Journal of Information Systems Engineering and Management
2025, 10(26s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Offline Signature Verification USING Siamese Neural Network

Janhavi Kadam[1*], Gauri Phadtare[2], Ashwini Pawar[3], Kamalkishor Maniyar[4]

*[1]Electrical Engineering Department, Dr. D.Y. Patil Institute of Technology, Pimpri, Pune, India. * janhavi.kadam@dypvp.edu.in*
[2]Mechanical Engineering Department, Pimpri Chinchwad College of Engineering, Nigdi, Pimpri-Chinchwad, Maharashtra, India
[3]Electrical Engineering Department, Dr. D.Y. Patil Institute of Technology, Pimpri, Pune, India
[4]Mechanical Engineering Department, Dr. D.Y. Patil Institute of Technology, Pimpri, Pune, India

ARTICLE INFO

Received: 30 Dec 2024

Revised: 20 Feb 2025

Accepted: 02 Mar 2025

ABSTRACT

Verifying offline signatures is challenging in identity verification procedures for legal documents and financial transactions. The subtle but significant distinctions between authentic and fraudulent signatures must be carefully considered because some forgeries may merely alter a portion of the signature. This task is more difficult when the identity of the writer is unknown, which frequently occurs in real-world scenarios. In this paper, we propose a system that verifies offline, writer-independent signatures automatically and accurately using a Siamese neural network. This network consists of two or more identical sub-networks that share parameters and weights. The network calculates Euclidean distances between images by processing both similar and distinct images. When two signatures are the same, the distance is smaller; when they are different, it is larger. One-shot learning, which allows the network to learn from fewer image pairs, is its primary advantage. Two BHsig260 datasets including both authentic and fake signatures in Bengali and Hindi were used to test our technique. The model's proficiency was demonstrated by the Siamese network's strong accuracy of 82% and 84% on the Bengali and Hindi signature datasets, respectively. These findings show how well the model can differentiate between authentic and fake signatures, indicating its potential for real-world use across a range of industries.

Keywords: Deep learning, Siamese networks, convolutional neural networks, and signature verification.

INTRODUCTION

Verification of handwritten signatures is crucial in a variety of contexts, including financial transactions and personal authenticity. However, traditional approaches frequently suffer from little labeled data, distortions, and disparate writing styles. In computer vision or pattern recognition, signature verification becomes an important research topic [1]. Depending on the input format, it might be either online or offline. Because online systems use more information, such as stroke order, writing speed, pressure, etc., they typically perform better than offline systems [3]. However, they are more costly and less useful since they require specialized technology to record the pen-tip trajectory. On the other hand, offline signature verification works better for things like document authentication and cheque transactions. As it has more applications, automatic offline signature verification is the main focus of this effort.

The practice of identifying fraudulent signatures is known as signature verification. Fake signatures that closely mimic a person's name and signature are the hardest to identify. The two primary methods for verifying signatures are Writer-Independent and Writer-Dependent. The system learns only from a single user's signatures and is only able to validate those signatures in writer-dependent verification. The system learns from a dataset of several signatures as in writer-independent verification, and it can validate any signature by using the patterns it discovered in the dataset [4]. The advantage of writer-independent verification over writer-dependent verification is that the former does not need to be updated for every new person, which would be unaffordable for organizations like banks that regularly get new clients. The goal of this research is to develop a CNN-based Siamese network model that can perform writer-independent verification and differentiate between authentic and fraudulent signatures.

