

# Signature Recognition and Forgery Detection Using Deep Learning.

Hanat Raji-Lawal<sup>1</sup>, Kehinde Sotonwa<sup>1</sup>, Gbadebo Adegbuyi David<sup>2</sup>, Aiyeniko Olukayode<sup>1</sup>, Zubair Adam<sup>1</sup>,  
Oloyede Ayodele<sup>1</sup>, Umoren Grace<sup>1</sup>

<sup>1</sup>Computer Science Department, Faculty of Science, Lagos State University, Ojo, Lagos, Nigeria

<sup>2</sup>Department of Computer Science Education, College of Information and Technology Education, Lagos State  
University of Education, Oto/Ijanikin, Lagos, Nigeria.  
halaw313@yahoo.com<sup>1</sup>

## ABSTRACT

*Signature verification is being applied broadly in industries. Characteristics like psychological and behavioural can be used to verify and authenticate an individual, these are called biometrics. Aim: This research focus is on offline signature recognition and forgery detection using Convolutional Neural Network (CNN) and Siamese Neural Network. Method: This study made use of the CEDAR (Center of Excellence for Document Analysis and Recognition) data set, which has a combined total of 1,320 authentic signatures and 1,320 forgeries. It maintains a balance between false rejection rate (FRR) and false acceptance rate (FAR). The model was trained and evaluated on a comprehensive dataset, resulting in an FRR of 0.0161 and a FAR of 0.2903. A comparison of the model's performance revealed that this approach performed favourably in terms of FRR and FAR. These results suggest that this study's approach is a promising solution for signature recognition and forgery detection applications.*

**Keywords:** Signature, Recognition, Biometrics, Forgery, Deep Learning

## 1. Introduction

A signature is a written representation of an individual. It is also a legal representation of an individual. Signatures are used to sign cheques, contracts, and other legally binding documents. Signatures are widely used and of great importance, due to these, attempts are sometimes made to forge one. Forgery refers to the false making or material alteration of a legal instrument; in this case, a signature is a legal instrument. To avoid forgery and keep confidentiality there is a need for an authentication system. Signature Verification is a system that detects the validity and authenticity of a signature thus detecting forgery. Signature verification and authentication can be done in two different methods: Online signature verification (Yanikoglu & Kholmatov, 2009) and Offline signature verification (R. Plamondon et al., 1989). Offline signature verification systems can be done using two approaches (D. Bertolinia et al., 2010) writer dependent signature verification and writer-independent signature verification. In a writer-dependent system, models are constructed for each writer and need to be retrained with every new writer or signer and thus would incur a huge cost (Avola, D. et al., 2021). For a Writer-independent system, a general model is built from a few samples taken from the user and the model is trained for a specific set of users that can be mapped to all the writers (Parcham, E., et al., 2021). The writer-independent approach is designed to handle skilled signature forgeries (Rantzsch Hannes, et al., 2016).

Signature verification is being applied in industries like the banking, finance, and insurance industries. Much research is being carried out on improving the signature verification system and developing better algorithms for such. Several psychological and behavioural characteristics can be used to verify and authenticate an individual. These characteristics are called biometrics. Examples of characteristics used in biometrics include voice, lips movement, fingerprint and signature (J. P. Campbell, 1997, A.W.C.; Leung, et al., 2000, A. K. Jain et al., 1997). A signature is another behavioural biometric that is mostly like handwriting and is very useful for individual authentication (Shanker & Rajagopalan, 2007). The signature is an acceptable proof of identity of the person in a transaction taken on his or her behalf. Thus, the users are more likely to approve this kind of computerised authentication method (R. Plamondon, et al., 1995). Signature verification software is software that compares signatures and checks for authenticity, they prevent human error during the signature process and lower the chances of fraud in the authentication process. Signatures have proven to be an important biometric and useful authentication tool. The accurate recognition of genuine signatures and detection of forgeries is crucial in various applications, such as banking and financial services, legal documents, contracts, and access control systems. Given the importance of signatures in several sectors, the need to secure this representation of individuals arises This research focuses on offline signature verification, using CNN and Siamese Neural Network.

## 2. Related Works

Signature recognition has experienced significant advancements in recent years due to the adoption of a wide variety of techniques and features. Online methods have utilized multi-segment vector quantization and moment-based approaches, while offline methods have employed deep learning, geometrical parameters, and morphological features. Machine learning techniques such as CNN, ResNet, and SVM have been particularly prominent in offline signature recognition, resulting in higher accuracy rates, with some studies reporting accuracy as high as 99.7-99.9%. The choice of features and methods greatly influences the performance of signature recognition systems. Combining different features and classifiers can lead to improved recognition and forgery detection rates. Despite the progress made in previous studies, there is still room for further improvement and exploration of novel methods and features. The field of signature recognition continues to evolve as researchers develop and apply new machine-learning techniques, pushing the boundaries of accuracy and error rates in the pursuit of more secure and reliable systems. Tables 1 and 2 entails online and offline approaches to signature recognition, where FRR and FAR denotes False Rejection Rate and False Acceptance Rate respectively.

