

Article

Siamese Convolutional Neural Network-Based Twin Structure Model for Independent Offline Signature Verification

Neha Sharma ¹, Sheifali Gupta ¹, Heba G. Mohamed ² , Divya Anand ^{3,4}, Juan Luis Vidal Mazón ^{4,5} ,
Deepali Gupta ^{1,*}  and Nitin Goyal ⁶ 

¹ Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India

² Department of Electrical Engineering, College of Engineering, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

³ Department of Computer Science, Lovely Professional University, Phagwara 144001, Punjab, India

⁴ Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain

⁵ Department of Project Management, Universidad Internacional Iberoamericana, Campeche 24560, Mexico

⁶ Computer Science and Engineering Department, Shri Vishwakarma Skill University, Palwal 121102, Haryana, India

* Correspondence: deepali.gupta@chitkara.edu.in

Abstract: One of the toughest biometrics and document forensics problems is confirming a signature's authenticity and legal identity. A forgery may vary from a genuine signature by specific distortions. Therefore, it is necessary to continuously monitor crucial distinctions between real and forged signatures for secure work and economic growth, but this is particularly difficult in writer-independent tasks. We thus propose an innovative and sustainable writer-independent approach based on a Siamese neural network for offline signature verification. The Siamese network is a twin-like structure with shared weights and parameters. Similar and dissimilar images are exposed to this network, and the Euclidean distances between them are calculated. The distance is reduced for identical signatures, and the distance is increased for different signatures. Three datasets, namely GPDS, BHsig260 Hindi, and BHsig260 Bengali datasets, were tested in this work. The proposed model was analyzed by comparing the results of different parameters such as optimizers, batch size, and the number of epochs on all three datasets. The proposed Siamese neural network outperforms the GPDS synthetic dataset in the English language, with an accuracy of 92%. It also performs well for the Hindi and Bengali datasets while considering skilled forgeries.

Keywords: signature verification; two-channel; Siamese network; convolutional neural network; deep learning



Citation: Sharma, N.; Gupta, S.; Mohamed, H.G.; Anand, D.; Mazón, J.L.V.; Gupta, D.; Goyal, N. Siamese Convolutional Neural Network-Based Twin Structure Model for Independent Offline Signature Verification. *Sustainability* **2022**, *14*, 11484. <https://doi.org/10.3390/su141811484>

Academic Editor: Andreas Kanavos

Received: 15 July 2022

Accepted: 23 August 2022

Published: 13 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Biometric identifiers, such as a signature, have been used for centuries to validate a variety of human entities, including papers, forms, and bank checks. However, much effort is being made to eliminate the inherent ambiguity in the human verification process, making signature verification a significant study area in computer vision or pattern recognition [1]. There are two types of signature verification based on the input format: (1) online and (2) offline. Acquiring an online signature requires a digital writing pad and a digital stylus that may continuously monitor the electronic pen tip during signatures. Besides the signature's writing characteristics, these devices may also retrieve information such as writing speed, pressure, and so on, which is utilized in online systems. Offline signatures are often recorded by a device, usually a scanner or other imaging technology that provides 2D signature pictures. For decades, verification of signatures has been a critical issue for study, and significant hard work has been undertaken for offline and online signature verification purposes [2]. Verifying an offline signature is the only choice in many

circumstances, such as check dealings and document authentication. This work emphasizes the most challenging automated offline signature verification job due to its more extensive application range [3]. The primary goal of signature verification is to identify forgeries. Forgeries may be categorized as random, simple/casual, and skilled. The forger in random forgeries knows nothing about the user; they do not know their name or signature. As a result, he/she substitutes his/her signature with a legitimate one. In simple forgery, the forger is aware of the individual's name but is unaware of his/her writing style. In the case of skilled forgery, the forger has detailed knowledge of the person's name and signature and practices copying that signature. This forgery is more challenging to identify since it is precisely similar to a genuine signature [4]. Our goal is to present a Siamese network model based on the convolutional neural network that can distinguish between authentic signatures and skilled forgeries. There are primarily two approaches to the process of signature verification. The first is writer-dependent, and the second is writer-independent. The writer-independent situation is superior to writer-dependent, as a writer-dependent system requires constant updating (retraining) with each new writer (signer). This is highly costly for a system, such as a bank, where a new customer might create daily accounts. In the case of a writer-independent system, the general approach is to simulate the disparity between a genuine and a forged signature [5]. The signers are divided into training and testing sets when developing an authentication model in a writer-independent environment.

Signature images are grouped as similar (genuine–genuine) and dissimilar (genuine–forge) signature pairs for each signer. To balance the cases, equal numbers of similar and distinct pairs are randomly picked up from a single user's possible pairs. This technique is repeated for each signer in the training and testing sets to create the classifier's training and testing instances. For example, a signature verifier may be easily simulated using a Siamese neural network, consisting of identical convolutional neural networks that accept two unique signature pictures derived from the same or different users. The underlying CNN is then linked at the top by a cost function that calculates a distance metric among the maximum level of feature representations on either side of this neural network. As a result, the parameters of these twin networks are identical [6]. The main contribution presented in the manuscript is a two-channel convolutional Siamese network-based model proposed to solve the challenges of offline signature verification in a distributed environment. The proposed model's performance was simulated using different parameters such as batch size, optimizer, and number of epochs. The proposed model was evaluated for three languages: Hindi, Bengali, and English.

The remainder of this paper is laid out as follows. The related work is described in Section 2. Section 3 represents the proposed Siamese neural network for offline signature verification. Section 4 illustrates the experimental setup and results. Section 5 defines the state of the art results. Finally, Section 6 concludes the whole paper.

