Sana'a University Journal of Applied Sciences and Technology https://journals.su.edu.ye/index.php/jast
Vol. 1 | No. 3 | Page 291 – 303 | 2023 | ISSN : 2958-9568
Doi: 10.59628/jast.v1i3.314

Developing a Model for Offline Signature Verification Using CNN Architectures and Genetic Algorithm

Abdulbaset M. Qaid Musleh [1,*]., Abdoulwase M. Obaid Al-Azzani[1]
[1] Department of Computer Science, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen.
***Corresponding author:** aledresi@su.edu.ye

ARTICLE INFO
Article history:
Received: March 13, 2023
Accepted: July 30, 2023
Published:, August, 2023

KEYWORDS
1. Offline Signature Verification
2. Convolutional Neural Network (CNN)
3. Deep Learning (DL)
4. Genetic Algorithm (GA)

ABSTRACT
The signature verification process has many applications, such as its use in financial operations, providing the electronic signature of documents, and providing an additional confidentiality standard to verify the identity of users in computer systems. This process has the advantage of being accepted by the community and is less intrusive compared to other biological methods. Deep learning (DL) and CNN (Convolutional Neural Network) are widely used by bioinformaticians. Due to the difficulty in extracting features in other systems or models, DL and CNN-based signature verification systems have been significantly improved. Yet Hyperparameter optimization for CNN models remains a challenging problem in designing highly efficient models with the most accurate results. It is often convolutional neural network (CNN) models that are manually designed. The proposed method is focused on a genetic algorithm that develops a population of CNN models in order to find the best fit architecture for designing an offline signature verification model. Our model is tested on more than one dataset, BHSig260-Bengali, BHSig260-Hindiin, GPDS, and CEDAR. The result of the approach proposed in this paper has the highest discrimination rate of FRR of 2.5, FAR 3.2, EER 2.35, and 97.73 %-accuracy rate.

CONTENTS

1. Introduction
2. Problem Definition
3. Related Work
4. Materials And Methods
5. Training And Testing Stage
6. Experimental Results
7. Conclusion.

1. Introduction:

Biometric systems have been used to identify individuals based on behavioral or physiological characteristics [39]. Verification, identification, and watchlists are primarily and continuously used in biometrics. A handwritten signature is one of the most widely used tools in biometrics for verifying secure personal authentication [40]. Signature verification has been extensively studied over the last two decades, and it is still a promising area of research. Previous studies on handwritten signature image verification have categorized it into two main fields: online and offline [41]. Online signature verification focuses on dynamic information captured during the writing process and does not consider static information, such as pen-up time, acceleration, or velocity. In contrast, offline signature verification deals with static signature images and is generally considered more challenging, resulting in lower accuracy than online signature verification [1].

Word count: 4,996 words, excluding references.

Funding Statement: The study was supported by grant NN from the Foundation of Basic Research. This work was carried out under research program NNN of NN University. Author NN was supported by grant NN from the Ministry of NN.

Ethical Compliance: All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

Data Access Statement: Research data supporting this publication are available from the NN repository at located at www.NNN.org/download/.

Conflict of Interest declaration: The authors declare that they have NO affiliations with or involvement in any organization or entity with any financial interest in the subject matter or materials discussed in this manuscript.

Author Contributions: AB and MJ contributed to the design and implementation of the research, JK to the analysis of the results and to the writing of the manuscript. VK conceived the original and supervised the project.

Despite its low accuracy, offline signature verification offers several advantages. It does not rely on special input devices, making it accessible and applicable to a wide range of scenarios. In addition, offline signature verification encompasses various domains, further expanding its potential applications and relevance. Thousands of financial documents and business transactions are continuously authorized through signatures [2]. Hence, the main purpose of handwriting signature verification systems is to detect genuine signatures (made by the owner writer) or forged signatures (made by a fraudulent person) [3]. Generally, three types of forgeries exist in the signature verification field [4]. Unskilled forgery can be committed by a person who forges another person's signature without knowing any information about that person. Random forgery can be made by a person who knows the signer's name without seeing the genuine signature. Skilled forgery can be made by a person who knows the name and shape of the genuine signature of the owner. Figure 1 depicts the forgery of three types of handwritten signatures.

Figure 1: (a) Signature of Genuine (b) Forgery of Random (c) Forgery of Unskilled (d) Forgery of Skilled.

Two classifications of learning are used in systems of Handwritten Signature Verification: The writer-independent (WI) and another is known as writer-dependent (WD) [5], [6]. The learning that is performed by all signatures in the database is known as the Writer-Independent (WI) state, but when the learning is performed by individual signatures independently, this is known as the Writer-Dependent (WD) state. In contrast, this made the WI method more popular and was used when building the WI system; it is easy to add a new person because the classification in the system depends on one category for all persons [7], [8]. Recently, many automated systems have been designed to verify the validity of handwritten signatures through different algorithms and methods, and deep learning has the largest share of these systems. This is because of the efficiency and ability of deep learning to classify images using different methods and techniques in image processing. The best technique for deep learning is Convolutional Neural Networks [9], [10]. There is a significant difference in the efficiency of many deep learning approaches in real-life applications [11], [12]. The performance and efficiency of CNNs rely heavily on their architectures [13], [14]. Therefore, many experts in the field of deep learning have designed several different structures for CNN, and they are named after their different names and versions. VGGNet, GoogleNet [15], ResNet [16], CapsNet [17], and DenseNet [18]. However, it is difficult to obtain a CNN model that solves all classification problems. Therefore, the design of the architecture CNN

depends solely on suitable and specific parameters to solve the problem by manual design through repeated attempts until the best results are obtained. However, this process can take longer than expected [19]. The method proposed in this paper uses a genetic algorithm for hypermetric optimization of the CNN architecture for the offline signature verification problem.

