification, involves getting signatures written by hand on paper, followed by scanning the image and finally feeding the image data to a verification system, for example, a computer program, for verification.

This work focusses on off-line handwritten signature verification system using artificial neural network approach, since the manual means of verification are now replaced with more accurate and robust automatic methods, this will aid in checking the chances of errors and reducing the complexity in traditional means of identification. The framework of this paper is organised as follows: The review of the related work is presented in section II, this section described the basic concept related to this work of ANN. Section III shows the methodology and materials used in achieving the desired process of offline signature verification using the ANN approach, section IV is the discussion of the results that shows the performance and epochs when data augmentation is used, Snapshot of the graphical user interface showing a signature correctly recognized as genuine and performance test accuracy of 82.5%. Finally, section V shows the concluding remark of the work.

## 2. Literature Review

Current works in the field of signature verification using neural networks are characterized by employing the use of pre-processing techniques like morphological operations and feature extraction to derive the most relevant and best features in the data as a trick for improving accuracy at the same time, bringing down the amount of training time. For instance, [9] proposed a novel offline signature verification technique implemented to reduce the error rate in the existing literature. The proposed method is based on the best features selection technique named as genetic algorithm (GA). Also, Panchal and [10] proposed an offline signature verification technique based on geometric feature extraction using artificial neural networks.

The area of handwritten signature verification has been largely studied in the last decades and remains an open research problem. Suggested in [11] Offline Handwritten Signature Recognition using Biometrics, which refers to identifying an individual based on his or her physiological or behavioural characteristics, can reliably distinguish between an authorized person and an imposter. Due to the uniqueness of biometric identifiers to individuals, they are more reliable in verifying identity than knowledge-based and token methods like password and pin. However, there is a privacy concern about the collection of biometric identifiers especially about their ultimate use [12]. The choice of a specific biometric feature for use in a given application involves the consideration and comparison of several factors, seven of such factors were identified by [13-15].

The problem of signature verification is generally confronted using three approaches: template matching approach, structural approach and statistical approach. Template Matching; Here, a signature is compared to one stored in the database. The most widely used technique for achieving template matching is the Dynamic Time Warping (DTW) [16]. Hyo-Rim Choi and Tae-Yong Kim came up with a modified DTW algorithm and observed improved performance than the conventional DTW [17]. An algorithm for 2D geometric warp was developed by D. J. Kennard et al [18]. In 2016, G. Alvarez et al. [19] proposed a model based on VGG16 architecture which used ICDAR 2011 SigComp dataset for training the model, the system produced 97% accuracy and 95% accuracy for Dutch and Chinese signatures respectively. Structural Approach; Here, graph trees and other similar structures are used to represent the signatures. The resultant structures are then compared through the use of matching algorithms to execute the verification job. Among popular of these structural approaches, the work [20] in which a signature is represented in a polygon formed by joining the endpoints of the signature. [21] Used structural features obtained from the contours of signature images to train a neural network. Statistical Approach; this involves feeding features from signatures into a learning algorithm so that the signatures can be classified. Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Hidden Markov Models (HMM) are the most popular and extensively used classifiers for this task. Presented in [22], is a robust system for off-line signature verification using simple features, different cell resolutions and multiple codebooks in an HMM framework. Just of recent Victor LF et al showed that the use of features extracted using a deep convolution neural network (CNN) combined with a writer-independent (WD) SVM classifier resulted in significant improvement in the performance of handwritten signature verification (HSV) when compared to the previous state-of-the-art methods [23]. Likewise, S. Parvesh et al proposed an Offline Graphical Analysis of Signatures Using Simple Geometric Features and Artificial Neural Network (ANN) [24].

A geometric features offline signature verification method was proposed using the six geometric features then ANN was employed to recognized and authenticate the samples, it produces an efficiency of about 89.24% with a threshold of 80% [25] but the system uses random forgeries but not much on skilled forgeries. The performance of the proposed method is examined using the Backpropagation learning technique, with 18 sets of different users having a

varying number of training and testing samples. Experiments were conducted on CEDAR dataset. A handwritten signature recognition method using CNN was proposed [26], the system isolated the signature pixels from the noise or background pixels via the use of image processing methods but did not provide the visual presence of the weights. Some difficulties were usually experienced in obtaining desired good features and system performance, these were addressed by learning the illustration from the images using CNN [27]. Four datasets were observed, and good progress was achieved on the equal error rate, but the system did not test the features for sorting. A deep multi-tasking metric learning using deep NN, [27] explained a distance metric among sets of signatures. Unlike in our proposed work, [27] used feature extractions using handcraft instead of using the signature pixels as the inputs. Muhammad Reza et al [28] proposed the client-entropy life, which has been planned to cluster and characterize dynamic signatures in classes that may be associated with signature variability, [28] used a simple technique of classification, though it has a lower error, our system uses more robust techniques of CNN.

In this work, a method for offline signature verification that doesn't need the time-consuming trial and error work of determining which set of features work best is employed. The neural network can by itself, learn what features are important and relevant for the task. All required on the part of the developer, is a careful selection of the right neural network architecture along with sufficient data or data augmentation in the case of scarce training data, to achieve high acceptable accuracy in verifying signatures with minimal computational complexity and training time. For training and testing, the signature dataset used is the SigComp2011 signature dataset due to its popularity, ease of use and suitability for this particular work. All the system does is simply categorizing an input signature as genuine or forged. For good enough accuracy during verification, the signature has to be registered during training by feeding some variants of it-both the genuine and forged variants.