2. Related Work

Signature verification can be performed using either handcrafted features or deep learning. Many handcrafted features are available for offline signature verification, such as block codes or wavelets and the Fourier series [7]. Many of them use the global signature picture [7]. Other methods consider the location, tangent path, blob organization, linked component, and the curvature of local features [8]. Projection and contour-based techniques are also widely used for signature authentication [9]. In addition, signature verification has seen a rise in the use of approaches based on direction profiles [9,10], surroundedness features [11], grid-based methods [12], geometrical moment-based methods [13], and texture-based features [14]. Several structural strategies that consider the links between local variables are also examined for this purpose. Compact correlated features (CCFs) have recently been presented as another example [15]. Indrajit Bhattacharya et al. proposed a pixel matching technique-based offline signature verification and identification in 2013. First, the user's signature was compared to a sample signature kept in the database using PMT (pixel matching technique). In 2016, Assia Hamadene et al. [16] suggested a one-

class WI (writer-independent) approach with fewer references and feature distinction methods using a threshold for classification. The contourlet transform-based directional code co-occurrence matrix feature creation method is used in the suggested system [17]. In the same year, Hannes Rantzsch et al. developed a new technique for offline signature verification, which is writer-independent. Deep metric learning was used to create the signature embedding approach. By comparing genuine and faked signature triplets, the scheme learned to insert signatures in a high-dimensional space and use the Euclidean distance to measure their similarity.

Deep learning techniques such as Siamese-like networks are widely used for many verification tasks, including online signature verification [17], face verification [18,19], etc. In addition, one-shot image identification and sketch-based image retrieval tasks have also been implemented [20,21]. Sounak Dey et al. published a writer-independent system for offline signature verification jobs using a Siamese network based on convolution networks in 2017. Siamese networks can be defined as two identical sub-networks with the same weights used to learn a feature space with similar observations clustered together. This was accomplished by revealing the network to a similar and dissimilar pair of statements and reducing the Euclidean distance between the identical pairs while increasing it between different pairs [22]. In 2019, Ramesh Kumar Mohapatra et al. developed a method for learning characteristics from pre-processed authentic and forging signatures using convolutional neural networks (CNNs). The architecture leveraged the concept of having several filters on the same level to make the network larger rather than deeper. The suggested model was evaluated on CEDAR, the BH-Sig260 signature, and UTSig, all publicly available datasets [23]. In 2019, Jahandad et al. introduced a method based on the GPDS synthetic signature, the largest handwritten signature dataset. The dataset was used to classify 1000 users' signatures, with 24 original (or authentic) signatures and 30 forged (or fraudulent) signatures for each user. In addition, CNN Inception-v1 and Inception-v3 were employed, which are two popular GoogLeNet architecture versions [24]. Amruta B. Jagtap et al. proposed a work in 2020 that represented a Siamese neural network utilizing a convolutional neural network as a sub-network. They proposed adding specific statistical features and then the contrastive loss function to the Siamese network's embedding vector. The suggested network outperforms existing solutions in accuracy, FAR, and FRR [25].

It can be concluded from the literature that many techniques have been implemented for verifying offline signatures, either handcrafted techniques or deep learning algorithms. Although much has been accomplished in this area, there is still potential to improve the verification system. In addition, the performance parameters can be increased. We thus propose an approach for a more accurate classification of offline signatures.

3. Proposed Methodology

This section outlines the proposed methodology for verifying genuine and forged signatures. Section 3.1 details the datasets used in this work. Section 3.2 explains the pairing of genuine and forged signatures. Section 3.2 shows the pre-processing techniques applied to the dataset. Section 3.4 represents the proposed two channel-based Siamese neural network to verify genuine and forged signatures. Figure 1 depicts the block diagram of the proposed methodology for verifying offline signatures.

3.1. Dataset Description

To analyze our proposed signature verification approach, we considered three widely used benchmark databases: (1) GPDS Synthetic Signature Dataset (English), (2) BHSig260 (Hindi) signature corpus, and (3) BHSig260 (Bengali) signature corpus. Figure 2 shows some of the genuine and forged signatures from each dataset.

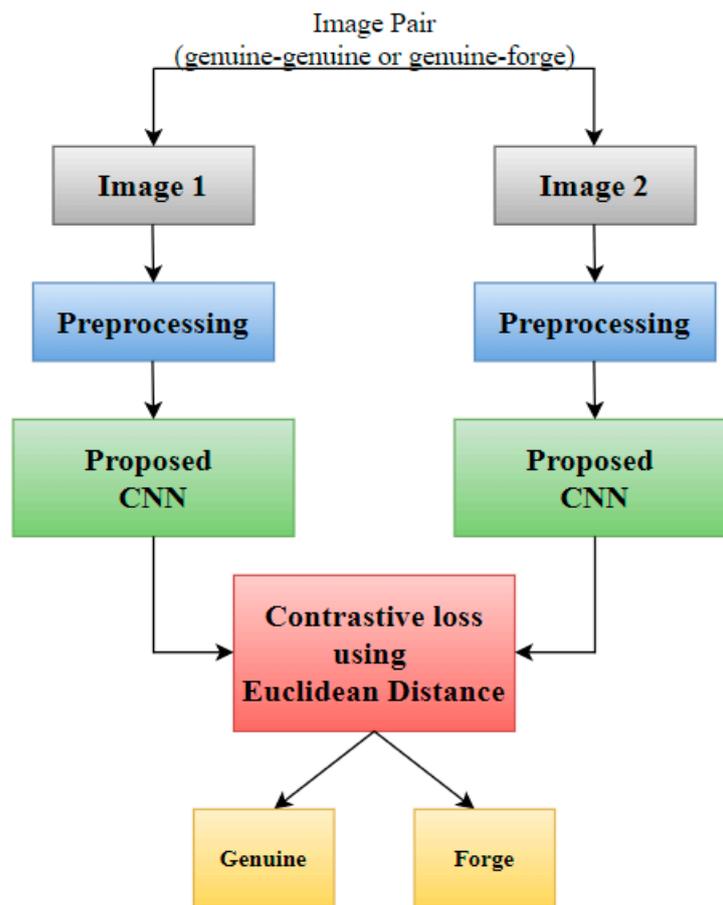


Figure 1. Block diagram of the proposed model.