LITERATURE REVIEW

An overview of the numerous techniques used by researchers to obtain successful outcomes in offline signature verification is given in this section. Both deep learning and handmade features are used in the methods. Hafemann et al. [2] looked at current developments in this field, assessed the progress made in solving the offline signature verification problem over the previous few decades, and considered possible future research avenues. Additionally, they combined the most recent results from several methods on three datasets: GPDS, MCYT, and CEDAR. Meekkashi et al. [5] obtained high accuracies of 78% and 93% for verification and identification by using gradient, structural, and concavity characteristics for feature extraction at the word level rather than the character level. In order to show that affine arc-length parameterization was better than traditional methods, Mario E. Munich et al. [3] created a vision-based biometric methodology. Other method such as location-based techniques, projection and contour-based methods, linked components, blob structure, tangent path, and the curvature of local features, were also frequently used for signature authentication. Local Randon Transform was employed by Vahid Kiani et al. [6] as a feature extractor. Additionally, a number of novel methods employed pixel matching [7], hybrid features [9, 10] that combined local and global characteristics to create a visual codebook with correlated features for signature verification, and structural features extracted from the contour of a signature using the modified direction feature and its enhanced version [8].

Bhunia et al. [9] trained the model with only authentic signatures and extracted statistical information from signatures using low quantized patterns and discrete wavelets. For a limited number of photos, their approach greatly enhanced performance. In order to learn features from photos, Luiz G. Hafemann et al. [11] trained a deep convolutional neural network on a different set of users. In order to standardize the representation of signatures of different sizes, they also modified the network architecture by utilizing Spatial Pyramid Pooling. Siamese Neural Networks were first presented by Koch et al. [12] for one-shot image recognition, which worked well for verification tasks requiring little training data. To distinguish between image pairs, they employed a twin network design. SigNet, a convolutional Siamese network, was proposed by Dey et al. [13] for writer-independent offline signature verification. For similar pairs, they minimized the Euclidean distance, and for dissimilar ones, they maximized it. They outperformed the present state-of-the-art in their studies on a variety of datasets. Convolutional neural networks (CNNs) were employed by Ramesh Kumar Mohapatra et al. [14] in 2019 to learn features from authentic and fraudulent signatures that had already been pre-processed.

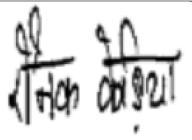
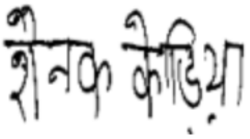
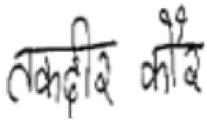
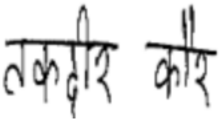
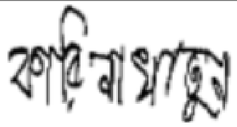
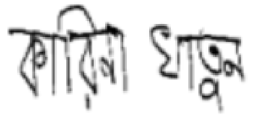
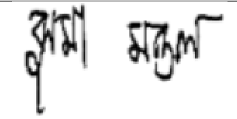
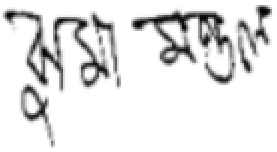
The network was horizontally expanded by the architecture, which employed several filters at the same level. The largest dataset of handwritten signatures, GPDS synthetic signature, was utilized by Jahandad et al. [15] in 2019 to categorize the signatures of 1,000 users, each of whom had 30 fraudulent and 24 legitimate signatures. Additionally, they made advantage of two well-known versions of the GoogLeNet architecture, CNN Inception-v1 and Inception-v3. In 2020, a Siamese neural network incorporating a convolutional neural network as a sub-network was proposed by Amruta B. Jagtap et al. [16]. They applied a contrastive loss function to the Siamese network's embedding vector after adding particular statistical information. For offline signature verification, Neha Sharma et al. [17] employed a Siamese neural network. In addition to performing well on Bengali and Hindi datasets containing expert forgeries, their network attained a 92% accuracy rate on the GPDS synthetic dataset in English. A graph matching technique was employed by Chen et al. [18] to verify off-line signatures. Local similarity is evaluated using multiresolution signature features that are based on gradient, structural, and concavity qualities. Combining deformation and similarity ratings allowed for the ultimate determination of whether a signature was authentic or fraudulent. This approach performs better than earlier offline methods and produces outcomes that are on par with online systems.