### 3. Material and Method

#### 3.1. Materials used

The software used in the development of the system is the WingWare Python IDE and the Anaconda IDE along with the required library modules.

#### 3.2. Methodology flowchart

To design a system, which will detect the forged signatures by comparing some special features with the original one, the following architecture in fig. 1 has been proposed.

From fig. 1 route 1 will be accomplished first before moving to route 2. Route 1 is the initial retrieval of the signature and NN training, route 2 is the next stage after the retrieval of the signature, to be testing and finally checking out the outcomes of the NN.

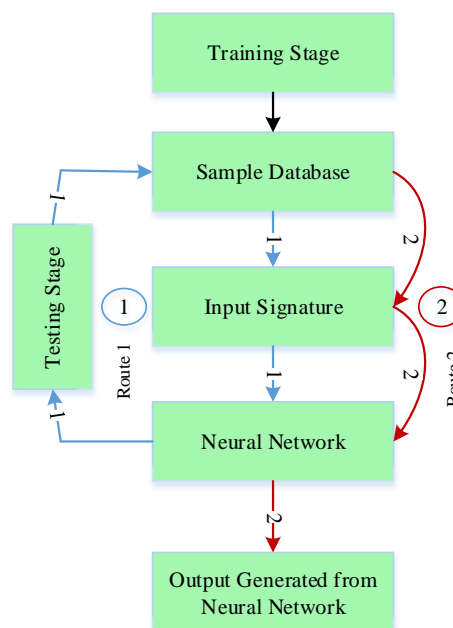


Fig.1. Methodology flow chart

These are further explained as:

- a. Data acquisition: The signature dataset is collected from Sigcomp 2011 database. The SigComp2011 training set comes with 239 genuine signatures and 123 forged signatures. 219 genuine signatures were used in the training and so, 20 reserved for testing. Likewise, of the 123 forged signatures, 103 were used in the training and 20 reserved for testing.
- b. Training Stage: training stage consists of the following two steps:
  - (1) Retrieval of a signature image from a database.
  - (2) Neural network training.
- c. Testing Stage: testing stage consists of following two steps:
  - (1) Retrieval of a signature to be tested from a database
  - (2) Checking output generated from a neural network
- d. Verification Stage: The trained neural network is used to check whether the signature is genuine or forged. If the signature matches, then it shows genuine otherwise forgery.

### 3.3. System Model Equations

Before diving deep into the model equations derivations, it would be worthy and helpful to have the picture of the MLP neural network used in this work and how it works, MLP can be aimed at exploiting resources utilization of the device and used to develop most of the CNN variants [29].

Multilayer Perceptrons are the classical type of neural network. They are comprised of one or more layers of neurons. Data is fed to the input layer, there may be one or more hidden layers providing levels of abstraction, and predictions are made on the output layer, also called the visible layer. MLPs are suitable for classification prediction problems where inputs are assigned a class or label. They are also suitable for regression prediction problems where a real-valued quantity is predicted given a set of inputs. Data is often provided in a tabular format.

MLPs mostly use for tabular datasets, classification prediction problems, and regression prediction problems.

They are very flexible and can be used generally to learn a mapping from inputs to outputs. This flexibility allows them to be applied to other types of data.

This work employed a perceptron-based approach to accomplishing the solution [30]. A 6-layer multi-perceptron neural network sketch is shown in fig. 2. Each node in any layer is forward connected to all nodes in the next layer. Notice from the diagram that some connections were intentionally left out just to avoid jumbling of the diagram. The first set of nodes arranged vertically in the diagram form the input layer. These nodes are in a real sense, nothing more than the pixel values that form the signature image being fed into the system. These inputs are then further fed into the first hidden layer depending on the values of the link weights connecting the input layer and the output layer. At each of the first hidden layer nodes, the signals are arithmetically summed and an activation function (the sigmoid function chosen in this work) is applied on the sum to produce outputs of this layer [30]. The inputs continue their journey in this fashion up to the output layer whereupon application of the sigmoid activation function, the output of the network is finally obtained which represents the network's answer. This phase is called the Forward Propagation. The next step that follows is the calculation of errors obtained by comparing the expected or target output values and what the network output. These errors are then back-propagated into the hidden layers and finally, the link weights are updated in such a direction as to minimize the error using the gradient descent algorithm. And thus, learning is said to have taken place [31, 32].

The form of learning used in this neural network is called supervised learning, meaning that the network needs to be trained first on examples before it can learn how to accomplish the task [33,34], as opposed to unsupervised learning which is beyond the scope of this work [35,36].

The fig. 2 would seem sufficient in carrying out the desired task but in a practical sense, it is known that the more the hidden layers and hence, the more the hidden nodes, the harder it is to train the network (i.e. the more the training data required) and the more would be the program run-time [37]. The solution to this problem is reducing the resolution of the signature images to 5pixels  $\times$  5pixels such that the number of input nodes of the network is only 25. Therefore, it follows that subsequent layers would have less and less number of nodes since the work of each layer is indirectly nothing more than representing the features in the previous layer in a smaller form. Let  $h$  be the number of hidden layers.

$$n(h) = 4 \quad (1)$$

Let  $w_i h_1$  be the matrix of link weights connecting the nodes in the input layer to the weights in the first hidden layer.