2. Problem Definition

The signature identifies and proves the signer's identity. As a result, a person may face some difficulties in deciding whether the two signatures are identical owing to work pressure and some ambiguity associated with manual signatures [40]. During commercial transactions, there is a growing need in the public and private sectors, especially in the banking system, for an efficient and accurate automated system capable of verifying the genuine signature and detecting skillful forgery of the signature [42]. The main problem is the automated validation of signatures and signature feature extraction, which enables the system to differentiate between genuine and forged signatures [43]. Therefore, this study aims to highlight the effective signature features and verify whether the system can automatically verify the real signature and detect skilled forgery, where the method depends on optimizing the CNN model hyperparameters using a Genetic Algorithm. A genetic algorithm was used to optimize the CNN architecture for offline signature verification and forgery protection problems. The genetic algorithm helps us select the best hyperparameters to build the best CNN model.

3. Related Work

The science of artificial intelligence and its modern and contemporary techniques, including deep learning, have been used in many applications, such as pattern recognition and natural language processing, the most important of which is computer vision [21]. Convolutional neural networks consist of basic components: a Convolutional Layer, the Activating function, pooling layer, and fully connected layer. A filter (also known as a kernel) is applied in the convolution layer that determines the presence of certain features or patterns in the original image (the input image matrix), and subsequently, several filters can be used to extract different features. The idea behind the Pooling Layer is very simple: it reduces the size of large matrices, and the process is performed by applying one of the following two functions: Max and Average. The fully connected layer is the last layer in the convolutional network, and is a multilayer perceptron. The neurons were fully connected to all the nodes of the previous layer. The reason it exist, in the end is that the final classification process occurs in it. In other words, the components of the last array are converted into an array with only one column. The method proposed in [22] for offline signature verification uses the Siamese network, which is based on user writer-independent (WI) feature learning. Euclidean distance is used to measure the similarity and dissimilarity between pairs of Siamese network outputs. In addition, a Siamese Neural Network was used by [23] to build a signature verification system. The structures of two similar neural networks were trained and evaluated using the same data. The Siamese network structure helps reduce the volume of training data necessary for implementation. They observed that the system efficiency increased by 13%. Two models for predicting the strength of adhesively bonded joints were designed using CNN, the architecture CNN: one was manually developed, and the other was designed using a genetic algorithm to optimize the architecture. The results of the two models were then compared. They noted that the improved model using the genetic algorithm had a better result, as proposed in [24]. The following datasets, CIFAR10, MNIST, and Caltech256, were used to test the CNN architecture optimized models using a genetic algorithm to solve the image classification problem. The genetic algorithm worked by automatically adjusting the

parameters of the model, and through the testing process of the model, they found that the results were more accurate than those of other models that were tested with the same dataset [19]. In the method proposed in [25], a genetic algorithm was applied to choose the possible solutions by selecting the following parameters: many filters, filter size, and the number of layers automatically added to the trainable layers of the CNN transfer model. The results show that the proposed method achieved an accuracy of 97% in classifying cat and dog datasets over 15 generations. Reverse inference based on the results of the genetic algorithm showed that the proposed method can detect gradient features in network layers, which plays a role in understanding the transfer of CNN models. They developed a system [26] for the recognition of finger veins using a CNN with a genetic algorithm. The proposed system Genetic Algorithm with a convolutional neural network (GA-CNN) performs better in terms of accuracy, sensitivity, and precision. The ability of the Genetic Algorithm (GA) to search is utilized to start the training phase of a CNN before the model's training process, rather than random initializers for the weights of the network created using the genetic algorithm. In the testing process system, the results were 97.98% for sensitivity, 98.50% for accuracy, and 99.01%, respectively. The selection of the most suitable set of features for a signer remains an open question. Genetic algorithms (GA) have recently been used a genetic algorithm to select the optimal set of partial curves from the signature and features of each curve. The locations of the partial curves and the features contained in the partial curves used for verification are encoded into the chromosome [27]. The focus of the study proposed in [44] is on offline signature verification, in which four different pattern representation schemes have been implemented. To weigh the individual feature components, their pattern characterization capabilities were determined using Genetic Algorithms. The conclusions of the four subsystems, each based on a representation scheme, are combined to make a final decision regarding the validity of the signature. The experimental results indicate that verification accuracy increases when feature-based classifiers are combined. The model was built to validate signatures through the transfer of learning and activation functions for the three different CNN models (VGG16, VGG19, and ResNet50) with the addition of some parameters for each model, training, and testing on SigComp2009 dataset, showing that the VGG16 model has a high efficiency of 97%- compared to other models this approach was proposed by [28]. Thus, this study focuses on developing a system for offline signature verification using a CNN with a genetic algorithm that can search for the best model architecture hyperparameters.

