International Journal of Advanced Trends in Multidisciplinary Research
ISSN: 3139-0986, Volume 2 Issue 7
July 2026, PP 1-6

# Online Signature Verification Using Deep Learning

[1]S. Rama Rao, [2]M.Radhika, [3]M.Prasanna Kuamr
[1]Professor, Department of CSE, MAM Women's Engineering College, Narasaraopet, Palnadu, A.P., India.
[2,3]Assoc. Prof., Department of CSE, MAM Women's Engineering College, Narasaraopet, Palnadu, A.P., India.

***Abstract* – Online signature verification (OSV) systems are vital for enabling secure authentication in digital environments that depend on electronic signature collection and verification. Not only do these systems examine the static structure of a signature, but they also use deep learning and machine learning techniques to assess its distinct dynamic properties, such the pen's pressure, speed, and stroke sequence. These days, researchers look at the distinct static and dynamic aspects of signatures using methods like hybrid models, RNNs, CNNs, and Generative Adversarial Networks (GANs). A major drawback of the aforementioned deep learning approaches for building dependable and responsive real-time systems is the constant training of these models in response to database updates involving new users. Our plan is to create an OSV system that uses a Convolutional Neural Network (CNN) and a ReactJS website so that users may save or upload their signatures to a database. Based on the degree to which the user's uploaded signature matches their original signature, the model uses the spatial information gathered from the signatures to make judgements. In order for the system to handle both genuine and fake signatures, it is regularly trained. This system aims to provide identity verification for critical digital applications and online transactions in a secure, reliable, and resilient manner. To do this, we combine methods for preprocessing signatures, extracting features, and building classification models.**

***Keywords* – Online signature verification, ReactJS, feature extraction, Convolutional Neural Network, and Siamese Network are index terms.**

## I. INTRODUCTION

The fact that most contemporary interactions and transactions occur online further highlights the need for reliable authentication solutions. One popular kind of identification that uses a person's signature and its unique features to confirm their identity is online signature verification. As a biometric method, signatures are far harder to copy than easily stolen passwords or PINs. Among the many potential applications of OSV are financial transactions, access control, and identity verification [1] [2] [3]. Certain signature verification approaches may be used to verify documents [4]. When compared to more traditional methods, OSV systems provide a better user experience and are easier to use. These solutions use people's natural signing habit to do away with cumbersome passwords or physical tokens. The authentication technique is made simpler while yet ensuring good security using this solution. With more and more people working remotely or using online services, OSV makes it possible to authenticate securely from anywhere. Some OSV systems absolutely must have real-time signature recording and analysis capabilities [5] [6]. Unlike traditional signatures that are taken from printed photographs or documents, digital signatures are obtained in real-time utilising digital input devices such as touch screens or digital pens. Online signature verification may be a game-changer, but there are a lot of problems that are preventing it from gaining traction. Various input device characteristics, environmental factors, and signature styles might introduce noise and inconsistencies into this data. Building reliable and accurate verification procedures is more difficult due to this. Cybercriminals might use forged digital signatures to get unauthorised access to private data. If users can't access devices that can capture dynamic signature traits, OSV's resiliency on different platforms can be damaged. New methods and advances in OSV technology are being considered by researchers as potential solutions to these issues. Deep learning algorithms are often more resistant to noise and distortions than their more traditional counterparts. In addition to the spatial organization of the letters, a signature also includes curves, strokes, velocity, and pressure. Conventional approaches to verification do not fully use these characteristics. Deep learning algorithms can self-adapt to changes over time, identify minute pattern deviations, and extract complicated information from images by continuously learning from new data. For this reason, the algorithms outperform the more traditional approaches to signature verification. The accuracy of OSV systems has been greatly enhanced by using CNN and other deep learning approaches [7, 8]. By improving the capture of both local and global feature representations, transformer-based methods have recently increased the quality of signature verification [9, 10]. The use of self-adjusting authentication systems in reaction to predefined risk criteria is one flexible approach to real-time identity verification. To improve the reliability and precision of authentication systems, researchers have turned to machine learning and deep learning, which have shown encouraging outcomes. While addressing present difficulties and potential future study avenues in signature verification, recent systematic reviews have emphasised the efficacy of biometric authentication and machine learning approaches [11, 12]. The second section of this article provides an overview of the literature on signature verification. In Section III, the proposed system is described in depth in the article. The article's Part IV describes the dataset used for model training and evaluation. Section V of the paper presents the results and metrics that were utilised to evaluate the system. In the last section of the paper, we talk about what we want to accomplish in the future.