Dataset Name	Genuine Image	Forge Image
(a) GPDS		
(b) Hindi	 आकृति अग्रवाल आयुष शर्मा	 आकृति अग्रवाल आयुष शर्मा
(c) Bengali	 তাজিনা খানম রাজশ্রী গুপ্ত	 তাজিনা খানম রাজশ্রী গুপ্ত

Figure 2. Sample genuine and forge images of datasets: (a) GPDS, (b) Hindi, and (c) Bengali.

3.1.1. GPDS Synthetic (English)

The University of Valencia in Valencia, Spain, provided the Synthetic Grupo de Proceso Digital de Seales (GPDS) dataset. A total of 4000 individuals comprise the GPDS Synthetic dataset, with 24 real signatures of each individual and 30 skilled forgeries. As a result, there are 216,000 images in the collection of real and fake signatures. Different types of pens were used to sign each document. The collection's signatures are available in various sizes and are stored in the jpg format with a resolution equivalent to 600 dpi [26].

3.1.2. BHSig260 (Hindi)

There are 160 Hindi signatures in the BHSig260 (Hindi) signature collection [27]. The authors used the same process as GPDS to create these signatures. There are also 24 genuine and 30 fake signatures for each signer. There are $160 \times 24 = 3840$ real signatures and $160 \times 30 = 4800$ fake signatures. The Hindi dataset contains 8640 images in total.

3.1.3. BHSig260 (Bengali)

The BHSig260 signature collection contains 100 individuals' signatures in Bengali [27]. The authors generated these signatures using the same procedure as GPDS and BHSig260 Hindi, with 24 real signatures and 30 forged signatures provided for each signer. As a result, $100 \times 24 = 2400$ genuine signatures and $100 \times 30 = 3000$ forged signatures are produced, and the total number of images in the Bengali dataset is 5400.

To maintain balance in the datasets, 100 users from each dataset were utilized to evaluate the proposed model. Therefore, the total number of images used from each dataset is 5400.

3.2. Pairing of Signature Images

The proposed system is designed for writer-independent verification, so each dataset is divided into training, test, and validation sets. Each dataset's total number of users is 100; they are divided as 80, 10, and 10 for training, testing, and validation, respectively. Since the datasets contain 24 genuine and 30 forged signatures per user, ${}^{24}C_2 = 276$ signature pairs per user are genuine. Similarly, since all three datasets contain 30 forged signatures per user, $24 \times 30 = 720$ genuine–forge pairs can be selected for each signer. To match similar and dissimilar categories, randomly, only 300 (genuine–forge) signature image pairs were chosen for each writer in the dataset.

3.3. Pre-Processing

Pre-processing can be defined as a technique to improve data or make it more compatible with the model. In addition, pre-processing procedures can be used to enhance the quality of data. The pre-processing techniques used in this work are resizing and normalization.

3.3.1. Resizing

The dimensions of the signature photos used in this investigation range from 153×258 to 819×1137 . Because neural networks can only process pictures of the same size, all images were downsized using bilinear interpolation to 155×220 .

3.3.2. Normalization

The data were normalized, a common practice in CNN systems to ensure numerical stability. To achieve rescaling, the pixel values were multiplied by a factor of $1/255$ [24].

3.4. Proposed Siamese Deep Convolutional Neural Network

The Siamese network is a multilayer neural network composed of numerous convolutional layers with varying sizes of kernels interspersed with pooling layers that summarize and down-sample the output of the convolution layers before feeding it to the subsequent layers. Additionally, rectified linear units are employed to obtain nonlinearity. We used various convolutional kernels of size 3×3 in our experiment. Usually, a differentiable loss

function is employed to enable gradient descent and optimization of the network weights. Backpropagation is used to update the weights of different layers given a differentiable loss function. Because optimization cannot be used for all training data when the training set is vast, batch optimizations provide a viable alternative for network optimization.

The Siamese neural network architecture is a network design consisting of two identical subnetworks. The twin CNNs are configured identically, with similar parameters and weight sharing. Parameter updates are replicated across both subnetworks. The proposed model's architecture is depicted in Figure 3. This methodology has been effectively applied to reduce dimensionality in weak supervised metric learning [18] and to verify faces [19]. A loss function at the top connects these subnetworks by computing a similarity metric based on the Euclidean distance among the feature representations on either side of the Siamese network.