4. Materials And Methods

Handwritten offline signature verification is a pattern recognition problem. For this purpose, the proposed model must be able to recognize and verify genuine handwritten offline signatures and detect the forgery skills that exist in handwritten offline signatures. Therefore, the offline signature verification model includes several stages. The main stages are preprocessing of the signature image, genetic algorithm for selecting the best hyperparameters, and CNN for feature extraction, training, and testing. Each stage consists of several steps. Figure 2 shows the main stages and steps of each stage of the model.

A. Signature Images Preprocessing

Before starting feature extraction, essential processes must be applied to the image signature. These operations include the following.

1) Convert the color images into grayscale.
2) Resize each image to the same size of 100 x 100 pixels.
3) Read the image points and store them in an image's matrix.
4) Store the image class name or label name for each image in the label's matrix.

B. The Genetic Algorithm

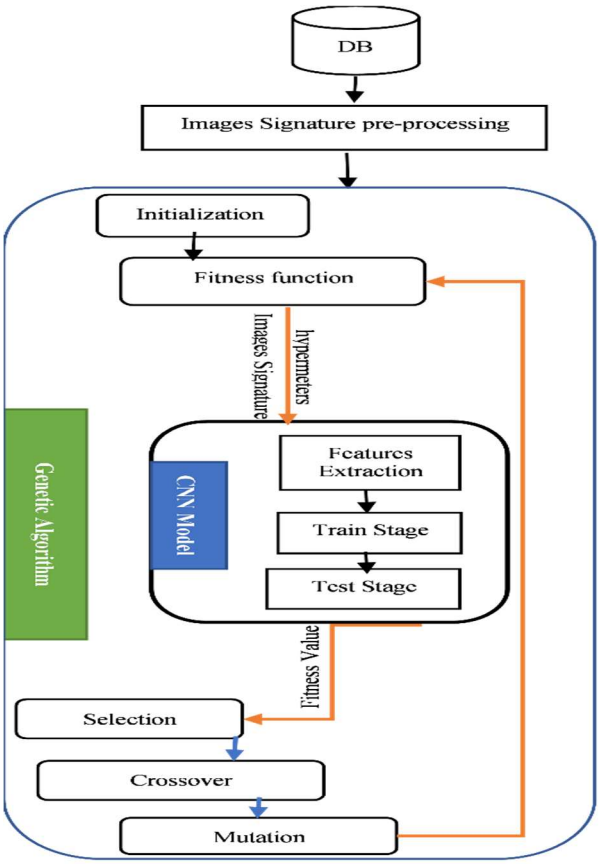

Figure2: Block Diagram for The System of Proposed.

Genetic algorithms are population-based search algorithms that are well suited for solving high-dimensional and non-convex optimization problems. In this study, the genetic algorithm helped us to optimize the best hyperparameters to build the best CNN model. However, the performance depends on the architecture used to solve the problem. The architecture of a CNN is often manually optimized by researchers, but a time-consuming process is difficult to obtain without previous knowledge of the CNN [19]. Each output chromosome from the genetic algorithm was tested on the (CNN) model to calculate the accuracy. The steps of the proposed algorithm are as follows:

1) Generate 6 initial chromosomes.
2) Compute the accuracy of each chromosome by CNN model.
3) Sort Chromosomes in descending order based on accuracy.
4) Select three-chromosome height accuracy.
5) Crossover Function is applied to chromosomes.
Child1 ← Crossover (Chrom1, Chrom3).
Child2 ← Crossover (Chrom2, Chrom1).
Child3 ← Crossover (Chrom3, Chrom2).
6) Mutation Function is applied to a new child.
Child1 ← Mutation (Child1).
Child2 ← Mutation (Child2).
Child3 ← Mutation (Child3).
7) Accuracy ←Compute the accuracy of each new child by CNN model.
8) If Accuracy >= 0.99 or Generation = N.
9) Stop: 10) Else 11) Repeat steps 2 - 7 12).
End.

Table (1): Hyperparameter range

Hyperparameter	Range
Epoch	Random (2, 25)
Filter Size	Choice (16,32,64, 96)
Kernel Size	Choice [(3x3), (5x5)(7x7)]
Unit	Choice (128, 256,512)
Dropout	Choice (0,25, 0.50)

Table (2): Random Generated Initial Population

layer 1	layer 2	layer 3	layer 4	Epoch	Dropout	Flatten	Fitness%				
Filters	Kernel	Filters	Kernel	Filters	Kernel	Filters	Kernel	Epoch	Dropout	Flatten	Fitness%
32	3	32	3	64	5	32	5	10	0.50	128	60
64	3	96	7	32	3	32	3	15	0.25	256	58
64	3	64	3	64	5	64	7	12	0.25	512	70
32	3	32	5	96	3	32	5	18	0.50	512	54
64	3	64	3	16	3	16	3	7	0.25	256	49
32	3	32	3	32	5	32	3	9	0.25	128	50

1. Initialization

The initial population is a set of randomly generated chromosomes, as listed in Table 1. Each line in Table 2 represents one of the solutions generated by the genetic algorithm. Six child chromosomes in the initial population were generated using a genetic algorithm. Each chromosome is represented by 11 genes, eight pairing genes, epoch, dropout, and a fully connected layer. Each gene represents the structure of the convolutional layer in the CNN model. For example, chromosome32,3,16,3,32,5,64,7,21,0.25,128is a representation of the parameters for a CNN structure with four convolution layers. The first layer contains 32 filters with a kernel size of 3 × 3, the second layer contains 16 filters with a kernel size of 3 × 3, the third layer consists of 32 filters with a kernel size of 5 × 5, the fourth layer compares 64 filters with a kernel size of 7 × 7, 21 is referred to as the epoch number, too 0.25 means a dropout ratio, and 128 is fully connected in the CNN model.