Therefore,

$$w_i h = \begin{matrix} & w_{11} & w_{21} & w_{31} & - & - \\ & w_{12} & w_{22} & w_{32} & - & - \\ & w_{13} & w_{23} & w_{33} & - & - \\ & - & - & - & - & - \\ & - & - & - & - & - \end{matrix} \quad (2)$$

Where:  $w_{11}$  is the link weight that connects the first node in the input layer to the first node in the first hidden layer (remember it is  $w_i h_1$  i.e. w from input to hidden 1),  $w_{21}$  is the link weight that connects the second node in the input layer to the first node in the hidden layer.

Similarly, let  $W h_1 h_2$  be the matrix of link weights connecting the nodes in the first hidden layer to the nodes in the second hidden layer.

$$w h_1 h_2 = \begin{matrix} & w_{11} & w_{21} & w_{31} & - & - \\ & w_{12} & w_{22} & w_{32} & - & - \\ & w_{13} & w_{23} & w_{33} & - & - \\ & - & - & - & - & - \\ & - & - & - & - & - \end{matrix} \quad (3)$$

Where:  $w_{11}$  here, is the link weight that connects the first node in the first hidden layer to the first node in the second hidden layer (remember it is  $W h_1 h_2$  i.e. w from hidden 1 to hidden 2),  $w_{21}$  is the link weight that connects the second node in the first hidden layer to the first node in the second hidden layer [30].

$$w h_2 h_3 = \begin{matrix} & w_{11} & w_{21} & w_{31} & - & - \\ & w_{12} & w_{22} & w_{32} & - & - \\ & w_{13} & w_{23} & w_{33} & - & - \\ & - & - & - & - & - \\ & - & - & - & - & - \end{matrix} \quad (4)$$

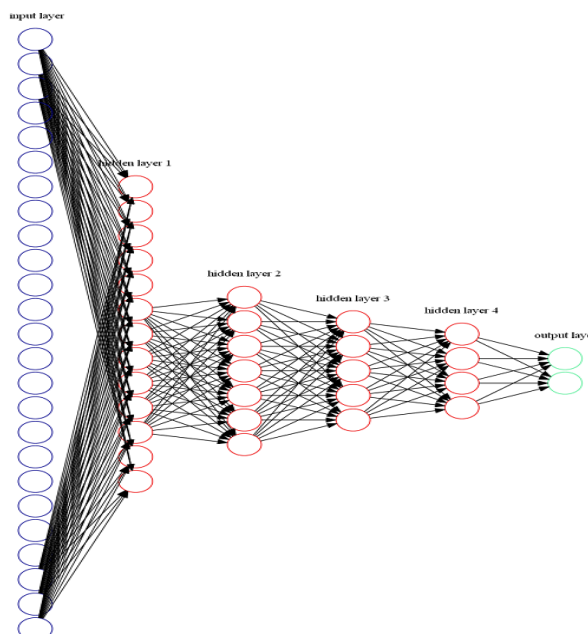


Fig.2. A 6-layer multi-perceptron neural network.

Similarly,

$$\begin{array}{cccccc}
 & w_{11} & w_{21} & w_{31} & - & - \\
 & w_{12} & w_{22} & w_{32} & - & - \\
 wh_3h_4 = & w_{13} & w_{23} & w_{33} & - & - \\
 & - & - & - & - & - \\
 & - & - & - & - & -
 \end{array} \quad (5)$$

Finally, for the link weights between the last hidden layer and the output layer was obtained as;

$$\begin{array}{cccccc}
 & w_{11} & w_{21} & w_{31} & - & - \\
 & w_{12} & w_{22} & w_{32} & - & - \\
 wh_4o = & w_{13} & w_{23} & w_{33} & - & - \\
 & - & - & - & - & - \\
 & - & - & - & - & -
 \end{array} \quad (6)$$

Input to the final output layer is obtained as in equation

$$io = wh_4 \bullet Oh_4 \quad (7)$$

So, the output of the final output layer is obtained as in equation (8);

$$o0 = s(io) \quad (8)$$

Given  $Wh_4o$  is the matrix of link weights between the fourth hidden layer and a final output layer of the network

The error in the final outputs is a matrix given by the matrix of the target values minus the output matrix of the final output layer is as in equation (9) and link weights to the final output layer in equation (10) [32, 38].

$$Errors = targets - Final\_Outputs \quad (9)$$

$$wh_4o = previous\ wh_4o - learning\ Rate \times \frac{d(Errors_o)}{dwh_4o} \quad (10)$$

Substitute and simplifying the above, equation (11) is obtained as;

$$New\ w_ih_i = previous\ w_ih_i + learning\ Rate \times (Errors) \quad (11)$$

### 3.4. Data Collection and Methodical details

Below is a summarised description of the system implementation:

- It is worth mentioning that the signature data set used is the SigComp 2011 dataset. The reason for use of this particular dataset is that it is in a format that completely fits the problem being addressed in this work; learning how to classify a signature as either genuine or forged. So, the dataset consists of collected images of genuine signatures and their corresponding possible forged signatures. Moreover, it is freely available, unlike the GPDS signature dataset [39] which involves a long process before one can access the dataset (GPDS-Group, 2018). Furthermore, data augmentation to improve performance was used. To achieve that, two additional images were generated for each signature image by rotating the original image by 10 degrees clockwise and anti-clockwise.
- As stated earlier in section with reasons, the image size was chosen to be 5px × 5px. Hence, the required number of input nodes is 25; each node for each of the pixels.
- The weights are randomly initialised with values between 0 and 1 each time the program is run
- To lessen the chance of getting stuck in a bad or local minimum after the training session.
- As explained before, the number of hidden layers for accomplishing the task is four (4).
- The selection of the number of nodes in each hidden layer follows from the use of the rule of thumb that the size of the hidden layer should be somewhere between the input layer size and the output layer size or the next