## II. RELATED WORKS

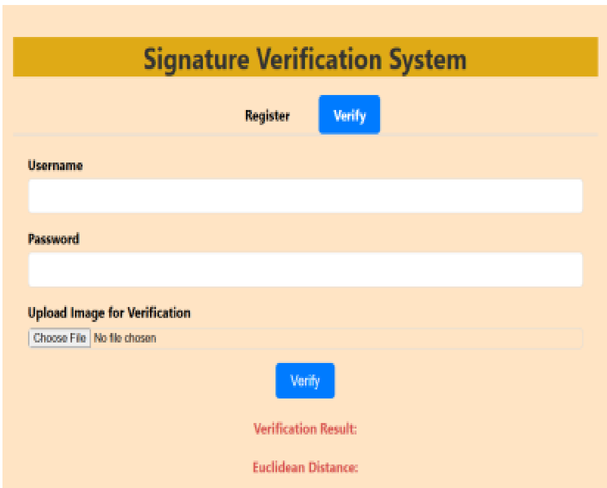
Signature verification techniques have been significantly enhanced by the widespread use of deep learning technology. Hafemann et al. [13] proposed a system that

International Journal of Advanced Trends in Multidisciplinary Research
ISSN: 3139-0986, Volume 2 Issue 7
July 2026, PP 1-6

could detect fake or real signatures by training deep convolutional neural networks (CNNs) on the original's properties. When training the model, we made use of both authentic and fictitious user signatures. Combining depth-wise and point-wise convolutions, a method was proposed by Chandra SekharVorugunti et al. [14] to enhance the discriminating capability of conventional CNNs. This method is called Depth-Wise-Separable CNN. To categorise signatures, we used this CNN in combination with LSTM architecture. However, user-forged signatures are required in order to train a model for signature categorisation. Convolutional neural networks (CNNs) train poorly on insufficient datasets. In an unbalanced dataset, they may provide biased results. Since faked signatures are common, their absence can impact the model's training accuracy. Generative Adversarial Networks (GANs) were created to tackle the problem of a small dataset. Vorugunti et al. [15] suggested a GAN that mainly includes a generator and an image discriminator to produce fresh false signatures. Since the signature quality improves with each cycle, this aids in training the discriminator to grow better at categorising. Other GAN enhancements, such as CycleGAN, which targets image-to-image translation, have been suggested over the years. It is possible for CycleGAN to remove any noise, such as a mark or stamp, when it verifies a signature. Due to their overfitting vulnerability, GANs could miss tiny but meaningful distinctions between real and synthetic imposters. Convolutional neural networks (CNNs) excel with picture data but struggle with sequential inputs. Textual signatures and other sequential data may be handled in part by using a Recurrent Neural Network (RNN) strategy. The practicality of RNNs was investigated by Ruben Tolosana et al. [16] in relation to signature verification. A new subclass of RNNs called Long Short Term Memory (LSTM) RNNs has recently arisen as a potential substitute for RNNs in prediction tasks that need long-term dependency memory accuracy. Integration of transformers with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) has been the subject of recent research aimed at improving the prediction capabilities of these models [17]. An important aspect for constructing a signature verification system is that transformers remember the context of earlier data, in contrast to RNNs that are great at processing sequential input.