Table 1: A Summary of Online Approach to Signature Recognition

Author(s)	Method/Technique	Features	FRR (%)	FAR (%)
Razzak, Alhaqbani (2014)	Multi-segment vector quantization for online signature identification	Dividing the signature into several parts, each with its own codebook	N/A	N/A

Author(s)	Method/Technique	Features	FRR (%)	FAR (%)
Mohamed et al., (2004)	baseline extraction algorithm for online signature recognition based on vector rules.	Direction, slant, baseline, pressure, speed, and the number of pen ups and downs.	N/A	N/A
Keit, Palanjppan, (2001)	A pen pressure-based technique.	Customized pen detecting writing pressure, and pressure signal for identification.	2.13	3.14
Nalwa, (1997)	A moment and torque base technique.	Parameterization of online curves over normalized arc length, moments, torque, and inertia in a moving coordinate frame.	7	1

Table 2: A Summary of Offline Approach to Signature Recognition.

Authors	Method	Features	Evaluation Metrics
Poddar et al. (2020)	CNN, Crest-Through Method, SURF, and Harris corner detection for signature identification and forgery detection.	Noise reduction, scaling, centralization, rotation, length-to-space ratio, width-to-space ratio, Crest-Through parameters.	94 (Recognition), 85-89 (Forgery Detection) Accuracy
Kshitij et al. (2021)	CNN and ResNet for forged and genuine signature classification.	Extracting behavioural characteristics like delays and hesitation in signatures.	Not specified
Eman et al. (2020)	Deep-learning method using CNN for signature verification and forgery detection, template matching, and Hidden-Markov model.	Distinctive signature attributes.	99.7-99.9 % Accuracy

Kiran et al. (2021)	Analysing techniques for determining the integrity of visual media, both online and offline methods.	Dynamic knowledge of the script, and comparison with reference signatures.	Not specified
Sabourin et al. (1997)	Granulometric size distributions, closest neighbour and threshold-based classifier.	Local form descriptors.	Total error rates: 0.02% - 1.0%
Abbas (2003)	Offline signature recognition using backpropagation neural network prototype, feed-forward neural networks, and training algorithms.	Angle distribution clustered with fuzzy c-means algorithm.	FAR: 10-40%, FRR: 5-16%
Zhang (2006)	Kernel Principal Component Self-regression (KPCSR) model.	Subject-specific KPCSR models, visual data coding.	FRR: 92%, FAR: 0.5%
Armand et al. (2004)	Modified Direction Feature (MDF) combined with neural network classifiers (RBF and Resilient Backpropagation).	Radial Basis Function neural network, Resilient Back Propagation neural network.	Verification: 91.12%
Baltzakis & Papamarkos (2001)	Two-stage perceptron one-class-one-network (OCON) classification structure.	Global features, grid features (pixel densities), and texture features (co-occurrence matrices).	FRR: 3%, FAR: 9.8%
Justino et al. (2002)	Discrete observation Hidden Markov model (HMM).	Pixel density feature, extended-shadow-code feature, axial slant feature.	FAR: 1.44%, 2.50%, 22.67%
Kaewkongka et al. (1999)	Hough transform with a back-propagation-trained neural network.	Parameterized Hough space.	Recognition: 95.24%
Fang et al. (2002)	Crossing method and fractal dimension method with minimum distance classifier.	Smoothness feature, global form features.	AER: 17.3%
Ferrer et al. (2005)	Offline geometric parameters using fixed-point arithmetic with HMM, SVM, and Euclidean distance	Signature envelop, inner stroke distribution in polar and Cartesian	FAR: 3.13%, FRR: 2.12%

	classifiers.	coordinates.	
Audet et al. (2006)	Support Vector Machine (SVM).	Grid, directional, and global aspects of signatures.	Classification: 0.95
Majhi et al. (2006)	Morphological parameter-based signature recognition.	Centre of mass of signature segments.	FRR: 14.58%, FAR: 2.08%
Pinge & Kekre (2005)	Template-matching with neural network classifiers.	40 separate predetermined form templates for feature extraction	FAR: 20% (Alg 1), FAR: 0% (Alg 2)
Samuel D, Samuel I (2010)	Offline method (Hidden Markov Model).	Image cell size, image centre angle, pixels normalized angle.	FAR: 1%, FRR: 0.50%
Abdala, Yousif (2009)	Back-propagation neural networks.	Not specified.	Recognition: 99.31%
Agrawal P et al. (2020)	Deep learning, Optical Character Recognition (OCR), CNN, Scale Invariant Feature Transform (SIFT), SVM.	OCR for typographic characters, CNN for handwritten digits and signatures, and SIFT for feature generation.	Not specified
Kiani et al. (2009)	Local Radon Transform with SVM classifier.	Radon Transform applied locally for line segments and feature extraction.	FRR: 4%, FAR: 17%

### 3. Methodology

Convolutional neural networks (CNNs) are used in this research to extract characteristics including area, centroid, eccentricity, kurtosis, and skewness, while Siamese neural networks are used to classify the data. This study makes use of the CEDAR (Center of Excellence for Document Analysis and Recognition) Signature database, which has a combined total of 1,320 authentic signatures and 1,320 forgeries.