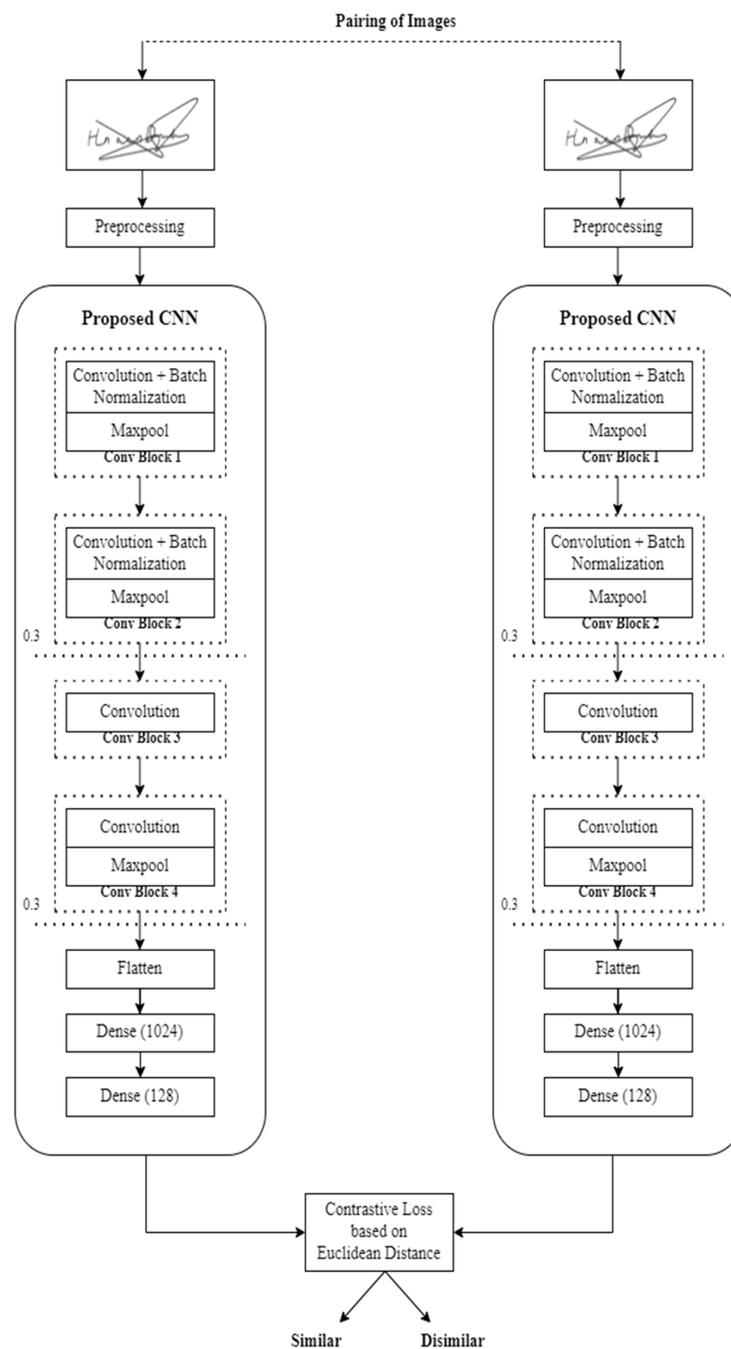


Figure 3. The architecture of the proposed model.

The contrastive loss [15] is a loss function frequently employed in Siamese networks.

$$L(c, s1, s2) = (1 - c) \frac{1}{2} (Dw)^2 + (c) \frac{1}{2} \{(\max(0, m - Dw))\}^2 \quad (1)$$

where c indicates whether or not two signature samples belong to the same category, $s1$ and $s2$ are signatures, m defines the margin between them, and Dw represents the Euclidean distance calculated in the feature space.

$$c = 0 \text{ if; } s1, s2 = \text{genuine, genuine} \quad (2)$$

$$c = 1 \text{ if; } s1, s2 = \text{genuine, forge} \quad (3)$$

where $s1$ and $s2$ are sample signatures; if both the signatures are genuine, they are labeled as 0, and if they are different, they are labeled as 1.

In contrast to conventional techniques that allocate binary similarity labels to the signature pairs, the Siamese network aims to bring the feature vectors nearer to input signature pairs labeled as similar and to push feature vectors away from input pairs labeled as dissimilar to improve classification accuracy. It is possible to think of each branch of the Siamese network as a task that embeds the input signature image into a spatial representation. Because of the loss function that has been chosen (Equation (1)), this space has the characteristic that signature images of the same class (authentic signature for a given writer) will be closer to each other than images of the same type but different writers (forgeries or signatures of different writers). It is connected to the other branches by a layer that calculates the Euclidean distance between the two points in the embedded space between them. Then, a threshold value on the distance between them must be determined to identify whether two photos belong to the same class (genuine, genuine) or a different class (real, forge).

Table 1 summarizes the proposed model's details and associated configurations. The suggested CNN model's fundamental architecture is composed of four convolutional blocks. Each block is constructed using a combination of convolutional layers, top pooling layers, and batch normalization. The number of filters following each convolution block was doubled. The first block, for example, has a convolution layer with 32 filters; the second block, 64 filters; the third convolution block, 128 filters; and the fourth convolution block, 256 filters. The rectified linear unit (ReLU) function activated each convolution layer.

Additionally, batch normalization was used for each convolutional and activation layer. It is a technique that enhances the performance of a neural network by normalizing the inputs in each layer to the point where the mean activation at the output is 0 and the deviation is 1. Additionally, the model contains two instances of the dropout layer. This layer sets input units to 0, randomly selected, with a rate-dependent frequency at each phase during the training period, preventing overfitting. Finally, following the final convolutional block, a fully linked network of two dense layers was created, the first with 1024 nodes and the second with 128 nodes, to account for the contrastive loss. This means that the most significant feature vector learned from either side of the CNN has a length of 128 units.

4. Experiments and Results

This section discusses the tests conducted with various parameters such as optimizers, batch size, and the number of epochs for Hindi, Bengali, and GPDS datasets and their outcomes.