## III. PROPOSED SYSTEM ARCHITECTURE

Websites built using JavaScript, CSS, and HTML are known as "frontend" and "backend" developers, respectively. The system's user interface is shown in Figure 1. One part of the user interface is a login form that requests a login and password. Flask, a framework for lightweight Python applications, was used to build the backend. Because it supports GridFS for picture storage, MongoDB is used as the database in this project. A well-known NoSQL database MongoDB is schema-flexible and provides horizontal scalability [18]. It uses documents, which are analogous to JSON objects, to store data. A document that has a signature is considered part of the collection; these documents are like entries in a table in a SQL database [19]. The name of the collection is the user's name in this instance. There is a unique identifier for the associated user's signature in addition to the document ID and username that are stored in each item in the collection. Divide the picture into smaller parts using GridFS once you send a signature to the database. At a later stage, the fs.chunks collection stores these components as individual documents. Explore the fs.files collection to learn all there is to know about the signature. File size, type, and references to fs.chunks chunks are all part of it. A unique identification from the user collection is used by GridFS to access the signature image in the fs.files and fs.chunks collections. This identifier is part of the signature file id. Users are able to enter their login information and submit the necessary signature on this page. We can check the authenticity of the user's signature by looking it up in the database. B. Extracting spatial feature images is crucial for signature analysis because to the complexity of signatures as visual patterns. Image processing is a strong suit of convolutional neural networks (CNNs) designed using the Siamese architecture. A similarity measure is determined by the two input signatures to the two Convolutional Networks of the CNN-based Siamese Network. whether you need to determine whether two inputs are similar or different, a Siamese Network is your best option. As a result of sharing architecture and weights, the model's two convolutional neural networks (CNNs) undergo a quick sequence of convolution and pooling operations on an input set of signatures to




The system's user interface (Fig. 1) allows users to submit signatures and authenticate themselves.

Acquire high-level feature representations; users may add signatures to the database via the register page and validate the submitted signatures' authenticity using the verify tab. Using a contrastive loss function, the feature representations of the two convolutional neural networks are combined. The distance between the two points will be calculated using the conventional formula after this

International Journal of Advanced Trends in Multidisciplinary Research
ISSN: 3139-0986, Volume 2 Issue 7
July 2026, PP 1-6

function has learnt the similarity metric. A common rule of thumb is that the loss function instructs the model to enhance the distance between unique data points and minimise it between similarities. There is an extra signature, which may be legitimate or a combination of the two, linked to every input signature. According to Figure 2, a Siamese network looks like this. The design of the experimental model is derived on the research of [20]. The input photographs, which are a combination of convolutional and fully connected layers, are two forms with dimensions of 224 × 224 × 3. Convolutional layers are one of the first phases. A 7 x 7 kernel and 96 filters are used in the first convolution. The spatial dimensions are reduced to 112 × 112 via max pooling, which allows the main characteristics to be zeroed in on afterwards. Afterwards, a further convolution is executed using 256 filters and 5 x 5 kernels. Lastly, the dimensions are reduced to 56 × 56 by means of an additional pooling layer. Two further convolutional layers with 256 filters and 3 x 3 kernels each are then added. After max pooling, the spatial dimensions are reduced to 27 × 27, and regularisation is accomplished using intermediate dropout layers. Converting the result into an 186,624 element vector is the next step. A two-layer fully linked architecture, consisting of one with 128 dimensions and the other with 1024 units and a 50% dropout, is used to manage this vector. Once this network has processed both input photos separately, you can compare the feature embeddings it produced to see how similar they are.

**Dataset**

Kaggle1 was used to get the experimental dataset. The dataset is based entirely on the ICDAR 2011 Signature Dataset. The dataset contains a grand total of 2149 photos, with 69 actual classes and 69 fictitious classes split evenly between the training set and the testing set. Although there are 5,747 signature pairs in the testing set and 23,205 in the training set, both sets include signatures. There are two alternatives available: formed signatures and genuine-genuine signatures. The real signature of a dataset user is shown in Figure 3, whereas the false signature of the same person is shown in Figure 4.