A dataset of offline signatures called CEDAR Signature was used to verify signatures. The data set is known to have a certain level of intra-class variation. The intra-class variation can be attributed to various factors such as differences in writing instruments, writing surfaces, and natural variability in an individual's handwriting due to mood, physical condition, or other factors. 1,320 real signatures were obtained through the contribution of 24 signatures from each of the 55 signatories. Eight times for each subject, three other

writers' signatures were forged, resulting in 1,320 forgeries. Each signature was binarized using a grey-scale histogram after being scanned at 300 dpi in grayscale. Image pre-processing included the stages of salt pepper noise reduction and slant normalization. Each writer's 24 genuine and 24 forgeries are contained in the database. In the pre-processing stage of the signature recognition and forgery detection system, several key techniques were employed to ensure optimal performance. First, the images were resized to a consistent dimension, which not only reduced the computational complexity but also maintained the aspect ratio of the original signatures. Next, the images were converted to grayscale, simplifying the input data while still preserving essential information about the signatures. Following this, normalization was applied to scale the pixel intensities to a standard range, typically between 0 and 1, which facilitated better convergence during the training process. Finally, Otsu's thresholding method was used to convert the grayscale images into binary form. By determining an optimal threshold value, this technique allowed for efficient segmentation of the signature's foreground from its background, effectively highlighting the key features necessary for accurate recognition and forgery detection.

Feature extraction plays a crucial role in the performance of deep learning models. CNN was used to extract relevant features from the signatures, such as Area, Eccentricity, Centroid, Kurtosis, and Skewness. The CNN model was designed with multiple layers, including convolutional, max-pooling, and dense layers, to learn the hierarchical representation of the signatures. Among Deep Learning networks, convolutional neural networks (CNN) are among the most widely utilized and well-liked models (Yao et al., 2022). Many domains, such as computer vision (Fang et al., 2020), voice processing (Palaz et al., 2019), and face recognition (Li et al., 2020), etc., have made use of CNNs. The three primary layer types in convolutional neural networks as seen in figure 1 are the convolutional layer, the pooling layer, and the fully connected (FC) layer.

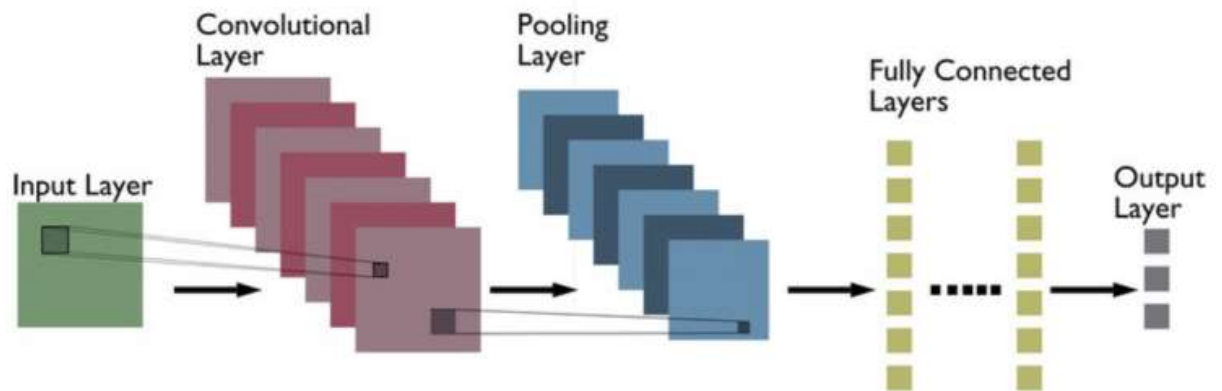


Figure 1: Typical CNN Architecture

A Siamese Neural Network is a deep learning architecture that is designed to handle tasks involving similarity comparisons or measuring the relationship between two input samples. Siamese Neural Networks consist of two identical neural network branches that share the same weights and parameters. It is comprised of two parallel branches, each containing a series of layers that process input data, as represented in figure 2. These branches share the same weights and are trained simultaneously. The two input samples, often referred to as the "anchor" and "positive/negative" samples, are fed into the respective branches. The

network processes both inputs independently and generates a feature vector or embedding for each input. These embedding's represent the high-level features of the inputs and are used to compute the similarity between the two samples. In this study, the network learns to extract discriminative features from signature images and generates embedding's that can accurately measure the similarity between signatures. This allows the system to verify if two signatures belong to the same person or not with a high degree of confidence.