2. Fitness Function

The CNN model was trained and evaluated using the training and testing data from the dataset, and the CNN model accuracy was calculated for each chromosome. The parents in the next step are selected based on the maximum accuracy value. In this process, each chromosome is input to the proposed CNN model with the dataset for training and testing the model to evaluate the performance and calculate the fitness value.

3. Selection

Three chromosomes are selected from the six current chromosomes to generate a new generation, and the selection process is based on the optimization function in which the addition of each chromosome has priority. The optimization function is a mathematical equation that varies from one problem to another, and is appropriate for solving the problem. In this method, selection depends on the chromosomes with the highest accuracy. The three most accurate chromosomes were selected from the originals of six chromosomes. Figure 3 shows the selection process.

Select 1	32	3	32	3	64	5	32	5	10	0.50	128
Select 2	64	3	96	7	32	3	32	3	15	0.25	256
Select 3	64	3	64	3	64	5	64	7	12	0.25	512

Figure 3: Selection Process.

4. Crossover

The process of mating between the parents selected from the previous process to obtain new children through the access point, as the main objective of this process, is to gather and identify information from two different sources and introduce a new product that is different from them. It selects half of the values from parent I and the rest from parent II to generate a new population. The crossover process is illustrated in Figure 4.

Cross 1	32	3	32	3	64	5	64	7	12	0.25	512
Cross 2	64	3	96	7	32	5	32	5	10	0.50	128
Cross 3	64	3	64	3	64	3	32	3	15	0.25	256

Figure 4: Crossover Process.

5. Mutation

When generating new children in the previous process, the mutation process involves a sudden change in the chromosome by changing one of the genes randomly to ensure that a different chromosome is obtained from its predecessor [29]. In the proposed method, randomly changing genes (epochs and neurons in the flattened layer). The mutation process is illustrated in Figure 5.

Muta 1	32	3	32	3	64	5	64	7	15	0.25	512
Muta 2	64	3	96	7	32	5	32	5	13	0.50	512
Muta 3	64	3	64	3	64	3	32	3	5	0.25	128

Figure 5: Mutation Process.

C. The Feature Extraction

The biggest challenge in solving the problem of handwritten signature verification is feature extraction, which enables the system to differentiate between real and forged signatures [20]. The convolutional neural network can learn different features in each stage based on the layer-by-layer advancement and finally realize related verification and classification functions [30]. Convolutional Neural Networks

are particularly suitable architectures for signature verification [31]. The proposed model consists of four convolution layers: the convolution layer, in which the size of the input image, grayscale dimensions (100 × 100 × 1) of the kernel, and number of filters are specified. The filter of a filter weight array of a certain size, for example, (3 × 3), wraps the signature image, starting from the upper left corner, passing through a certain number of rows, until it reaches a corner at the bottom right of the image. This process repeats the number of filters entering the layer with a change in filter content each time. The equation below describes the size of the feature map (output layer) as the sum of [the receptive field array for each filter position multiplied by the filter weight array]:

$$Feature\ Map = \sum_{k=1}^{n} ReciptiveFiledk * Filter \qquad (1)$$

This implies that if the number of filters used in the layer is 32, the result of the equation will be 100 × 100 × 32 and equal to 320,000. After creating the Convolution Layers and applying the activation functions to them as the ReLU function, a method called max-pooling, also called downsampling or subsampling, is used. Reduce the size of the output matrix and obtain only the values of the greatest importance. However, a filter is usually assigned with dimensions of 2 × 2 and a pass-through step = 2 pixels, and it is applied to the feature map, where the filter extracts the maximum value within each effective area. The activation function is a nonlinear function that is considered a bridge to the next layer in the convolutional neural network. Its usefulness lies in reducing the number of computations performed by not activating all the feature map points at the same time; in other words, activating the points that represent the features and excluding the points that do not represent them. It has several types, the most famous of which are Tanh [10], ReLU[33], Softmax [34], and Sigmoid [38], which will be used in this search. The fully connected layer has full connections to all activations in the previous layer, as observed in regular neural networks. Their activation can be computed with matrix multiplication followed by a bias offset. Figure 6 shows the CNN architecture of the proposed model.

The model CNN proposed steps are:

1) Create a Sequential model.
2) **Add a Conv2D layer** to the model with a specified number of filters (32), kernel size (3 × 3), and activation function (activation). The input shape was set to the specified input shape (100x100x32).
3) **Add another Conv2D layer** to the model with a specified number of filters (16), kernel size (3 × 3), padding, and activation function (activation).
4) **Add a MaxPooling2D layer** was added to the model with a specified pool size (2 × 2).
5) **Add another Conv2D layer** to the model with a specified number of filters (32), kernel size (5 × 5), padding, and activation function (activation).
6) **Add another MaxPooling2D layer** was added to the model with a specified pool size (2 × 2).
7) **Add another Conv2D layer** to the model with a specified number of filters (64), kernel size (7 × 7), padding, and activation function (activation).
8) **Add another MaxPooling2D layer** was added to the model with a specified pool size (2 × 2).
9) **Add a dropout layer** was added to the model with a specified dropout rate (0.25).
10) **Add a Flatten layer** to the model.
11) **Add a dense layer** was added to the model with a specified number of units (128) and activation function (activation).
12) **Add another Dense layer** to the model with a specified number of classes (number of classes) and activation function (activation).