4.1. Hyper Parameters

Along with understanding the network's design, it is critical to understand the model's implementation and performance, including the complexities of the training process itself. The network was built entirely from scratch without the aid of pre-trained transfer learning

models. Instead, the selection of training hyperparameters was based on experimentation (Table 2). The model's weights were set via Golort initialization [28].

Table 1. The architecture of the proposed Siamese model.

Block	Layers	Input Size	Filter Size	Activation Function	Output Size	No. of Parameters
Conv Block 1	Input image	$155 \times 220 \times 1$	—	—	—	—
	Conv1	$155 \times 220 \times 1$	3×3	ReLU	$77 \times 109 \times 32$	320
	Batch_normalization_1	$77 \times 109 \times 32$	—	—	$77 \times 109 \times 32$	128
	Max_pooling2d_1	$77 \times 109 \times 32$	3×3	—	$38 \times 54 \times 32$	0
	Zero_padding2d_1	$38 \times 54 \times 32$	—	—	$42 \times 58 \times 32$	0
Conv Block 2	Conv2	$42 \times 58 \times 32$	—	ReLU	$40 \times 56 \times 64$	18,496
	Batch_normalization_2	$40 \times 56 \times 64$	—	—	$40 \times 56 \times 64$	256
	Max_pooling2d_2	$40 \times 56 \times 64$	3×3	—	$19 \times 27 \times 64$	0
	Dropout_1	$19 \times 27 \times 64$	0.3	—	$19 \times 27 \times 64$	0
	Zero_padding2d_2	$19 \times 27 \times 64$	—	—	$21 \times 29 \times 64$	0
Conv Block 3	Conv3	$21 \times 29 \times 64$	3×3	ReLU	$19 \times 27 \times 128$	73,856
	Zero_padding2d_3	$19 \times 27 \times 128$	—	—	$21 \times 29 \times 128$	0
Conv Block 4	Conv4	$21 \times 29 \times 128$	3×3	ReLU	$19 \times 27 \times 256$	295,168
	Max_pooling2d_4	$19 \times 27 \times 256$	3×3	—	$9 \times 13 \times 256$	0
	Dropout_2	$9 \times 13 \times 256$	—	—	$9 \times 13 \times 256$	0
	Flatten	—	—	—	29,952	0
	Dense_1	—	—	—	1024	30,671,872
	Dense_2	—	—	—	128	131,200

Table 2. Values of hyperparameters.

Hyperparameters	Values
Initial Learning Rate (LR)	0.001
LR Schedule	LR \times 0.1
Weight Decay	0.0005
Fuzz Factor	0.00000001

The model was created using Python and the Keras Tensorflow Package. Keras is a freely available, easy-to-use framework optimized for neural network development. It is freely available and compatible with both Theano and Tensorflow. In addition, it is optimized for deep neural network computations. The Google Colab Platform was used to conduct all simulations in this work, with a Colab notebook equipped with Tensorflow and a GPU.

4.2. Analysis Based on Different Parameters

The proposed model was evaluated based on three different parameters. Firstly, the best optimizer was selected by comparing three different optimizers: RMS prop, Adam, and SGD. Then, three different batch sizes were compared: 128, 64, and 32. The final analysis was based on the number of epochs: 5, 10, and 15. This section contains all the results obtained from the analysis.

4.2.1. Accuracy Analysis Based on Different Optimizers

Three different optimizers were used for the evaluation of the proposed Siamese network, namely RMS [29], Adam [30], and SGD [31], on three different language datasets, Hindi, Bengali, and GPDS (English). Table 3 shows the accuracy obtained from different optimizers on all three datasets. The table shows that the best results were obtained using

RMS optimizers, and the worst results were obtained from Adam. The best accuracy is from RMS and the GPDS dataset, with a value of 92%.

Table 3. Accuracy of different optimizers.

Dataset Name	Optimizers		
	Adam	SGD	RMS
Hindi	0.50	0.67	0.78
Bengali	0.50	0.68	0.80
GPDS	0.65	0.80	0.92

4.2.2. Accuracy Analysis Based on Different Batch Sizes

After selecting the best optimizer, the analysis was performed by considering three different batch sizes: 32, 64, and 128. These three batch sizes were taken for the datasets. Table 4 shows the accuracy of the three different batch sizes for each dataset. From the table, it can be concluded that batch size 128 gives the best results for all three datasets. Batch 32 also gives similar results to 128, but the best results were achieved with batch size 128.

Table 4. Accuracy of different batch sizes.

Dataset Name	Batch Size		
	32	64	128
Hindi	0.77	0.76	0.78
Bengali	0.80	0.74	0.80
GPDS	0.88	0.75	0.92

4.2.3. Accuracy Analysis Based on Different Number of Epochs

The model was also analyzed by taking the different number of epochs. The model was run for 5, 10, and 15 epochs for all three datasets. Table 5 shows the results. As can be seen from the table, the best results were obtained by running the model for 10 epochs. If the model was run more than this, it shows a decrease in accuracy; thus, 10 epochs were chosen to run the model.

Table 5. Accuracy of different numbers of epochs.

Dataset Name	Number of Epochs		
	5	10	15
Hindi	0.77	0.78	0.75
Bengali	0.80	0.80	0.77
GPDS	0.85	0.92	0.92

4.3. Confusion Matrix Parameters for Best Optimized Model

The model's performance can be analyzed by comparing the different performance parameters, and the confusion matrix is shown in Figure 4 for the Hindi, Bengali, and GPDS datasets. The total number of image pairs in the test set for all three datasets is 5760. For example, in Figure 4a, representing the Hindi dataset, the total number of forge labels is 2395, and that of genuine labels is 3365. Of these, 2023 pairs were true positive, 2454 were true negative, 372 were false positive, and 911 were false negative.