## IV. EXPERIMENTATION AND RESULTS

Evaluation metrics are very helpful for tasks like signature verification when assessing the performance and effectiveness of a model. The metrics that should be employed are determined by the system's intended use case. It provides a high-level overview of the model's efficiency. Before diving into a breakdown of the different metrics, we'll go over some definitions of crucial concepts: 1. A True Positive (TP) situation occurs when the signature's validity is correctly ascertained. 2. FP: When a forger's signature is erroneously believed to be authentic, this is known as a false positive. Thirdly, when the authenticity of an imposter's signature is definitively confirmed, it is called a True Negative (TN). 4. When a real signature is incorrectly thought to be fake, it's called a False Negative (FN).

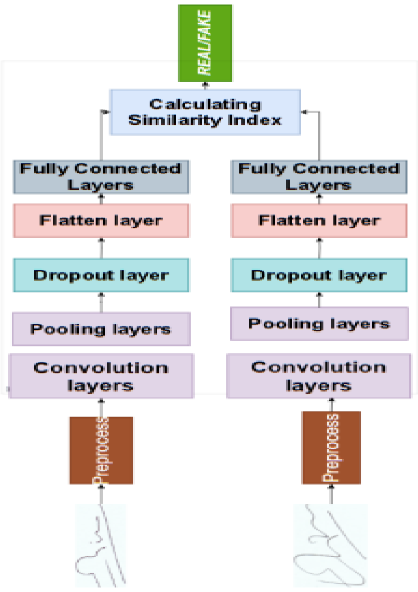
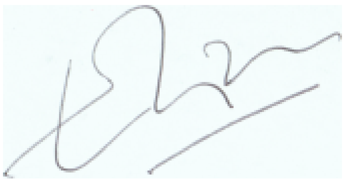



Fig. 2. Workflow of the Proposed Signature Verification Model

As seen in Figure 2 of the model's overall design, the implemented model receives two signatures as inputs. It is necessary to preprocess these signatures before feeding them into the model. The model's output feature vectors are compared to determine the similarity index.



The real signature of a dataset sample user is shown in Figure 3.

Figure 4 shows the same example user's forged signature.

The ratio of the number of properly identified real signatures to the total number of genuine signatures (both right and wrong) is the measure of precision in signature verification. Assuming the model successfully reduces the validity of forgeries, according to the high degree of accuracy.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Among the several measures of memory, the True Positive Rate (TPR) accounts for the proportion of valid classifications that properly detected real signatures. The likelihood of the model rejecting legitimate signatures

International Journal of Advanced Trends in Multidisciplinary Research
ISSN: 3139-0986, Volume 2 Issue 7
July 2026, PP 1-6

decreases as recall increases. It is crucial to avoid accidentally tagging actual signatures.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Accuracy is the proportion of correct classifications. When considering both real and fraudulent signatures, accuracy gives a good picture of how well the model predicts. Look at this:

$$CorrectClassifications = TP + TN \quad (3)$$

$$TotalClassifications = TP + TN + FN + FP \quad (4)$$

$$Accuracy = \frac{CorrectClassifications}{TotalClassifications} \quad (5)$$

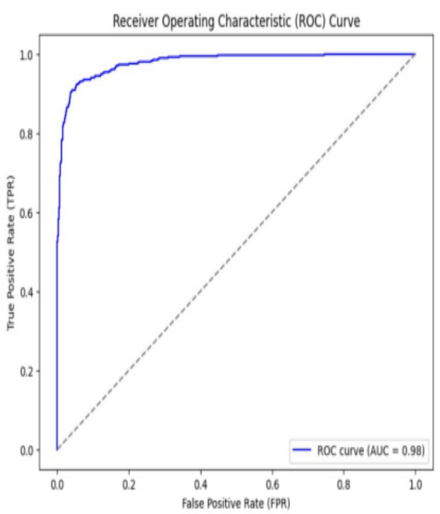
What we have here is the amount of the dataset used to test the model, TotalClassifications, and the number of signatures that the model correctly classified as authentic and forged, respectively, called CorrectClassifications and TotalClassifications, respectively.