13) **Compile the model** is compiled with the specified optimizer (optimizer), loss function (loss), and metrics.
14) **Train the model** was trained on the training data (x_train and y_train) with a specified batch size number of epochs (21).
15) **Evaluate the performance** for a model with data (x_test and y_test).

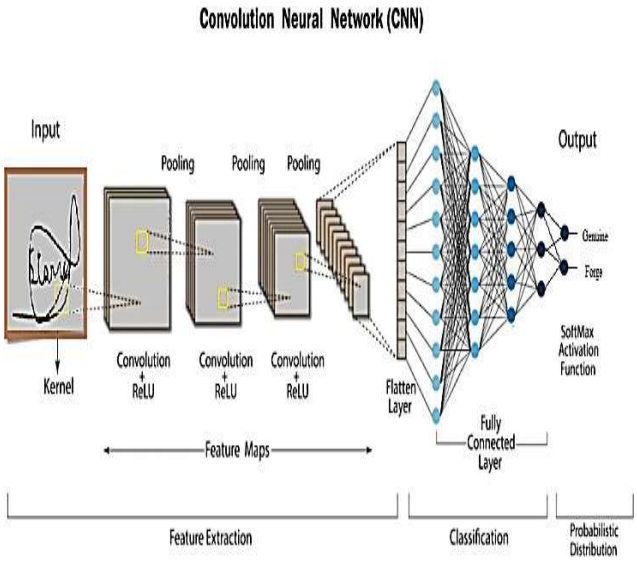

Figure 6: Block Diagram of CNN Proposed Architecture.

5. Training and Testing Stage

At this stage, the genetic algorithm sends the generated chromosomes represented by the parameters to the CNN model. Each chromosome contains a set of values called parameters, where the parameters represent the number of filters for four layers in the CNN model, the size of the kernel for four layers, the number of neurons exiting from the layers to the fully connected layer, and the number of cycles for this solution is called EPOCH. These parameters are fed to the model with the training image array, and then the efficiency of the model is evaluated using the test dataset to obtain the accuracy that represents the specific solution of the parameters sent from the genetic algorithm. This precision was then sent to Genetic Algorithm for its value. The fitness is a function that determines the efficiency of the model. If the number of proposed generations for the GA is completed, the best accuracy obtained is chosen to preserve the weights produced from these parameters in the CNN. Based on the previous steps in our proposed model, we obtained the highest possible accuracy of 97.7%. The proposed system was trained and tested with more than one dataset: (GPDS-160, GPDS-300, CEDAR, BHSig260Hindi, and BHSig260Bengali). In the GPDS, the dataset consisted of 24 genuine signatures and 30 simulated forgeries from 160 and 300 individuals. Fourteen genuine and 20 forgeries signatures from each individual were used in the training stage, and 10 genuine and 10 forgeries signatures were used in the test stage [37]. The CEDAR dataset collected 2640 total signatures images for 55 persons. Each participant had 24 genuine and 24 forged signatures. Fourteen genuine and 14 forgeries signatures from each individual were used in the training stage, and the remaining images were used for testing. The BHSig260-Bengali and BHSig260-Hindi datasets contain 24 genuine signatures and 30 simulated forgeries, with 100 people in BHSig260-Bengali and 160 persons in BHSig260-Hindi. Fourteen genuine and 20 forgeries signatures from each individual were used in the training stage, and 10 genuine and 10 forgeries signatures were used in the testing stage. Table 3 presents the details of the datasets used in the proposed system. A genetic algorithm selects the best hyperparameters for each dataset. The CNN model consists of four layers: the input layer, Conv1, Conv2, Conv3, and Conv4. The model includes a set of parameters that are used for signature image processing and verification of genuine handwriting signatures. These parameters are listed in table 4 according to the different datasets.

Table(3): Dataset Used in Proposed System

Type	GPDS-160	GPDS-300	CEDAR	Bengali	Hindi
Signers	160	300	100	100	160
Genuine	24	24	24	24	24
Forged	30	30	24	30	30
Training	5440	10200	1540	3400	5440
Testing	3200	6000	1100	2000	3200

Table(4): The Parameter Setting of The Proposed System.