- immediate layer size.
- g. Thus, the size of the first hidden layer is set to 13. The size of the second hidden layer is 7. The third hidden layer is 5 and the fourth hidden layer is 4.
  - h. The activation function chosen is the sigmoid function which was used due to its popularity and most importantly, it is simple and easy to use in calculations.
  - i. The number of output layer nodes is inevitably two (2), representing the two signature classes; genuine and forged. Note: The output layer nodes represent a matrix with only two elements. It is designed in such a way that for a genuine signature, it should target [0.01, 0.99] and for a forged signature, it should target [0.99, 0.01]. So, during test or usage, if the first element is the highest, it means that the system believes the signature is forged otherwise it means the signature is genuine.
  - j. The errors are back-propagated to the hidden layers in a direct proportion to the values of the connected weights.
  - k. Gradient descent algorithm is used in the error minimisation process and the subsequent weights update.
  - l. Determining the values of the learning rate and the epochs is a matter of trial and error along with observing a trend in the performance/accuracy of the system as the values are being changed. Setting the learning rate to 0.01 and epochs to 1000 yields a performance of up to 82.5% when the initial random weights generated are good enough.

### 3.5. Organization of the Training Dataset

The SigComp2011 signatures training dataset consisting of two folders: The Chinese signatures and the Dutch signatures [40]. The Dutch signatures folder contains two sub-folders: one for offline signatures and the other for online signatures. The offline signatures folder was taking, it further consists of two folders one for genuine signatures and the other containing forged signatures of the genuine signatures. Note that the Dutch signatures are no different than English signatures because the Dutch alphabet system is the same as the English alphabet system. So, there is nothing to worry about in training a network using both Dutch and English based signatures together.

### 3.6. Organization of the Test Dataset

The test dataset has a similar organization to the training dataset except that there are further two sub-folders: Questioned and References. The questioned contains both genuine and forged signatures while the References contain only genuine signatures. All the signatures in both of these two sub-folders are not in the training dataset and so, meaning never enrolled in the system before and so, never seen by the network before. This makes the test dataset unfit for this system going by its original design; verifying or recognizing a signature as genuine or forged given that both the genuine and forged variants of the signature were once enrolled into the system and not to be a universal signature classifier that can identify any forged or genuine signature even if never seen before. Therefore, for this work, the only way to use the SigComp2011 test dataset is to divide the training dataset into two portions; use one portion in the training and test with the other portion.

### 3.7. Experiment

The following experiments/tests were carried out. Even though finding the best performance of a neural network involves trial and error, that doesn't mean we go blindly, placing the burden on ourselves for having to test all possible parameter combinations, which is not feasible and practical. But rather we take a systematic approach so that we don't miss the sweetest spots. To accomplish that, the basic idea is this; Firstly, we set out with learning rate = 0 and epochs = 1. Secondly, we continue to increase the learning rate in say sub-multiples and multiples of 10 (0.0001, 0.001...0.1, 1...), maintaining the epochs at 1 and observing the performance till the best is obtained. The reasoning behind this is that the performance of a neural network is always better with the optimum number of epochs which is always greater than 1 i.e. can't be 1. So, we set epochs=1 and vary the learning rate. As soon as we find the best performance with that 1 epoch, we then stop varying the learning rate and proceed to increase the epochs to get better performance. As for the other network parameters like the number of hidden nodes and hidden layers, we don't tamper with them because they form the backbone of this research work as earlier explained in this section.

Experiment 1: In this experiment, no form of data augmentation was used. The SigComp2011 training set comes with 239 genuine signatures and 123 forged signatures. 219 genuine signatures were used in the training and so, 20 reserved for testing. Likewise, of the 123 forged signatures, 103 were used in the training and 20 reserved for testing.

Experiment 2: In this experiment, data augmentation in the form of rotating the signature images by 10 °clockwise and 10 °anti-clockwise was incorporated into the training.

## 4. Result and Discussion

This part embodies the outcomes of tests performed on the system to assess its accuracy and performance. These results are then interpreted in the light of how practical the system is in identifying genuine and forged signatures.

#### 4.1. Testing the Verification System

Initially, the system performance was constant at 50% for both experiments 1 and 2. After careful analysis, it was understood that due to the complexity of the task and the more than 500 number of link weights to be adjusted each time a new training signature is learnt, to get any higher performance or level of accuracy, a large number of variants of each signature and high amount of training time is required. Why not keep some weights constant? Most likely the numbers of weight updates were too much each time a training data is passed in causing the network to easily overshoot the true global minimum of the error function after each mass weights update. That was exactly done; the link weights that emerge from the second hidden layer to the third hidden layer were arbitrarily chosen to be kept constant without updating them after the first initialization of the entire link weights to randomly generated values. Thereafter, following testing of the network with the 20 reserved genuine signatures and the 20 reserved forged signatures, the graph shown in fig. 3 displays the results of experiment 1 described in the previous section, with the learning rate varied and epochs maintained at 1.