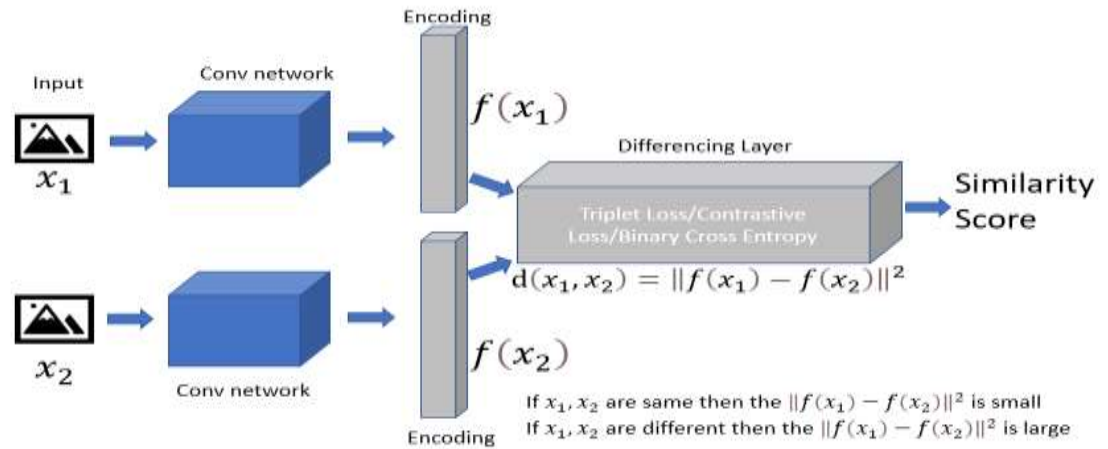


Figure 2: Typical Siamese Network Architecture

Source: <https://medium.com/swlh/one-shot-learning-with-siamese-network-1c7404c35fda>