PROPOSED METHODOLOGY

3.1. Dataset Description

We employed two well-known benchmark databases—the (1) BHSig260 (Hindi) signature collection and the (2) BHSig260 (Bengali) signature collection—to assess our proposed signature verification approach. Examples of both authentic and fake signatures from each dataset are displayed in Table 1.

Table 1. Sample genuine and forged images of Hindi and Bengali datasets

Dataset name	Genuine signature	Forged signature
Hindi dataset		
		
Bengali dataset		
		

3.1.1. BHSig260 (Hindi)

The BHSig260 (Hindi) signature dataset has 160 Hindi signers. Each signer has 24 real and 30 forged signatures. The real signatures are 3840 (160 * 24) and the forged signatures are 4800 (160 * 30). The Hindi dataset has 8640 images altogether.

3.1.2. BHSig260 (Bengali)

There are 100 signatures in the Bengali signature corpus BHSig260. Every individual has 30 fake signatures and 24 legitimate ones. Thus, there are 5400 signatures in the Bengali dataset, consisting of 2400 authentic and 3000 fraudulent signatures.

3.2. Pairing of Signature Images

Each dataset is divided into training, test, and validation sets so that the system may validate signatures from various authors. For training, testing, and validation, 120, 20, and 20 of the 160 signers in the Hindi dataset are utilized. Since there are 100 signers in the Bengali dataset, 80, 10, and 10 are used for the same. There are 276 genuine and 720 real-fake signature pairs for each user, with each user having 24 real and 30 fraudulent signatures. Each user's real signature is matched with twelve randomly selected false signatures of the same user in order to maintain category balance. Thus, each user has 300 real-fake pairs. Table 2 shows the total number of real-real and real-fake pairs for the Bengali and Hindi datasets.

Table 2. Total no of image pairs in the dataset

Pairs	Hindi Dataset			Bengali Dataset		
	Training	Testing	Validation	Training	Testing	Validation
Genuine-Genuine pairs	33120	5520	5520	22080	2760	2760
Genuine-Forged pairs	36000	6000	6000	24000	3000	3000
Total no. of data points	69120	11520	11520	46080	5760	5760

3.3. Pre-Processing

To enhance the model's fit and quality, the input data must be pre-processed. Normalization and resizing are pre-processing methods used in this study. The sizes of the signature photos range from 153 * 258 to 819 * 1137. Bilinear interpolation was used to reduce all of the signature images to 155 * 220 pixels. The pixel values in each image were normalized by dividing them by 255.

3.4. Siamese Deep Convolutional Neural Network

Two identical subnetworks with the same parameters and weights make up the Siamese neural network design. In addition to pooling layers that minimize and summarize the output, they contain many convolutional layers with varying kernel sizes. For nonlinearity, rectified linear units was used. For writer-independent signature verification, the architecture employed by Sounak Dey [13] is depicted in Figure 1. The input is a 155 x 220 signature image. After channel-wise normalization and pooling with size 3x3 and stride 2, it moves on to the first convolutional layer with 96 kernels of size 11x11 and stride 1. The second convolutional layer contains 256 5x5 filters. The third lacks pooling and normalizing and contains 384 3x3 kernels. The fourth contains 256 3x3 kernels with pooling and dropout. The contrastive loss is then taken into account by two completely connected dense layers, one with 1024 nodes and one with 128 nodes. The embedding is the final layer, which has 128 neurons. The subnetworks are connected by a contrastive loss function at the top that calculates the Euclidean distance between the embeddings. One popular loss function for Siamese networks is the contrastive loss [18]. It is described as

$$L(c, s_1, s_2) = (1 - c)\frac{1}{2}(D_W)^2 + (c)\frac{1}{2}\{max\,(0, m - D_W)\}^2$$

where, c is 1 if two samples are from the same class and 0 otherwise, s1 and s2 are signature samples, m is the margin and DW is the feature space distance given by f(s1 ; w1) – f(s2 ; w2), where f is a function that uses CNN to map a signature image to a vector and w1, w2 are learned weights for a network layer. The training input is a pair (s1, s2) with a binary label as follows.