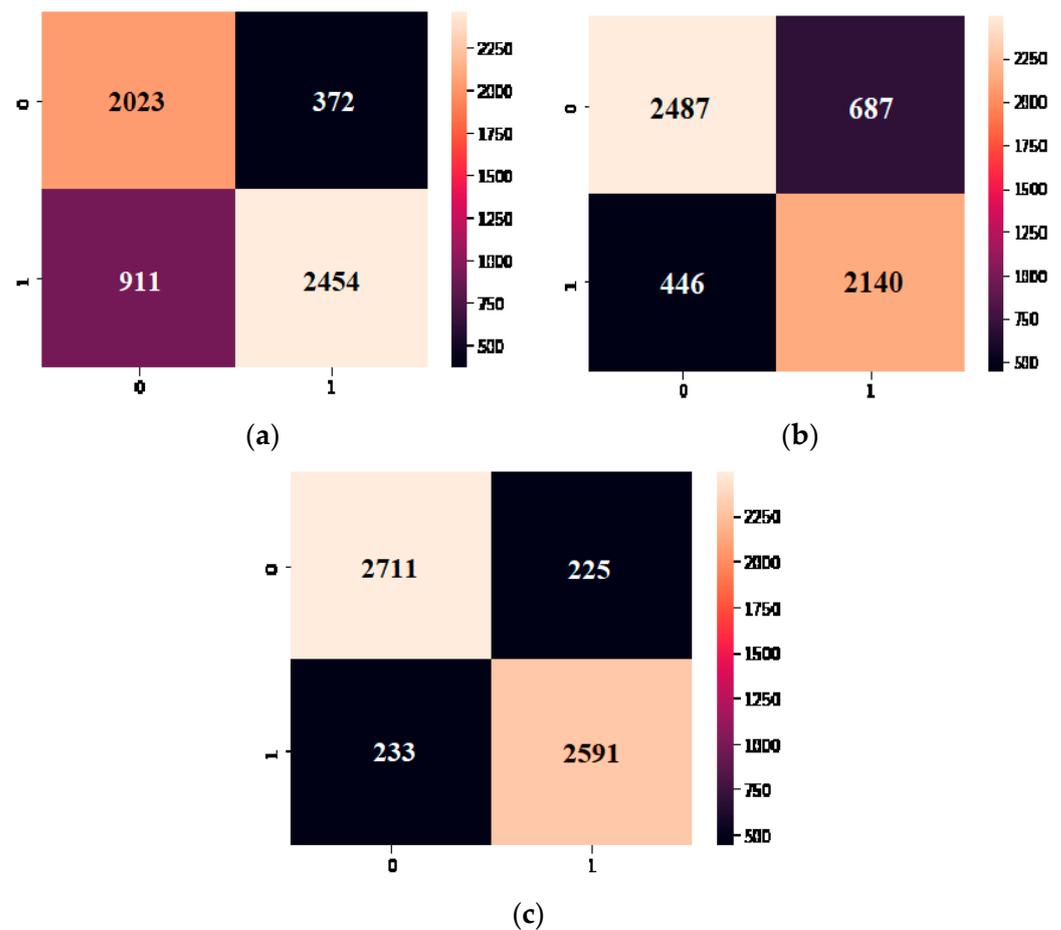


Figure 4. Confusion matrix: (a) Hindi, (b) Bengali, and (c) GPDS.

Figure 4b represents the Bengali dataset. Of 5760 image pairs, there are 3174 forged signatures and 2586 genuine ones. A total of 2487 images were true positive, 687 were false positive, 446 were false negative, and 2140 were true negative image pairs. In Figure 4c, representing the GPDS dataset, the values of genuine–forge pairs are 2936 and 2824. The values of true positive, false positive, false negative, and true negative are 2711, 225, 233, and 2591, respectively.

Table 6 shows the different parameters obtained by the proposed Siamese model for the three datasets. The performance parameters chosen for this study are precision, sensitivity, specificity, F1 score, and accuracy, as discussed in the previous section. It can be concluded from the table that the system gives the best results for the GPDS dataset in terms of all the performance parameters.

Table 6. Performance parameters analysis for all three datasets.

Dataset	Label	Precision	Sensitivity	Specificity	F1 Score	Accuracy
Hindi	Forge	0.69	0.84	0.81	0.76	78%
	Genuine	0.87	0.73	0.81	0.79	
Bengali	Forge	0.85	0.78	0.84	0.81	80%
	Genuine	0.76	0.83	0.82	0.79	
GPDS	Forge	0.92	0.90	0.91	0.92	92%
	Genuine	0.92	0.92	0.91	0.92	

Figure 5 shows the loss plot for the three datasets. Figure 5a shows the loss plot for the Hindi dataset. Testing loss is less than training loss for the Hindi dataset. Figure 5b shows the loss plots for the Bengali dataset. It can be seen that training and testing loss are almost similar for the Bengali dataset. Figure 5c shows the loss plots for the GPDS synthetic dataset. The training loss is less than the testing loss for this dataset, although the accuracy obtained by the GPDS dataset is the highest of all three datasets.