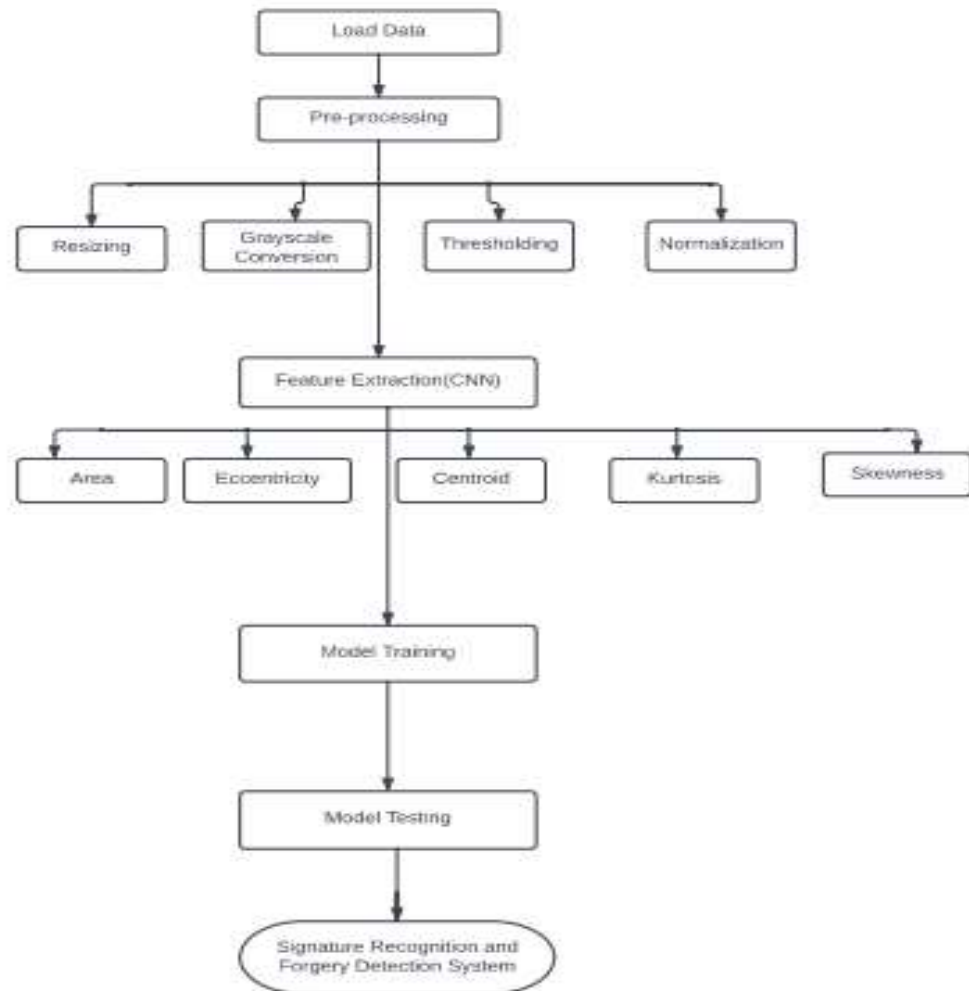


Figure 3: Signature Forgery Detection Model

The process begins with pre-processing as depicted in figure 3, where the input signature image is cleaned up to remove any noise that could interfere with the recognition process. This step involves resizing, converting to grayscale, thresholding and normalisation of the image. Next, the pre-processed signature image is analysed to extract relevant features that can be used to distinguish it from other signatures. The extracted features are then combined to create a representation of the signature that can be used as input to a machine-learning algorithm. The signature representation is passed through a machine learning algorithm that has been trained on a dataset of known signatures. The algorithm uses the input representation to

classify the signature as genuine or forged. This classification step involves comparing the input signature to a set of known genuine signatures or analysing patterns and characteristics that are unique to forged signatures. Finally, the classification result is outputted to the user, along with the similarity score that indicates the likelihood of the classification result. This output can be used to verify the authenticity of a signature.

## 4. Result and Discussion

### 4.1 False Rejection Rate (FRR) and False Acceptance Rate (FAR)

To evaluate the model's performance two additional metrics were calculated: The False Rejection Rate (FRR) and the False Acceptance Rate (FAR). The FRR measures the proportion of genuine signatures that were incorrectly classified as forgeries, while the FAR measures the proportion of forgeries that were incorrectly classified as genuine signatures. The model achieved an FRR of 1.61% and a FAR of 29.03%. A low FRR indicates that the model is effective at recognizing genuine signatures. However, the relatively high FAR suggests that the model is prone to classifying forgeries as genuine signatures. This could be due to the inherent challenges in distinguishing between skilful forgeries and genuine signatures or the limitations of the model's architecture and feature extraction capabilities.

### 4.2 Training and validation performance

The model was trained for 50 epochs. The training process involved minimizing the loss function and maximizing the accuracy of the train set. At epoch 50, the model achieved a training loss of 0.1548 and a training accuracy of 97.92% as depicted on figures 4 and 5 respectively. The relatively low training loss and high training accuracy indicate that the model has learned to recognize and differentiate between genuine and forged signatures effectively. During the training process, the model's performance on the validation set was also monitored. At epoch 50, the validation loss and validation accuracy were 1.4240 and 87.69%, respectively.

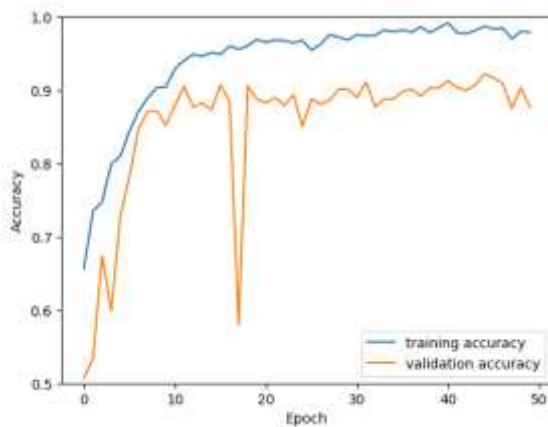


Figure 4: Accuracy Chart

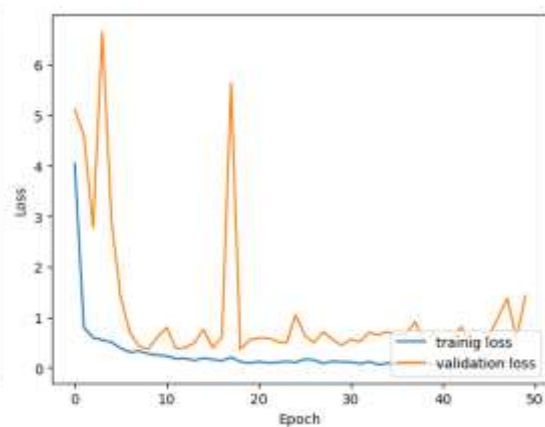


Figure 5: Loss Chart

### 4.3 Test performance

After training, the model's performance on the test set was evaluated. The test loss was found to be 2.1846, and the test accuracy was 83.90%. The test accuracy provides a measure of how well the model can generalize to unseen data. While an accuracy of 83.90% is reasonable.