c=1 ..if (s1,s2) (Genuine, Genuine)

c=0 ..if (s1,s2) (Genuine, Forged)

where s1 and s2 are sample signatures; they are labeled as 0 if they are both genuine, and as 1 if they are not the same. Here, the threshold value is used for comparison. If the sample signature's distance is less than or equal to the threshold value, then the signature is genuine. If the distance value is more than the threshold value, then the signature is a fake signature. This is how we can verify if the signature is real or not.

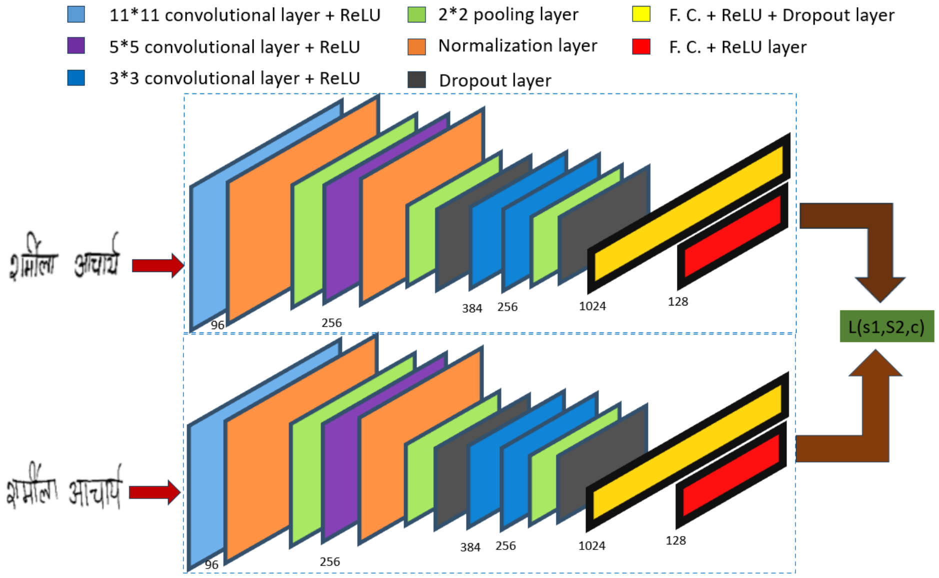

11*11 convolutional layer + ReLU
5*5 convolutional layer + ReLU
3*3 convolutional layer + ReLU
2*2 pooling layer
Normalization layer
Dropout layer
F. C. + ReLU + Dropout layer
F. C. + ReLU layer
96
256
384
256
1024
128
L(s1,S2,c)
96
256
384
256
1024
128

Figure 1. Architecture of Siamese network

Table 3 shows the parameters for the CNN layers. We used Glorot and Bengio [19] to initialize the weights and set the biases to 0. We trained the model with RMSprop and Adam for 20 epochs with 128 mini batches. The initial LR was 1e-4 with m = 0.9 and ϵ = 1e- 8. Table 4 has these hyperparameter values. We used Python, Keras, and TensorFlow to make the model on a Intel® Xeon® Gold 6248R CPU @3.00GHz workstation with 512GB RAM. Training took about 7 hours for different databases.