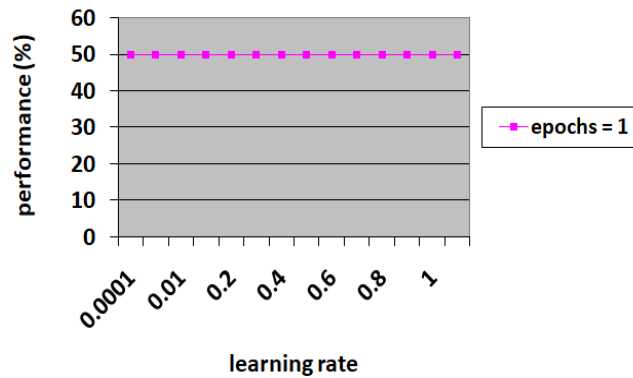


Fig.3. Performance versus learning rate when no data augmentation

The second graph shown in fig. 4 shows the system performance as epochs is varied and the learning rate maintained at the optimal value obtained in fig. 3 above i.e. learning rate = 0.01. From the graph shown in fig. 4, it is clear that the network starts over-fitting when the number of epochs exceeds 14,000 and so no need of going further.

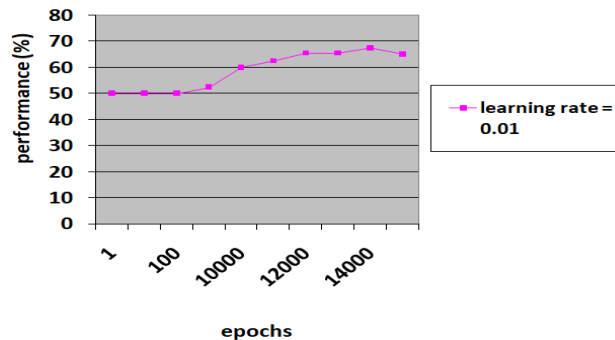


Fig.4. Performance versus epochs when no data augmentation used

The graph is shown in fig. 5 presents the results of experiment 2 with the learning rate varied and epochs maintained at 1. As can be seen from the graph above, the results are the same as those in the graph of fig. 3 in experiment 1 above. Just as in that experiment, the optimum learning rate was found to be 0.01.

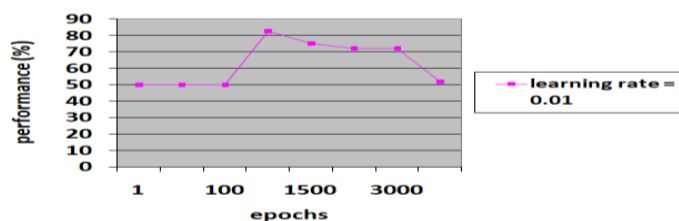


Fig.5. Performance versus learning rate when data augmentation is used



The graph in fig. 6 below shows the system performance as epochs are varied and the learning rate maintained at the optimal value obtained in the graph above i.e. optimal learning rate = 0.01. As can be seen from the graph above the result are same as those in the fig.3 in experiment I above. Just as in that experiment the optimum learning rate was found to be 0.01. The graph is shown in fig. 7 shows the system performance as epochs is varied and learning rate maintained at the optimal value obtained in the graph above i.e., optimal.

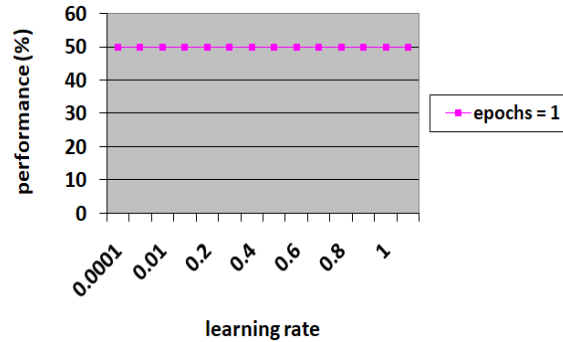


Fig.6. Performance versus epochs when data augmentation is used

Learning rate = 0.01. It is evident from the graph that the best overall performance is 82.5%. The set of link weights that give such performance are saved in external files for sake of future use e.g. transfer learning or when the system is put to use such that there is no need of retraining before use.

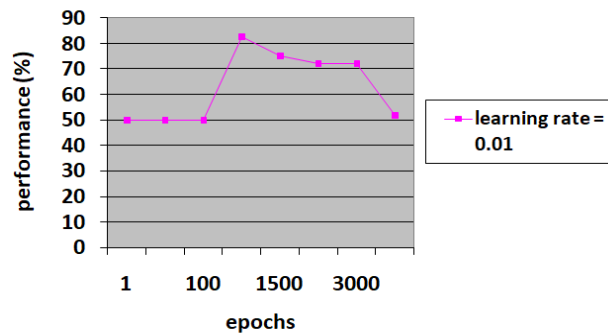


Fig.7. Performance versus epochs when data augmentation is used

#### 4.2. Findings and Interpretation of Results

From the two experiments performed, it was evident that experiment 2 yields far better performance than experiment 1. Since the only difference between the two experiments is the use of data augmentation in experiment 2, then it follows that the higher performance in experiment 2 is a direct effect of the use of data augmentation. Moreover, the training time in experiment 2 is about 10 minutes unlike in experiment 1 which takes about 40 minutes to achieve the performance presented. This counters the expectation that experiment 2 training session should take longer due to data augmentation making the size of the training data to triple! But the clear truth was that data augmentation helps the network in converging faster with little efforts (epochs).

#### 4.3. Graphical User Interface (GUI)

Through the following interface, the user can interact with the system in two ways: verifying a signature which was already enrolled earlier and enrolling new signatures. The interface also shows the image of the signature that is undergoing verification.