### 4.4 Comparison with Other Techniques

This section presents a comparison of the model's performance with other models that used CNN, SNN, and other deep learning approaches for signature recognition and forgery detection as depicted in table 3. A study by Hafemann et al. (2017) used a CNN for feature extraction and an SVM classifier for verification, achieving an FRR of 0.029 and a FAR of 0.221. In another study by Zois et al. (2018), a deep Siamese neural network was proposed, resulting in an FRR of 0.023 and a FAR of 0.277. Sarkhel et al. (2017) focused on Bengali handwriting identification using a CNN for feature extraction and an SVM classifier for verification. Yilmaz and Yanikoglu (2018) combined CNNs and other classifiers using score-level fusion, obtaining an FRR of 0.012 and a FAR of 0.295. Lastly, Soleimani et al. (2017) proposed a deep multi-task metric learning approach using a Siamese network for signature verification with an FRR and FAR of 0.0200 and 0.2670 respectively. The implemented model's performance in terms of FRR and FAR compares favourably with other models that utilized CNN, SNN, and other deep learning approaches for signature recognition and forgery detection. While some models show slightly better performance in one metric, the implemented model generally maintains a good balance between FRR and FAR.

Table 3: Summary of Comparison with other techniques

Model	Method	FRR	FAR
Implemented Model	CNN (feature extraction) + Siamese Network	0.00161	0.2903
Hafemann et al. (2017)	CNN (feature extraction) + SVM Classifier	0.0290	0.2210
Zois et al. (2018)	Deep Siamese Neural Network	0.0230	0.2770
Sarkhel et al. (2017)	CNN (feature extraction) + SVM Classifier	0.0180	0.3120
Yilmaz & Yanikoglu (2018)	CNN + Score-level Fusion of Classifiers	0.0120	0.2950
Soleimani et al. (2017)	Deep Multitask Metric Learning + Siamese Network	0.0200	0.2670

#### 4.5 Implementation of Signature Recognition and Forgery Detection Application

In implementing the signature recognition and forgery detection application, the tools Jupyter Notebook, Python, Tensor Flow, Keras, and OpenCV the following tools were used for the implementation of the system.

##### Graphical User Interface GUI



Figure 6: Signature Recognition and Forgery Detection GUI

The Signature Recognition and Forgery Detection application in figure 6 is a user-friendly desktop interface designed for verifying the authenticity of handwritten signatures. Built using Python and Tkinter, it employs a Convolutional Neural Network (CNN) for feature extraction and a Siamese neural network. The pre-trained Siamese network compares and predicts the similarity between a pair of signatures based on the features extracted by the CNN. The interface comprises several key elements to facilitate signature verification, including image selection, result display, and an explanation of the similarity and confidence scores. Upon launching the application, users are presented with a full-screen window that is divided into several sections. The top section contains two buttons, "Select Signature 1" and "Select Signature 2," which enable users to choose the images of the two signatures they wish to compare. Once selected, the images are displayed below the respective buttons, along with the file names. The "Verify" button, located below the signature images, triggers the verification process. When clicked, the application pre-processes the images, feeds them into the pre-trained Siamese neural network model, and calculates the similarity and confidence scores. These scores are then displayed in the results section, which is positioned centrally between the "Verify" button and the explanatory message.

The results section contains three labels, "Genuine signature" or "Forged signature," "Similarity," and "Confidence." A similarity scores closer to 1 indicates high similarity between the signatures, while a score closer to 0 indicates low similarity, as depicted in figures 7, 8 and 9. The confidence score represents the level of certainty of the prediction, with a higher score denoting a more reliable prediction. Overall, the application offers an intuitive and user-friendly interface for verifying the authenticity of signatures. By leveraging machine learning algorithms and an easy-to-understand layout, it simplifies the signature verification process and enhances the user experience.



Figure 7: Comparing Two Genuine Signatures



Figure 8: Comparing Two Forged Signatures



Figure 9: Comparing a Genuine and Forged Signature

## 5. Conclusion

The evaluation metrics used for signature verification model in the signature recognition and forgery detection application provides a comprehensive understanding of its capabilities and limitations. The model performs reasonably well but there is still room for improvement. Future work could explore alternative architectures, feature extraction methods, or training strategies to enhance the model's performance in signature verification tasks. Additionally, incorporating a larger and more diverse dataset could potentially improve the model's generalization capabilities and reduce the FAR.