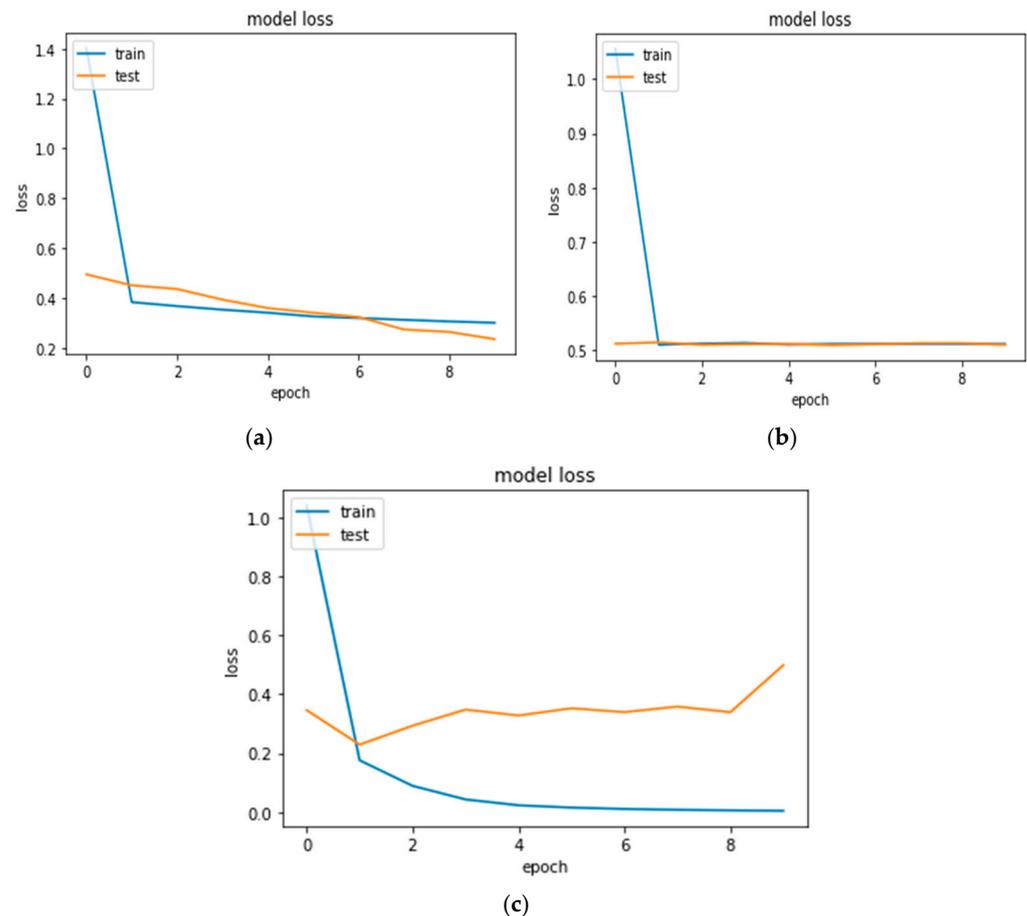


Figure 5. Loss plot for different datasets: (a) Hindi, (b) Bengali, and (c) GPDS.

5. State of the Art Comparison

Table 7 shows the state of the art comparison for offline signature verification. The proposed model has the maximum accuracy for the GPDS synthetic dataset. The Siamese model was first implemented in 2017, named SigNet, and the accuracy achieved by SigNet was 77.76% for the GPDS synthetic dataset. The Siamese model was also implemented in the year 2020 by Jagtap et al., and they obtained an accuracy of 86.47%. Our proposed Siamese model achieved an accuracy of 92% for the same (GPDS) dataset, which is the highest compared with the literature. Furthermore, the proposed model achieved accuracy of 80% and 78% for the Bengali and Hindi datasets, respectively.

Table 7. State of the art comparison.

References	Technique	Dataset Used	Accuracy Achieved
[6]	SigNet	GPDS synthetic, GPDS 300, Bengali, Hindi, CEDAR	77.76%, 88.79%, 86.11%, 85.90%, 100%
[22]	Siamese	GPDS synthetic, MCYT 75	84.58%, 85.38%
[23]	CNN (writer-independent and writer-dependent)	GPDS synthetic	62.5%, 75%
[24]	Inception v1 and v3	GPDS synthetic	83%, 75%
[25]	Siamese	GPDS synthetic, MCYT 75, CEDAR	86.47%, 88.49%, 100%
Proposed model	Siamese Neural Network	GPDS synthetic, Hindi, Bengali	92%, 78%, 80%

6. Conclusions and Future Work

We propose a method for offline signature verification based on Siamese networks that employ writer-independent feature learning. Unlike its predecessors, this technique does not rely on handcrafted characteristics; instead, it learns from data in a writer-independent situation. In the proposed technique, two similar CNNs were built with identical parameters. Then, the image pair (genuine–genuine or genuine–forge) was fed to the model. A contrastive loss using Euclidean distance was then calculated to classify the query image as genuine or forged. The novelty of this work is the CNN model design from scratch, which is used as the backbone of the Siamese network. Experiments were performed on the GPDS Synthetic, Bengali, and Hindi datasets, considering different parameters such as optimizers, batch size, and number of epochs. These experiments show that the proposed model achieves the highest accuracy in detecting skilled forgeries in the English-language GPDS synthetic dataset.

Furthermore, the proposed Siamese network outperformed the current state-of-the-art techniques for GPDS synthetic signature datasets. However, the performance of the proposed model is not satisfactory for the Hindi and Bengali datasets. In future work, we will try to improve the results for Hindi, Bengali, and other datasets.

Author Contributions: Conceptualization, N.S.; methodology, S.G.; validation, D.G.; investigation, H.G.M.; writing—original draft preparation, N.S.; writing—review and editing, D.A.; supervision, J.L.V.M.; project administration, N.G. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project (number PNURSP2022TR140), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Data Availability Statement: Data will be made available on request.