Parameter	GPDS-300	CEDAR	BHSig260-B	BHSig260- H
Generations	7	2	5	1
Max Epochs	20	21	12	14
Parameters	2,402,731	2,545,911	2,426,120	1,241,250
Layer 1	Conv2D (32, 3x3)	(32, 3x3)	(64, 3x3)	(32, 3x3)
Layer 2 &MaxPool (2, 2)	Conv2D (32,3x3))	Conv2D (64,3x3)	Conv2D (32, 3x3)	Conv2D (32,3x3)
Layer 3 &MaxPool (2, 2)	Conv2D (32, 3x3)	Conv2D (32,3x3)	Conv2D (64, 3x3)	Conv2D (32,3x3)
Layer 4 &MaxPool (2, 2)	Conv2D (32, 3x3)	Conv2D (64,3x3)	Conv2D (32, 3x3)	Conv2D (32,3x3)
Dropout	0.50	0.25	0.25	0.50
Flatten	512	264	512	256
Accuracy	0.93	0.977	0.958	0.922

6. Experimental Results

Each dataset was subdivided into two parts: training set and testing. The performance of the proposed system was measured using three global scales which are as follows: Accuracy is the ratio of the number of correctly categorized signatures to the total number of complete signatures. These are the False Acceptance Rate (FAR) and False Rejection Rate (FRR), where FAR is the presence of the forgeries signature that are incorrectly classified. An equal Error Rate (EER) is applied to evaluate the equilibrium point where the FRR equals the FAR. A lower EER indicates better performance for the model). The results obtained from the proposed method of constructing the CNN model using the genetic algorithm were compared with those of other methods using hand-built CNN models. The results were compared with those of other studies. Figures (7,8,9, and 10) show the graphs of the curves of the results of the proposed study, where the curves are loss, val_loss, val_accuracy, and accuracy with the used dataset.

Table (5): Comparison Results for CEDAR Dataset.

Method	CEDAR			
	FAR	FRR	EER	Accuracy%
Surroundedness Features [[7]]	8.33	8.33	8.33	91.67
Multi-Path Siamese (MA-SCN) [32]	19.21	18.35	18.92	80.75
Siamese CNN [30]	6.78	4.20	–	95.66
Our method	**2.5**	**2.2**	**2.35**	**97.73**

Table (6): Comparison Results for GPDS-160 and GPDS-300.

Method	GPDS-160- GPDS-300			
	FAR	FRR	EER	Accuracy%
CNN_GP [35]	9.08	20.60	12.83	92
GoogLeNet Inception-v1 and Inception-v3 [36]	–	–	26	72
Our method	**9.1**	**20**	**11**	**93**

Table (7): Comparison Results for BHSig260-B and BHSig260-H

Method	BHSig260- B				BHSig260- H			
	FAR	FRR	EER	Accuracy%	FAR	FRR	EER	Accuracy%
Multi-Path Siamese (MA-SCN) [32]	5.73	4.86	8.18	94.99	9.96	5.85	5.32	92
Siamese CNN [30]	14.25	6.41	–	90.64	12.29	9.6	–	88.98
Multi-scripted with CNN [6]	1.50	3.14	–	95	2.31	6.65	–	90
Our method	**1.3**	**2.1**	**1.7**	**95.82**	**6.8**	**4.7**	**5.2**	**92.26**

Table 5 shows our results for the (CEDAR) dataset using two methods: the Multi-Path Attention Siamese Convolution Network (MA-SCN) model [32] and A Two-Stage Siamese Network Model for Offline Handwritten Signature Verification [30]. Surroundedness features study using local binary pattern (LBP) and uniform Local Binary Patterns (ULBP) [7].

The results in table 6 compare our system with other systems using the GPDS-160 and GPDS-300 datasets. The results in Table 7 are displayed as a comparison of some of the study's results with our results for BHSig260-B and BHSig260-H datasets.