Table 3. The architecture of Siamese model

Block	Layer	Size	Parameters
Conv block 1	Convolutional	96*11*11	Stride =1
	Normalization		
	Pooling	96*3*3	Stride =2
Conv block 2	Convolutional	256*5*5	Stride =1, padding=2
	Normalization		
	Pooling + Dropout	256*3*3	Stride =2
Conv block 3	Convolutional	384*3*3	Stride =1, padding=1
Conv block 4	Convolutional	256*3*3	Stride =1, padding=1
	Pooling + Dropout	256*3*3	Stride =2
	Fully connected + Dropout	1024	
	Fully connected	128	

Table 4. Value of hyperparameters

Hyperparameter	Value
Initial learning rate	1E-4
Momentum	0.9
Learning rate schedule	LR*0.1
Weight decay	5E-4
Fuzzy factor	1E-8

RESULTS

Using the BHSig260 Bengali dataset, the suggested Siamese network was assessed using three optimizers: RMS, Adam, and SGD. The accuracy of these three optimizers on BHSig260 Bengali dataset is given in Table 5. According to the table, the SGD optimizer had the lowest accuracy and the RMS optimizer the greatest. For the BHSig260 Hindi dataset, the RMS optimizer was also utilized because it provided the best accuracy of 84%.

Table 5. Accuracy for different optimisers

Dataset Name	Optimizer		
	Adam	RMSprop	SGD
BHSig260 Bengali	0.79	0.84	0.51

1. Hindi dataset

The training and validation losses for the Hindi dataset are displayed in Figure 2. The model training was stopped after 20 epochs. The confusion matrix of test data with 82% accuracy is displayed in Figure 3. There are 11520 image pairs in the Hindi test dataset with 5654 authentic labels and 5866 fake labels. In addition to 1046 false positives and 1004 false negatives, the model correctly identified 4862 true positives and 4608 true negatives. The model accurately distinguishes between real and fake signatures, as seen in Figure 4.