Part D: Findings from Formula One The F1 score is a measure of a model's accuracy in identifying each class that considers both recall and precision. It finds the sweet spot where accuracy and memory meet.

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

Section E: Specifics This metric illustrates the proportion of actual forged signatures that the model incorrectly identifies as phoney. Another name for it would be "False Positive Rate" (FPR).

$$Specificity = \frac{FP}{FP + TN} \quad (7)$$




With FPR on the x-axis and TPR on the y-axis, Figure 5 displays the ROC curve. Also included is the area under the curve (AUC) value, a general indicator of how well the model performed. The AUC of 0.98 is the defining feature of this model.

The Assessment and Outcomes (E) In order to make the dataset's signatures match the dimensions of the model, which were (224,224,3), they were shrunk. We used a portion of the training dataset consisting of around 10,000 signature pairs for the purpose of training the model and testing the system. We proceeded to train it to make predictions using the testing dataset, which had 5747 signature pairings (2777 authentic-genuine and 2975 authentic-forged). In order to construct our framework, we relied on the Keras library. The model's execution yields two vectors as its output. To do this, we compare the vectors' geometric distances to an established value. The receiver operating characteristic (ROC) curve is one kind of graph that shows how well a binary classifier performs across different thresholds. By plotting TPR against FPR at different threshold values, the ROC curve-creation procedure yields the result shown in Figure 5. On the curve, you can see a cutoff value at every point. The receiver operating characteristic (ROC) curve illustrates the trade-off between sensitivity and specificity. Area under the ROC curve (AUC) is the only statistic that can be used to assess the model's performance. The area under the curve (AUC) is a great measure of how well the model can classify data at different threshold levels. I can get an AUC between 0 and 1. The AUC value of 0.98 is shown in Figure 5. To achieve optimal threshold tuning, it is necessary to first identify the optimal balance between TPR and FPR. Based on the results of the experiment, a threshold value of around 0.24 is optimal. We used the obtained threshold value to create the model's confusion matrix for the test dataset. Below, in Table I, you may see it. You may see the results of the experiments in Table II II. Among them, you may find the f1 score, recall, accuracy, and precision of the model. The counts of True Negatives, False Positives, True Positives, and True Negatives using the model are shown in Table I.

| | Predicted:Real | Predicted:Forged |
|---|---|---|
| Actual:Real | 2572 | 200 |
| Actual:Forged | 198 | 2777 |

Table 2: This table displays the many metrics used to evaluate the model's performance.

| Metric | Value |
|---|---|
| Precision | 92.85% |
| Recall | 92.78% |
| F1 Score | 92.81% |
| Accuracy | 93.07% |

The model successfully identifies genuine signatures with few false positives, as shown by its remarkable 92.85% accuracy rate in the signature classification test. With a recall score of 92.78%, it is clear that the model is effectively recognising actual signatures while exhibiting no false negatives. At 92.81%, our f1 Score is 0.9281. An excellent model at avoiding misclassifications while correctly detecting actual signatures has a high f1 score. With an impressive overall accuracy of 93.07%, the model proves to be reliable for real-world applications by successfully classifying signatures in the majority of cases.

## V. CONCLUSION

In this research, we introduced a Siamese network that uses CNNs as its foundation for online signature verification. Using the open-source Kaggle dataset, the study demonstrated how the model was tested and evaluated. The metrics used for assessment include recall, accuracy, f1-score, and precision of the model. A precision of around 93% is achieved. Because of its Siamese-based design, the model is able to outperform previous algorithms while using far less data. There is room for improvement. The only aim of using a subset of the training dataset was to train the models. It is possible that bigger subparts may improve the model's performance. The performance problem may be fixed if more convolutional and pooling layers were added to the model and more epochs were performed on the dataset. Our next priority will be to enhance the model's performance, consistency, and applicability.