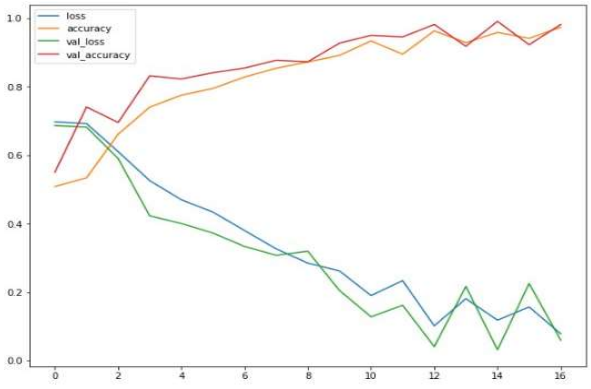

Figure 7 : The Resulting CEDAR Dataset

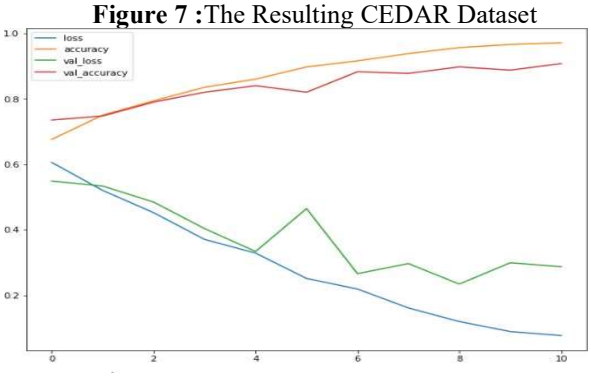

Figure 8 : The Resulting GPDS-300

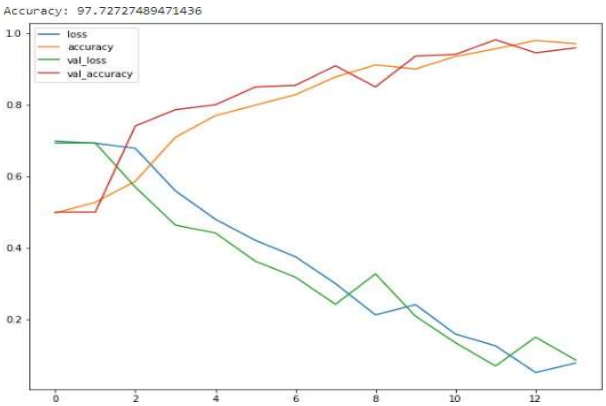

Figure 9 :The Resulting BHSig260-B

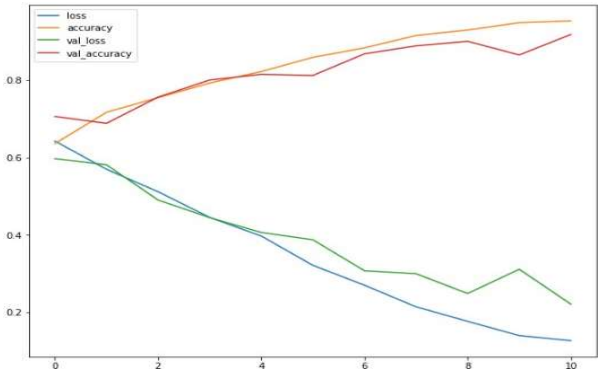

Figure 10 : The Resulting BHSig260-B

7. Conclusion

This study focuses on the development of an offline signature verification model using a genetic algorithm and convolutional neural networks (CNNs). The goal was to optimize the CNN architecture and hyperparameters to achieve highly efficient models with accurate results. The proposed method was tested on multiple datasets including BHSig260-Bengali, BHSig260-Hindiin, GPDS, and CEDAR.

The results obtained from the approach presented in this study demonstrate a promising

performance. The model achieved a high discrimination rate with a False Rejection Rate (FRR) of 2.5, False Acceptance Rate (FAR) of 3.2, Equal Error Rate (EER) of 2.35, and an accuracy rate of 97.73%.

The use of deep learning and CNN-based signature verification systems has proven to be effective, providing a less intrusive and widely accepted method for verifying signatures. The genetic algorithm plays a crucial role in developing a population of CNN models and finding the best-fit architecture for the offline signature verification task.

Overall, the findings of this study highlight the potential of utilizing genetic algorithms and CNNs in signature verification systems. The optimized model presented in this manuscript demonstrates promising results and contributes to the advancement of automated signature verification techniques. Further research and improvements in this field can lead to enhanced security measures, efficient document handling, and increased user confidence in various applications such as financial operations and computer systems.

8. References

[1] M. A. Ferrer and C. M. Travieso, "Offline signature verification: An overview and some recent advances," Pattern Recognition Letters, vol. 34, no. 3, pp. 249-256, 2013.

[2] B. M. Al-Maqaleh and A. M. Musleh, "An efficient offline signature verification system using local features," International Journal of Computer Applications, vol. 131, no. 10, pp. 39-44, 2015.

[3] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Offline handwritten signature verification—literature review," in Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Nov. 1-8, 2017, IEEE.

[4] R. Verma and D. Rao, "Offline signature verification and identification using angle feature and pixel density feature and both methods together," International Journal of Soft Computing and Engineering, vol. 2, no. 4, pp. 740-746, 2013.

[5] Y. Muhtar, W. Kang, A. Rexit, and K. Ubul, "A Survey of Offline Handwritten Signature Verification Based on Deep Learning," in 2022 3rd International Conference on Pattern Recognition and Machine Learning in PRML, July 2022, IEEE, pp. 391-397.

[6] T. Longjam, D. R. Kisku, and P. Gupta, "Multi-scripted Writer Independent Off-line Signature Verification using Convolutional Neural Network," Multimedia Tools and Applications, pp. 1-18, Aug. 2022.

[7] S. Pal, A. Alaei, U. Pal, and M. Blumenstein, "Performance of an off-line signature verification method based on texture features on a large indicscript signature dataset," in 12th IAPR Workshop on Document Analysis Systems, pp. 72-77, IEEE, April 2016.

[8] A. Foroozandeh, A. Hemmat, and H. Rabbani, "Offline handwritten signature verification and recognition based on deep transfer learning," in International Conference on Machine Vision and Image Processing (MVIP), IEEE, Feb. 2020, pp. 1-7.

[9] N. Sharma, S. Gupta, P. Mehta, X. Cheng, A. Shankar, P. Singh, and S.R. Nayak, "Offline signature verification using deep neural network with application to computer vision," Journal of Electronic Imaging, vol. 31, no. 4, pp. 041210-1-041210-10, Jul. 2022.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, May 2015.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017.

[12] C. Clark and A. Storkey, "Training deep convolutional neural networks to play go," in International Conference on Machine Learning, PMLR, Jun. 2015, pp. 1766-1774.

[13] N. Purohit, Sapna Purohit, and C. S. Satsangi, "Offline Handwritten Sig- nature Verification Using Template Matching and Clustering Technique," International Journal of Computer Science and Mobile Computing, vol. 2, no. 4, pp. 295-301, 2014.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint, pp.1409-1556, 2014.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[17] E. Parcham, M. Ilbeygi, and M. Amini, "CBCapsNet: A novel writer- independent offline signature verification model using a CNN-based

architecture and capsule neural networks," Expert Systems with Applications, p. 115649, Dec. 2021.