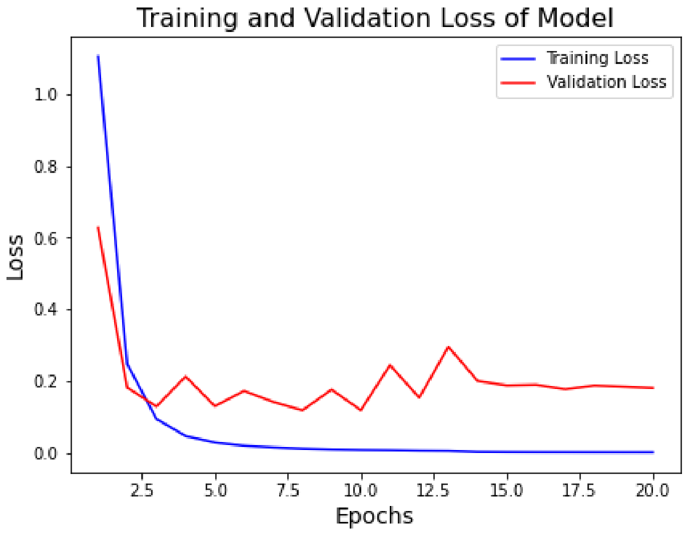


Figure 2. Training and validation loss for BHSig260 Hindi dataset

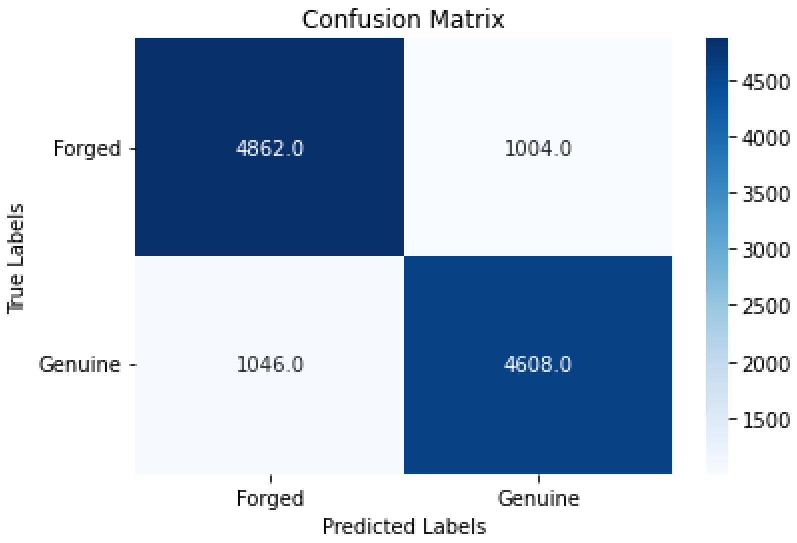
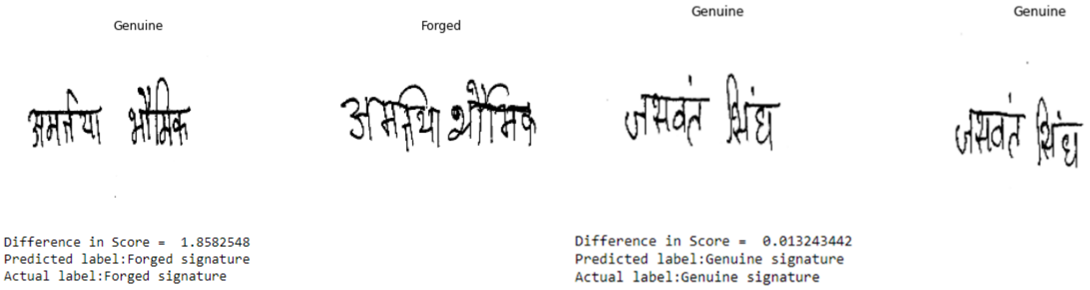


Figure 3. Confusion matrix for BHSig260 Hindi dataset



```
Difference in Score =  1.8582548
Predicted label:Forged signature
Actual label:Forged signature
```

```
Difference in Score =  0.013243442
Predicted label:Genuine signature
Actual label:Genuine signature
```

Figure 4. Model prediction for Hindi dataset

2. Bengali dataset

Using the Bengali dataset, the model trained for 20 epochs. The Bengali dataset has a smaller training and validation loss than the Hindi dataset, as seen in Figure 5. The confusion matrix for the 84% accurate test results is displayed in Figure 6. There are 5760 image pairs in the test set for the Bengali dataset, with 2825 real labels and 2935 fake labels. 346 false positive and 570 false negative pairs were incorrectly identified by the model, while 2365 true positive and 2479 true negative pairs were successfully classified. The model accurately distinguishes between real and fake signatures as shown in Figure 7.