The snapshot of the graphical user interface shown in fig. 8 presents a signature correctly recognized as a forge, the system will compare with the original enrolled signature and present the result via the developed graphic user interface (GUI). Similarly, fig. 9 shows when the verified signature is genuine, the GUI

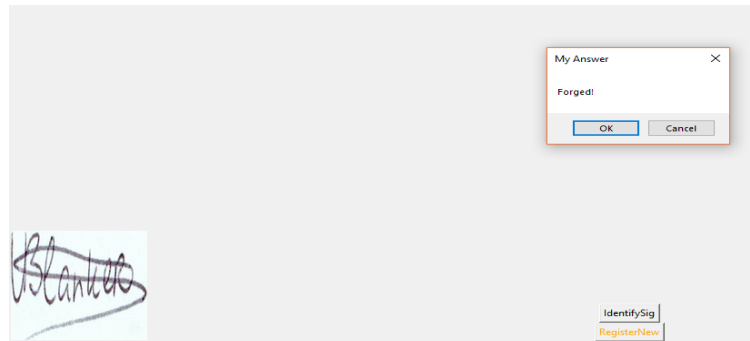


Fig.8. Snapshot of the graphical user interface showing a signature correctly recognized as forged

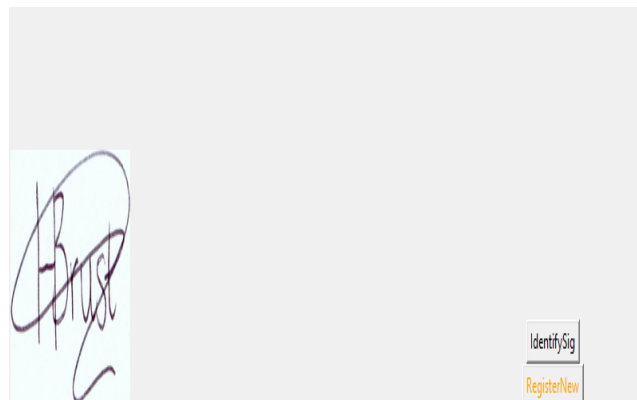


Fig.9. Snapshot of the graphical user interface showing a signature correctly recognized as genuine

Table 1 shows the performance comparison of the proposed method with state-of-the-art approaches. The proposed method reported 82.50% accuracy which outperforms the baseline techniques [41-43].

Table 1. Performance of different signature verification techniques and approaches

NO	Technique/Author	Performance/Test Accuracy (%)
1	Proposed Technique	82.5
2	H. Baltzakis and N. Papamarkos [43]	80.81
3	S. T. Panchal and V. V. Yerigeri [41]	80.00
4	S. M. Odeh and M. Khalil [42]	78.80

## 5. Conclusion

With the highest attained performance of 82.50% and minimal training time of about 10 minutes, it is evident that signature verification with an acceptable level of accuracy has been realized without the need for use of image pre-processing and feature selection techniques; the rights NN architecture together with data augmentation are all needed. The reason why the performance of roughly 80% is ok and acceptable for signature verification is the fact that in banks, for example, a customer writes his signature more than once in the hope that one of the signatures is very close to the one enrolled into the system and so that if the system doesn't correctly identify one, it correctly identifies the other. So, we don't need a system that has to be 100% accurate. And above all, in practice, no any signature verification system or no any neural network, in particular, can attain performance level of 100% due to the complexities in the task and high level of dissimilarities inherent across signatures belonging to even the same person not to talk of signatures of different individuals. But, the performance of the proposed system can be improved in the following ways; use of more data augmentation techniques not only rotation, like adding noise is sure to improve the performance of the system. Higher and higher performance can still be attained with the use of more and more variants or different examples of each of the enrolled signatures. The performance can be greatly enhanced by the use of optimization algorithms that yield faster convergence than the gradient descent algorithm used in this work.

## Acknowledgements

The authors wish to thank the Department of Mechatronics and System Engineering, Abubakar Tafawa Balewa University Bauchi for the provision of Laboratory equipment.