[18] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 2017, pp. 2261-2269.

[19] F. Johnson, A. Valderrama, C. Valle, B. Crawford, R. Soto, and R. Ñanculef, "Automating configuration of convolutional neural network hyperparameters using genetic algorithm," IEEE Access, vol. 8, pp. 156139- 156152, 2020.

[20] V. Malekian, A. Aghaei, M. Rezaeian, and M. Alian, "Rapid off-line signature verification based on signature envelope and adaptive density partitioning," in 2013 First Iranian Conference on Pattern Recognition and Image Analysis (PRIA), IEEE, Mar. 2013, pp. 1-6.

[21] J. Donahue, L. Anne, S. Guadarrama, M. Venugopalan, S. Rohrbach, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2625-2634.

[22] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," arXiv preprint arXiv:1707.02131, 2017.

[23] C. Yinka-Banjo and C. Okoli, "Signature Verification Using Siamese Convolutional Neural Networks," Covenant Journal of Informatics and Communication Technology, 2019.

[24] E. G. Arhore, M. Yasaee, and I. Dayyani, "Optimisation of convolutional neural network architecture using genetic algorithm for the prediction of adhesively bonded joint strength," Structural and Multidisciplinary Optimization, vol. 65, no. 9, pp. 1-16, 2022.

[25] C. Li, J. Jiang, Y. Zhao, R. Li, E. Wang, X. Zhang, and K. Zhao, "Genetic Algorithm based hyper-parameters optimization for transfer Convolutional Neural Network," in International Conference on Advanced Algorithms and Neural Networks (AANN 2022), vol. 12285, SPIE, June 2022, pp. 232-241.

[26] O. M. Assim and A. M. Alkababji, "CNN and Genetic Algorithm for Finger Vein Recognition," in 2021 14th International Conference on Developments in Systems Engineering (DeSE), IEEE, Dec. 2021, pp. 503- 508.

[27] Xuhua Yang, Xianyi Zeng, Hongbo Fu, and Yaping Zhang, "Selection of features for signature verification using the genetic algorithm," Computers and Industrial Engineering, vol. 30, no. 4, pp. 1037-1045, 1996.

[28] D. P. Sudharshan and R. N. Vismaya, "Handwritten Signature Verification System using Deep Learning," in 2022 IEEE International Conference on Data Science and Information System (ICDSIS), IEEE, July 2022, pp. 1-5.

[29] A. S. Mondal, "Evolution of convolution neural network architectures using genetic algorithm," in 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, July 2020, pp. 1-8.

[30] W. Xiao and Y. Ding, "A Two-Stage Siamese Network Model for Offline Handwritten Signature Verification," Symmetry, vol. 14, no. 6, pp. 1216, 2022.

[31] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," Pattern Recognition, vol. 70, pp. 163-176, 2017.

[32] X. Zhang, Z. Wu, L. Xie, Y. Li, F. Li, and J. Zhang, "Multi-Path Siamese Convolution Network for Offline Handwritten Signature Verification," in 2022 The 8th International Conference on Computing and Data Engineering, Jan. 2022, pp. 51-58.

[33] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 807-814.

[34] I. Goodfellow, Y. Bengio, and A. Courville, Adaptive Computation and Machine Learning Series, 2016.

[35] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Writer-independent feature learning for offline signature verification using deep convolutional neural networks," in 2016 International Joint Conference on Neural Net- works (IJCNN), IEEE, July 2016, pp. 2576-2583.

[36] S. M. Sam, K. Kamardin, N. N. A. Sjarif, and N. Mohamed, "Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3," Proce- dia Computer Science, vol. 161, pp. 475-483, 2019.

[37] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordonez, "Robustness of offline signature verification based on gray level features," IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, pp. 966-977, 2012.

[38] Venkata Saichand, Nirmala Devi M, Arumugam S, and N. Mohankumar, "FPGA Realization of Activation Function for Artificial Neural Net- works," in IEEE Eighth International Conference on Intelligent Systems Design and Applications, 2008.

[39] A. K. Jain, A. Ross, and K. Nandakumar, Introduction to Biometrics, Springer, 2016.

[40] D. Impedovo and G. Pirlo, "Automatic signature verification: the state of the art," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 38, no. 5, pp. 609-635, 2008.
[41] C. L. Liu and Y. H. Yin, "Offline handwritten signature verification - literature review," Pattern Recognition, vol. 40, no. 8, pp. 2293-2307, 2007.
[42] S. Panigrahi, A. K. Tiwari, and R. Sharma, "A study of various off- line signature verification approaches," International Journal of Computer Science and Network Security (IJCSNS), vol. 12, no. 10, pp. 78-83, 2012.
[43] L. V. Batista, D. Rivard, R. Sabourin, and P. Maupin, "State of the art in off-line signature verification," in Iberoamerican Congress on Pattern Recognition, Springer, Berlin, Heidelberg, 2009, pp. 227-234. DOI: 10.1007/978-3-642-10268-4_28.
[44] V. E. Ramesh and M. Narasimha, "Off-line signature verification using genetically optimized weighted features," Pattern Recognition, vol. 32, no. 2, pp. 217-233, 1999.