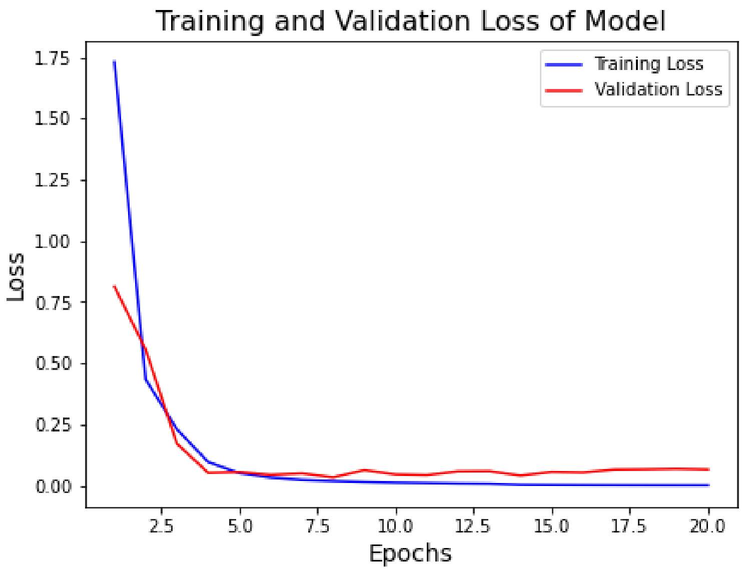


Figure 5. Training and validation loss for BHSig260 Bengali dataset

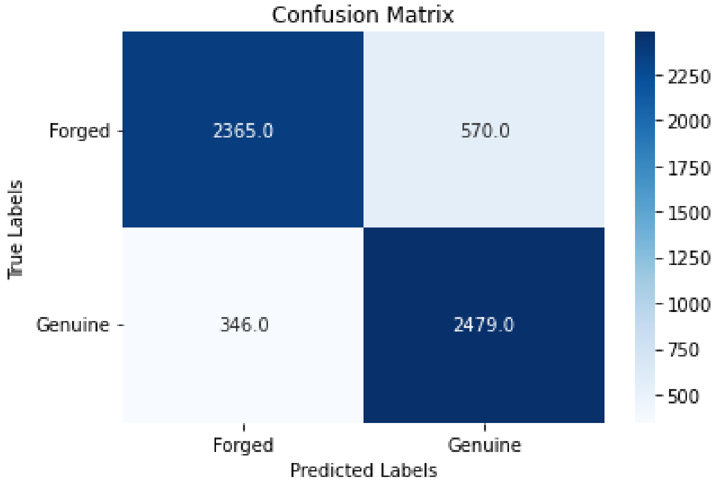
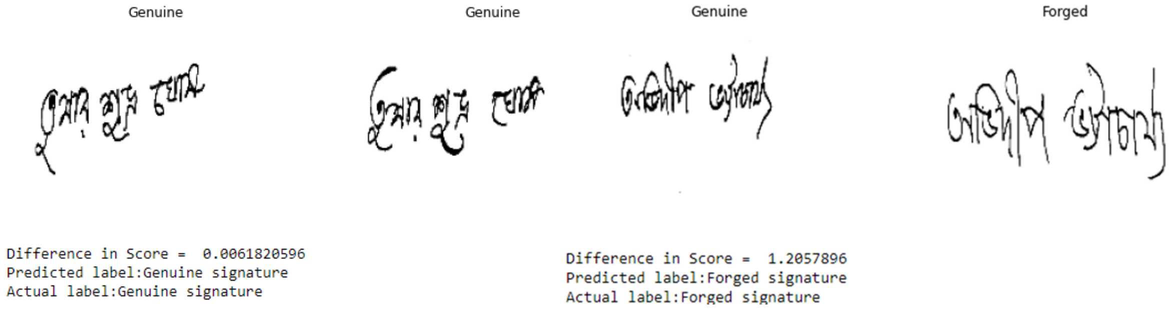


Figure 6. Confusion matrix for BHSig260 Bengali dataset



```
Difference in Score =  0.0061820596
Predicted label:Genuine signature
Actual label:Genuine signature
```

```
Difference in Score =  1.2057896
Predicted label:Forged signature
Actual label:Forged signature
```

Figure 7. Model prediction for Bengali dataset

Table 6 displays the model's performance metrics for both datasets, including accuracy, precision, recall, and F1-score. The created class's accuracy, precision, recall, and F1-score for the Hindi dataset are 82%, 82%, 83%, and 83%, respectively. These metrics are 84%, 87%, 81%, and 84% for the forged class, respectively, for the Bengali dataset. Table 6 also provides the performance parameters for the genuine class. The table demonstrates that the Bengali dataset achieves the maximum accuracy across all performance factors.

Table 6. Performance parameters for Hindi and Bengali dataset

		Precision	Recall	F1 - score	Accuracy
BHSig260 Hindi	Forged	0.82	0.83	0.83	0.82
	Genuine	0.82	0.81	0.82	
BHSig260 Bengali	Forged	0.87	0.81	0.84	0.84
	Genuine	0.81	0.88	0.84	

CONCLUSION

A method for offline signature verification utilizing the Siamese network—which learns writer-independent properties from data rather than hand-crafted ones—is presented in this research. In order to accept pairs of images as input—either genuine-genuine or genuine-forged—the technique employs two identical CNNs with the same parameters. To determine if the signatures are authentic or fake, a contrastive loss based on Euclidean distance is applied. Using several optimizers, the approach is evaluated on the Hindi and Bengali datasets. According to the findings, the technique can effectively identify forgeries in both datasets, with the Bengali dataset exhibiting a marginally higher accuracy.