## References

- [1] Ahmed, Zainab J., and Loay E. George. "A Recognition System for Subjects' Signature Using the Spatial Distribution of Signature Body." *Journal of Southwest Jiaotong University*, 2020, 55(1).
- [2] Ge, Can, Juanjuan Li, Lina Liu, Gukai Li, Guoxing Yang, Hui Tang, Hanlin Yang, Yifan Xia, Rui Bao, and Haiyu Zhang. "Design and Implementation of Intelligent Singer Recognition System." *MS&E*, 2020, 790(1).
- [3] Shah, Jamal Hussain, Zonghai Chen, Muhammad Sharif, Mussarat Yasmin, and Steven Lawrence Fernandes. "A novel biomechanics-based approach for person re-identification by generating dense color sift salience features." *Journal of Mechanics in Medicine and Biology*, 2017, 17(7).
- [4] Bibi, Kiran, Saeeda Naz, and Arshia Rehman. "Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities." *Multimedia Tools and Applications*, 2020, 79(1).
- [5] C. S. Barhmi and O. E. Fathi, "Hourly Wind Speed Forecasting based on Support Vector Machine and Artificial Neural Networks," *International Journal of Artificial Intelligence*, 2019, (8): pp. 286-291
- [6] Nur Haizum Abd Rahman and Muhammad Hisyam Lee, "Artificial neural network forecasting performance with missing value imputations," *International Journal of Artificial Intelligence*, 2020, (9): pp. 33-39.
- [7] F. Yumono, et al., "Artificial Neural Network for Healthy Chicken Meat Identification," *International Journal of Artificial Intelligence*, 2018, (7): pp. 63-70.
- [8] K. C. Rani and Y. Prasanth, "A Decision System for Predicting Diabetes using Neural Networks," *International Journal of Artificial Intelligence*, 2017, (6): pp. 56-65.
- [9] Sharif, Muhammad, Muhammad Attique Khan, Muhammad Faisal, Mussarat Yasmin, and Steven Lawrence Fernandes. "A framework for offline signature verification system: Best features selection approach." *Pattern Recognition Letters*, 2018.
- [10] Panchal, S. T. and Yerigeri, V. V.: "Offline Signature Verification based on Geometric Feature Extraction using Artificial Neural Network", *IOSR Journal of Electronics and Communication Engineering*, 2018, 13(3): pp. 55-59.
- [11] Khuwaja, Gulzar A., and Mohammad S. Laghari. "Offline handwritten signature recognition." *World Academy of Science, Engineering and Technology*, 2011, (59): pp.1300-1303.
- [12] Pinto, João Ribeiro, Jaime S. Cardoso, and André Lourenço. "Evolution, current challenges, and future possibilities in ECG biometrics." *IEEE Access* 6, 2018: pp34746-34776.
- [13] Jain, Anil K., Ruud Bolle, and Sharath Pankanti, eds. *Biometrics: personal identification in a networked society*. Springer Science & Business Media, 2006, (479).
- [14] Kalyani, C. "Various biometric authentication techniques: a review." *Journal of Biometrics & Biostatistics* 2017, 8(5).
- [15] Jain, Anil K., Karthik Nandakumar, and Arun Ross. "50 years of biometric research: Accomplishments, challenges, and opportunities." *Pattern Recognition Letters*, 2016, (79): pp. 80-105.
- [16] Diaz, Moises, Miguel A. Ferrer, Donato Impedovo, Muhammad Imran Malik, Giuseppe Pirlo, and R egan Plamondon. "A prospective analysis of handwritten signature technology." *ACM Computing Surveys (CSUR)*, 2019, 51(6): pp. 1-39.
- [17] Choi, Hyo-Rim, and Tae-Yong Kim. "Modified dynamic time warping based on direction similarity for fast gesture recognition." *Mathematical Problems in Engineering* 2018.
- [18] Kennard, Douglas J., William A. Barrett, and Thomas W. Sederberg. "Offline signature verification and forgery detection using a 2-D geometric warping approach." In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, IEEE, 2012. pp. 3733-3736.
- [19] Alvarez, Gabe, Blue Sheffer, and Morgan Bryant. "Offline signature verification with convolutional neural networks." Technical report, Stanford University, 2016.
- [20] Zafar, Sohail, and Rashid Jalal Qureshi. "Off-line signature verification using structural features." In *Proceedings of the 7th International Conference on Frontiers of Information Technology*, 2009, pp. 1-3.
- [21] Armand, Stephane, Michael Blumenstein, and Vallipuram Muthukumarasamy. "Off-line signature verification based on the modified direction feature." In *18th International Conference on Pattern Recognition (ICPR'06)*. IEEE, 2006, (4): pp. 509-512.
- [22] Justino, Edson JR, Fl vio Bortolozzi, and Robert Sabourin. "Off-line signature verification using HMM for random, simple and skilled forgeries." In *Proceedings of Sixth International Conference on Document Analysis and Recognition*. IEEE, 2001, pp. 1031-1034.
- [23] Souza, Victor LF, Adriano LI Oliveira, and Robert Sabourin. "A writer-independent approach for offline signature verification using deep convolutional neural networks features." In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, IEEE, 2018, pp. 212-217.
- [24] Saini, Parvesh, Ishita Uniyal, and Neeraj Singh. "Offline Graphical Analysis of Signatures Using Geometric Features and Artificial Neural Network." In *Intelligent Communication, Control and Devices*, Springer, Singapore 2018, pp.1065-1077.
- [25] Chandra, Subhash, and Sushila Maheskar. "Offline signature verification based on geometric feature extraction using artificial neural network." In *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, IEEE, 2016, pp. 410-414.
- [26] Hafemann, Luiz G., Robert Sabourin, and Luiz S. Oliveira. "Learning features for offline handwritten signature verification using deep convolutional neural networks." *Pattern Recognition* 70, 2017, pp.163-176.
- [27] Soleimani, Amir, Babak N. Araabi, and Kazim Fouladi. "Deep multitask metric learning for offline signature verification." *Pattern Recognition Letters* 80, 2016, pp.84-90.