## REFERENCES

- Abbas, R. (2003). Back propagation Neural Network Prototype for off-line signature verification. Thesis Submitted to RMIT.
- Armand, S., Blumenstein, M., & Muthukkumarasamy, V. (2004). Off-line Signature Verification based on the Modified Direction Feature. *Engineering Applications of Artificial Intelligence*, 14, 0952-1976/04/\$ - PII: S 0 9 5 2.
- Audet, S., Bansal, P., & Baskaran, S. (2006). Off-line signature verification using virtual support vector machines. *ECSE 526 - Artificial Intelligence*, April 7.
- Avola, D., Bigdello, M. J., Cinque, L., Fagioli, A., & Marini, M. R. (2021). R-SigNet: Reduced space writer-independent feature learning for offline writer-dependent signature verification. *Pattern Recognition Letters*, 150, 189-196.
- Baltzakis, H., & Papamarkos, N. (2001). A new signature verification technique based on a two-stage neural network classifier. *Engineering Applications of Artificial Intelligence*, 14, 95-103.
- Bertolini, D., Oliveira, L. S., Sabourin, R., & Justino, E. (2010). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition Letters*, 31(11), 1376-1385.
- Bertolini, D., Oliveira, L., Justino, E., & Sabourin, R. (2010). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition*, 43(1), 387-396.
- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "Siamese" time delay neural network. *Advances in Neural Information Processing Systems*, 6, 737-744.
- Campbell, J. P. (1997). Speaker recognition: A tutorial. *Proceedings of the IEEE*, 85, 1437-1462.

- Das, R. (2014). *Biometric technology: Authentication, biocryptography, and cloud-based architecture* (1st ed.). CRC Press.
- Drouhard, J. P., Sabourin, R., & Godbout, M. (1996). A neural network approach to off-line signature verification using directional PDF. *Pattern Recognition*, 29(3), 415-424.
- Edson, J., Justino, R., Bortolozzi, F., & Sabourin, R. (2002). The Interpersonal and Intrapersonal Variability Influences on Off-line Signature Verification Using HMM. *Proc. XV Brazilian Symp. Computer Graphics and Image Processing*, 197-202.
- Fang, B., Leung, C.H., Tang, Y.Y., Kwok, P.C.K., Tse, K.W., & Wong, Y.K. (2002). Offline signature verification with generated training samples. *IEEE Proc, Image Signal Process*, 149(2), 85-90.
- Fang, W., Love, P. E., Luo, H., & Ding, L. (2020). Computer vision for behaviour-based safety in construction: a review and future directions. *Advanced Engineering Informatics*, 43, 100980.
- Ferrer, M.A., Alonso, J.B., & Travieso, C.M. (2005). Offline Geometric Parameters for Automatic Signature Verification Using Fixed-Point Arithmetic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6).
- Fierrez-Aguilar, J., Nanni, L., Lopez-Peñalba, J., Ortega-Garcia, J., & Maltoni, D. (2005). An on-line signature verification system based on fusion of local and global information. *International Conference on Audio- and Video-Based Biometric Person Authentication*, 523-532.
- Galbally, J., Fierrez, J., & Ortega-Garcia, J. (2010). Performance evaluation of on-line signature recognition. *Pattern Recognition and Image Analysis*, 20(3), 378-384.
- Gupta, R. P., & Joyce, C. A. (1972). Signature identification by computer. *Pattern Recognition*, 4(4), 369-377.
- Hafemann, L. G., Sabourin, R., & Oliveira, L. S. (2017). Learning features for offline handwritten signature verification using deep convolutional neural networks. *Pattern Recognition*, 70, 163-176.
- Hanmandlua, M., Hafizuddin, M., Yusofb, M., & Madasu, V.K. (2005). Off-line signature verification and forgery detection using fuzzy modelling. *Pattern Recognition*, 38, 341-356.
- Herbst, N. M. (1978). Signature classification and verification: A problem in pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(5), 341-346.
- Jain, A. K., Hong, L., Pankanti, S., & Bolle, R. (1997). An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 85(9), 1365-1388.
- Justino, E. J., Oliveira, L., Justino, R., & Sabourin, R. (2002). The interpersonal and intrapersonal variability influences of offline signature verification using HMM. *Proceedings of the XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2002)*.
- Kaewkongka, T., Chamnongthai, K., & Thipakom, B. (1999). Off-Line Signature Recognition using parameterized Hough Transform. *Proceedings of Fifth International Symposium on Signal Processing and its Applications, ISSPA '99, Brisbane, Australia, 22-25 August*.
- Kshitij Swapnil Jain, Udit Amit Patel, Rushab Kheni "HANDWRITTEN SIGNATURES FORGERY DETECTION" (2021). *International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 01 | Jan 2021*.
- Kumar, A., & Ravikanth, C. H. (2007). Personal Authentication Using Hand Vein Triangulation and Knuckle Shape. *IEEE Transactions on Image Processing*, 16(9), 2138-2146.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- LeCun, Y., et al. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pp. 396-404.
- Leung, A. W. C., Liew, S. H., & Lau, W. H. (2000). Lip contour extraction using a deformable model. *Proceedings of the 2000 International Conference on Image Processing*, 2, 255-258.