REFRENCES

[1] Bibi K, Naz S, Rehman A, Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities. Multimedia Tools and Application 2020; 79: 289–340.

[2] Hafemann LG, Sabourin R, Oliveira LS, Offline handwritten signature verification—literature review. In: Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA); 28 November 2017; Montreal, QC, Canada. pp. 1–8.

[3] Munich ME, Perona P, Visual identification by signature tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence 2003; 25: 200–217.

[4] Hafemann LG, Sabourin R, Oliveira LS, Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognition 2017; Vol. 70: 163–176.

[5] Kalera MK, Srihari S, Xu A, Offline signature verification and identification using distance statistics. International Journal of Pattern Recognition and Artificial Intelligence 2004; 18(7): 1339-1360.

[6] Kiani V, Pourreza R, Pourreza HR, Offline signature verification using local radon transform and support vector machines. International Journal of Image Processing 2010; 3: 184-194.

[7] Bhattacharya I, Ghosh P, Biswas S, Offline signature verification using pixel matching technique. Procedia Technology 2013; 10: 970–977.

[8] Nguyen V, Blumenstein M, Muthukkumarasamy V, Leedham G, Off-line signature verification

[9] using enhanced modified direction features in conjunction with neural classifiers and support vector machines. In: Ninth international conference on document analysis and recognition (ICDAR 2007), IEEE, 2007; pp. 734–738.

[10] Bhunia AK, Alaei A, Roy PP, Signature verification approach using fusion of hybrid texture Features. Neural Computing Application 2019; 31: 8737-8748.

[11] Dutta A, Pal U, Llad´os J, Compact correlated features for writer independent signature verification, In: 23rd international conference on pattern recognition (ICPR 2019), IEEE; 2019; pp. 3422–3427.

[12] Hafemann LG, Sabourin R, Oliveira LS, Writer-independent feature learning for offline signature verification using deep convolutional neural networks. In: International joint conference on neural networks (IJCNN 2016), IEEE; 2016; pp. 2576–2583.

[13] Koch G, Zemel R, Salakhutdinov R, Siamese neural networks for oneshot image recognition, In: Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), 2015; Lille, France. pp.1-8

[14] Dey S, Dutta A, Toledo JI, Ghosh SK, Lladós J, Pal U, Signet: Convolutional siamese network for writer independent offline signature verification. Pattern Recognition Letters 2017.

[15] Mohapatra R, Shaswat K, Kedia S, Offline Handwritten Signature Verification using CNN inspired by Inception V1 Architecture. In: 5th International Conference on Image Information Processing; 2019.

[16] Jahandad, Sam SM, Kamardin K, Sjarif NNA, Mohamed N, Offline Signature Verification using Deep Learning Convolutional Neural Network (CNN) Architectures GoogLeNet Inception-v1 and Inception-v3. Procedia Computer Science 2019; 161: 475–483.

[17] Jagtap AB, Sawat DD, Hegadi RS, Hegadi RS, Verification of genuine and forged offline signatures using Siamese Neural Network (SNN), Multimedia Tools and Application 2020; 79: 35109–35123.

[18] Sharma N, Gupta S, Mohamed HG, Anand D, Mazón JLV, Gupta D, Goyal N, Siamese Convolutional Neural Network-Based Twin Structure Model for Independent Offline Signature Verification. Sustainability 2022; 14: 11484.

[19] 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); 20–24 August 2006; Hong Kong, China. pp. 869–872.