Acknowledgments: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022TR140), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bibi, K.; Naz, S.; Rehman, A. Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities. *Multimed. Tools Appl.* **2020**, *79*, 289–340. [[CrossRef](#)]
2. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Offline handwritten signature verification—literature review. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November 2017; pp. 1–8.

3. Vargas, J.F.; Ferrer, M.A.; Travieso, C.M.; Alonso, J.B. Offline signature verification based on grey level information using texture features. *Pattern Recognit.* **2011**, *44*, 375–385. [[CrossRef](#)]
4. Justino, E.J.; Bortolozzi, F.; Sabourin, R. Offline signature verification using HMM for random, simple and skilled forgeries. In Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, 3–13 September 2001; pp. 1031–1034.
5. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognit.* **2017**, *70*, 163–176. [[CrossRef](#)]
6. Dey, S.; Dutta, A.; Toledo, J.I.; Ghosh, S.K.; Lladós, J.; Pal, U. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv* **2017**, arXiv:1707.02131.
7. Kalera, M.K.; Srihari, S.; Xu, A. Offline signature verification and identification using distance statistics. *Int. J. Pattern Recognit. Artif. Intell.* **2004**, *18*, 1339–1360. [[CrossRef](#)]
8. Munich, M.E.; Perona, P. Visual identification by signature tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 200–217. [[CrossRef](#)]
9. Dimauro, G.; Impedovo, S.; Pirlo, G.; Salzo, A. A multi-expert signature verification system for bankcheck processing. *Int. J. Pattern Recognit. Artif. Intell.* **1997**, *11*, 827–844. [[CrossRef](#)]
10. Ferrer, M.A.; Alonso, J.B.; Travieso, C.M. Offline geometric parameters for automatic signature verification using fixed-point arithmetic. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 993–997. [[CrossRef](#)] [[PubMed](#)]
11. Kumar, R.; Sharma, J.D.; Chanda, B. Writer-independent offline signature verification using surroundedness feature. *Pattern Recognit. Lett.* **2012**, *33*, 301–308. [[CrossRef](#)]
12. Huang, K.; Yan, H. Offline signature verification based on geometric feature extraction and neural network classification. *Pattern Recognit.* **1997**, *30*, 9–17. [[CrossRef](#)]
13. Ramesh, V.E.; Murty, M.N. Offline signature verification using genetically optimized weighted features. *Pattern Recognit.* **1999**, *32*, 217–233. [[CrossRef](#)]
14. Pal, S.; Alaei, A.; Pal, U.; Blumenstein, M. Performance of an offline signature verification method based on texture features on a large indic-script signature dataset. In Proceedings of the 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece, 11–14 April 2016; pp. 72–77.
15. Chen, S.; Srihari, S. A new offline signature verification method based on graph. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; Volume 2, pp. 869–872.
16. Dutta, A.; Pal, U.; Lladós, J. Compact correlated features for writer independent signature verification. In Proceedings of the 2016 23rd international conference on pattern recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 3422–3427.
17. Bromley, J.; Bentz, J.W.; Bottou, L.; Guyon, I.; LeCun, Y.; Moore, C.; Shah, R. Signature verification using a “siamese” time delay neural network. *Int. J. Pattern Recognit. Artif. Intell.* **1993**, *7*, 669–688. [[CrossRef](#)]
18. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 539–546.
19. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
20. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille Grande Palais, France, 10–11 July 2015; Volume 2.
21. Qi, Y.; Song, Y.Z.; Zhang, H.; Liu, J. Sketch-based image retrieval via siamese convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 2460–2464.
22. Jagtap, A.B.; Sawat, D.D.; Hegadi, R.S.; Hegadi, R.S. Siamese Network for Learning Genuine and Forged Offline Signature Verification. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 131–139.
23. Yapici, M.M.; Tekerek, A.; Topaloglu, N. Convolutional neural network-based offline signature verification application. In Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 3–4 December 2018; pp. 30–34.
24. Sam, S.M.; Kamardin, K.; Sjarif, N.N.A.; Mohamed, N. Offline signature verification using deep learning convolutional neural network (CNN) architectures GoogLeNet inception-v1 and inception-v3. *Procedia Comput. Sci.* **2019**, *161*, 475–483.
25. Jagtap, A.B.; Sawat, D.D.; Hegadi, R.S.; Hegadi, R.S. Verification of genuine and forged offline signatures using Siamese Neural Network (SNN). *Multimed. Tools Appl.* **2020**, *79*, 35109–35123. [[CrossRef](#)]
26. Ferrer, M.A.; Diaz-Cabrera, M.; Morales, A. Synthetic offline signature image generation. In Proceedings of the 2013 International Conference on Biometrics (ICB), Madrid, Spain, 4–7 June 2013; pp. 1–7.
27. Alaei, A.; Pal, S.; Pal, U.; Blumenstein, M. An efficient signature verification method based on an interval symbolic representation and a fuzzy similarity measure. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2360–2372. [[CrossRef](#)]
28. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
29. CUI, G.X.; Li, D.K. Research on Handwritten Digit Recognition Based on Adam Optimizer Self-encoding. *J. Jiamusi Univ. Nat. Sci. Ed.* **2018**, *154*, 3–11.

-
30. Zhang, Z. Improved adam optimizer for deep neural networks. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–2.
 31. Keskar, N.S.; Socher, R. Improving generalization performance by switching from adam to sgd. *arXiv* **2017**, arXiv:1712.07628.