- Li, H. C., Deng, Z. Y., & Chiang, H. H. (2020). Lightweight and resource-constrained learning network for face recognition with performance optimization. *Sensors*, 20(21), 6114.
- Lu, X., Huang, L., & Yin, F. (2021). Cut and compare: End-to-end offline signature verification network. 2020 25th International Conference on Pattern Recognition (ICPR), 3589-3596.
- Majhi, B., Reddy, Y., & Babu, D. (2006). Novel Features for Off-line Signature Verification. *International Journal of Computers, Communications & Control*, 1(1), 17-24.
- Nalwa, V. S. (1997). Automatic on-line signature verification. *Proceedings of the IEEE*, 85(2), 215-239.
- Oliveira, H. P., Justino, E., & Freitas, C. (2007). The Graphology Applied to Signature Verification. 2007 International Conference on Image Processing, San Antonio, Texas, USA, 277-280.
- Osborne, A. S. (1951). *Questioned documents*. Boyd Printing Company.
- Palaz, D., Magimai-Doss, M., & Collobert, R. (2019). End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. *Speech Communication*, 108, 15-32.
- Parcham, E., Ilbeygi, M., & Amini, M. (2021). CBCapsNet: A novel writer-independent offline signature verification model using a CNN-based architecture and capsule neural networks. *Expert Systems with Applications*, 185, 115649.
- Pinge, S., & Kekre, H. (2005). Signature Identification using Neural Networks. *Proceedings of National Conference on Image Processing 2005 (NCIP2005)*, Organized by TSEC, Mumbai, 31-39.
- Plamondon, R. (1995). The handwritten signature as a biometric identifier: Psychophysical model and system design. *Proceedings of the European Convention on Security and Detection*, 23-27.
- Plamondon, R., & Lorette, G. (1989). Automatic signature verification and writer identification - The state of the art. *Pattern Recognition*, 22(2), 107-131.
- Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 63-84.
- Poddar, Jivesh & Parikh, Vinanti & Bharti, Drsantosh. (2020). Offline Signature Recognition and Forgery Detection using Deep Learning. *Procedia Computer Science*. 170. 610-617. 10.1016/j.procs.2020.03.133.
- Rantzsch, H., Yang, H., & Meinel, C. (2016). Signature embedding: Writer-independent offline signature verification with deep metric learning. *Lecture Notes in Computer Science*, 10073, 616-625.
- Sabourin, R., Genest, G., & Preteux, F.J. (1997). Off-line signature verification local granulometric size distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9), 976-988.
- Samuel, D., & Samuel, I. (2010). Novel Feature Extraction Technique for Offline Signature Verification System. *International Journal of Computer Science and Information Security*, 7(2), 234-240.
- Sarkhel, R., Das, N., & Kundu, M. (2017). A deep learning-based approach for Bengali handwriting identification. In 2017 3rd International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) (pp. 245-250). IEEE.
- Sato, T., Nishida, M., & Oka, R. (1994). On-line signature verification based on shape, motion, and writing pressure. *International Conference on Pattern Recognition*, 2, 537-539.
- Soleimani, A., Fouladi, K., & Araabi, B. N. (2017). Deep Multitask Metric Learning for offline signature verification. In 2017 25th Iranian Conference on Electrical Engineering (ICEE) (pp. 1388-1393). IEEE.
- Sultana, S., & Chowdhury, M. (2010). An Efficient Algorithm for Offline Signature Verification. 2010 International Conference on Electrical & Computer Engineering (ICECE), Dhaka, Bangladesh, 237-240.
- Vargas, J. F., Travieso, C. M., Alonso, J. B., & Ferrer, M. A. (2011). Off-line signature verification based on grey level information using texture features. *Pattern Recognition*, 44(2), 375-385.

- Yang, X., Yu, S., Liang, S., & Xie, C. (2017). Online signature verification using a new extreme learning machine based on deep belief networks. *Soft Computing*, 21(5), 1345-1352.
- Yao, G., Lei, T., & Zhong, J. (2019). A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118, 14-22.
- Yılmaz, M. B., & Yanıkoğlu, B. (2018). Score normalization for signature verification systems. *Pattern Recognition Letters*, 115, 24-30.
- Zagoruyko, S., & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4353-4361.
- Zhang, B. (2006). Off-line Signature Recognition and Verification by Kernel Principal Component Self-regression. *Proceedings of the 5th International Conference on Machine Learning and Applications (ICMLA'06)*, 0-7695-2735-3/06, 4 - 6.
- Zhang, Z., Liu, X., & Cui, Y. (2016). Multi-phase offline signature verification system using deep convolutional generative adversarial networks. *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, 2.
- Zois, E. N., Aleiferis, P. G., & Loukas, A. (2018). DeepSiamese: A deep learning-based approach for offline signature verification. *Information Fusion*, 43, 52-62.