- [28] Sigari, Mohamad Hoseyn, Muhammad Reza Pourshahabi and Hamid Reza Pourreza. "Offline Handwritten Signature Identification and Verification Using Multi-Resolution Gabor Wavelet.", 2011.
- [29] Hailesellasiye, Muluken Tadesse, and Syed Rafay Hasan. "MulNet: A flexible CNN processor with higher resource utilization efficiency for constrained devices." *IEEE Access* 7, 2019, pp.47509-47524.
- [30] Castro, Wilson, Jimy Oblitas, Roberto Santa-Cruz, and Himer Avila-George. "Multilayer perceptron architecture optimization using parallel computing techniques." *PloS one*, 2017, (12).
- [31] Anuse, Alwin, and Vibha Vyas. "A novel training algorithm for the convolutional neural network." *Complex & Intelligent Systems* 2016, 2(3): pp. 221-234.
- [32] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6.
- [33] Hwang, Jun Kwon, Geun Young Yun, Sukho Lee, Hyeongjoon Seo, and Mat Santamouris. "Using deep learning approaches with variable selection process to predict the energy performance of a heating and cooling system." *Renewable Energy* 149, 2020, pp.1227-1245.
- [34] Hwang, Jun Kwon, Patrick Nzivugira Duhirwe, Geun Young Yun, Sukho Lee, Hyeongjoon Seo, Inhan Kim, and Mat Santamouris. "A Novel Hybrid Deep Neural Network Model to Predict the Refrigerant Charge Amount of Heat Pumps." *Sustainability* 2020, 12(7): pp. 2914.
- [35] Riachy, Chirine, Fouad Khelifi, and Ahmed Bouridane. "Video-based person re-identification using unsupervised tracklet matching." *IEEE Access* 7, 2019, pp.20596-20606.
- [36] Zhong, Zhun, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. "Learning to adapt invariance in memory for person re-identification." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [37] Umamaheswari, J., and A. Akila. "Improving Speech Recognition Performance using Spectral Subtraction with Artificial Neural Network." *International Journal of Advanced Studies of Scientific Research*, 2018, 3(11).
- [38] K. Kancharla, V. Kamble and M. Kapoor, "Handwritten Signature Recognition: A Convolutional Neural Network Approach," *2018 International Conference on Advanced Computation and Telecommunication (ICACAT)*, Bhopal, India, 2018, pp. 1-5.
- [39] Hafemann, Luiz G., Robert Sabourin, and Luiz S. Oliveira. "Learning features for offline handwritten signature verification using deep convolutional neural networks." *Pattern Recognition* 70, 2017, pp. 163-176.
- [40] Liwicki, Marcus, Muhammad Imran Malik, C. Elisa Van Den Heuvel, Xiaohong Chen, Charles Berger, Reinoud Stoel, Michael Blumenstein, and Bryan Found. "Signature verification competition for online and offline skilled forgeries (sigcomp2011)." *In 2011 International Conference on Document Analysis and Recognition*, pp. IEEE, 2011, pp. 1480-1484.
- [41] S. T. Panchal and V. V. Yerigeri., "Offline signature verification based on geometric feature extraction using artificial neural network", *IOSRJECE*, 2018, 13(3): pp. 53-59.
- [42] S. M. Odeh and M. Khalil, "Apply multi-layer perceptron neural network for off-line signature verification and recognition", *IJCSI*, 2011, 8(6): pp.261.
- [43] Baltzakis, H., & Papamarkos, N. A new signature verification technique based on a two-stage neural network classifier. *Engineering applications of Artificial intelligence*, 2001, 14(1), pp. 95-103.

## Authors' Profiles



**Nura Musa Tahir** received the B.Eng. degree in Electrical Engineering from Bayero University, Kano in 2012 and M.Eng. Degree in Mechatronics and Automatic Control from the Universiti Teknologi Malaysia in 2016. He has been involved in research at Mechatronics systems and Robotics Laboratory in Universiti Teknologi Malaysia on vibration control, robotics and Mechatronics system design. He has been a lecturer at the Department of mechatronic and system Engineering, Abubakar Tafawa Balewa University, Bauchi Nigeria since 2014. His main research interests include Vibration Control, Mechatronics Systems Design, Control System Design, and Robotics.



**Adam N. Ausat** received a B.Eng. in Mechatronics and System Engineering from Abubakar Tafawa Balewa University Bauchi Nigeria in 2020. His major field of research interest includes artificial intelligence, embedded system designs and cybersecurity.



**Usman Isyaku Bature** is currently a PhD. Candidate in the Department of Electrical Engineering, Universiti of Teknologi Petronas (UTP) Malaysia. He received a B.Eng. Computer Engineering from Bayero University Kano (BUK), Kano city, Nigeria and degree of Master of Engineering (Electrical - Computer and Microelectronic system) from Universiti Teknologi Malaysia (UTM), Skudai, Johor Bahru, Malaysia. He is currently a lecturer in the Department of Computer and Communications Engineering, Abubakar Tafawa Balewa University Bauchi (ATBU), Nigeria. His research interest includes Image processing, Nano-material, Embedded systems and Biomedical Systems.



Robotics and Software development.

**Kamal Abubakar Abubakar** received his B.Eng. degree in Electrical Engineering from Bayero University, Kano in 2012 and M.Eng. Degree in Mechatronics and Automatic Control Engineering from the Universiti Teknologi Malaysia in 2016. He has been involved in research works at Mechatronics Systems Laboratory and Centre for Artificial Intelligence and Robotics (CAIRO) in Universiti Teknologi Malaysia on Robotics, Control, Artificial Intelligence, Neural Network and Mechatronics system design. He has been a lecturer at the Department of Materials and Metallurgical Engineering, Nigerian Army University Biu, Borno, Nigeria, since February 2020. His main research interests include Vibration Control, Mechatronics Systems Design, Control System Design,



**Ibrahim Gambo** is currently a Postdoctoral fellow at the Department of Mathematical Sciences, Faculty of Science Universiti Teknologi Malaysia (UTM). He received his Doctor of Philosophy Mathematics as well as his Master degree in Mathematics from Universiti Teknologi Malaysia and his Bachelor degree of Mathematics from Bayero University Kano (BUK), Kano, Nigeria. His research interest includes Fuzzy sets theory, Algebra and its Applications in other multidisciplinary areas of Engineering. He has published quite a several refereed index journals and conference papers in the areas of his expertise.

**How to cite this paper:** N. M. Tahir, Adam N. Ausat, Usman I. Bature, Kamal A. Abubakar, Ibrahim Gambo, "Off-line Handwritten Signature Verification System: Artificial Neural Network Approach", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.13, No.1, pp.45-57, 2021. DOI: 10.5815/ijisa.2021.01.04