## REFERENCES

1. M. C. Fairhust and S. Ng, "Management of access through biometric control: A case study based on automatic signature verification," Universal Access in the Information Society, vol. 1, pp. 31–39, 2001.
2. E. J. Potter, Customer authentication: The evolution of signature verification in financial institutions. PhD thesis, Citeseer, 2002.
3. J. Putz-Leszczynska and M. Kudelski, "Hidden signature for dtw ´ signature verification in authorizing payment transactions," Journal of telecommunications and information technology, no. 4, pp. 59–67, 2010.
4. D. Engin, A. Kantarci, S. Arslan, and H. K. Ekenel, "Offline signature verification on real-world documents," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2020.
5. J. Huang, Y. Xue, and L. Liu, "Dynamic signature verification technique for the online and offline representation of electronic signatures in biometric systems," Processes, vol. 11, no. 1, p. 190, 2023.
6. X. Song, X. Xia, and F. Luan, "Online signature verification based on stable features extracted dynamically," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 10, pp. 2663–2676, 2016.
7. C. S. Vorugunti, V. Pulabaigari, R. K. S. S. Gorthi, and P. Mukherjee, "Osvfusenet: online signature verification by feature fusion and depthwise separable convolution based deep learning," Neurocomputing, vol. 409, pp. 157–172, 2020.
8. C. S. Vorugunti, P. Mukherjee, V. Pulabaigari, et al., "Osvnet: Convolutional siamese network for writer independent online signature verification," in 2019 international conference on document analysis and recognition (ICDAR), pp. 1470–1475, IEEE, 2019.
9. H. Li, P. Wei, Z. Ma, C. Li, and N. Zheng, "Transosv: Offline signature verification with transformers," Pattern Recognition, vol. 145, p. 109882, 2024.
10. V. Bharadi, Z. Hamdule, S. Kambli, R. Salvi, R. Chavan, and V. Nimbalkar, "Online signature recognition using transformer networks: An overview," in 2023 6th International Conference on Advances in Science and Technology (ICAST), pp. 116–121, IEEE, 2023.
11. M. M. Hameed, R. Ahmad, M. L. M. Kiah, and G. Murtaza, "Machine learning-based offline signature verification systems: A systematic review," Signal Processing: Image Communication, vol. 93, p. 116139, 2021.
12. K. Bibi, S. Naz, and A. Rehman, "Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities," Multimedia Tools and Applications, vol. 79, no. 1, pp. 289– 340, 2020.
13. L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," Pattern Recognition, vol. 70, pp. 163–176, 2017.
14. C. SekharVorugunti, V. Pulabaigari, P. Mukherjee, and A. Sharma, "Deepfuseosv: online signature verification using hybrid feature fusion and depthwise separable convolution neural network architecture," IET Biometrics, vol. 9, no. 6, pp. 259–268, 2020.
15. C. S. Vorugunti, P. Mukherjee, and V. Pulabaigari, "Online signature profiling using generative adversarial networks," in 2020 International Conference on COMmunication Systems &NETworkS (COMSNETS), pp. 894–896, IEEE, 2020. [
16. R. Tolosana, R. Vera-Rodriguez, J. Fierrez, and J. Ortega-Garcia, "Biometric signature verification using recurrent neural networks," in 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 652–657, IEEE, 2017.
17. T. Do Thanh, C. T. Nguyen, N. H. Phung, N. H. Minh, and V.- H. Nguyen, "Vit-signet: Combining deep cnn and vision transformer for enhanced signature verification," in International Conference on Advances in Information and Communication Technology, pp. 215–224, Springer, 2023.
18. A. Chauhan, "A review on various aspects of mongodb databases," International Journal of Engineering Research & Technology (IJERT), vol. 8, no. 05, pp. 90–92, 2019.

19. C. Gyor˝ odi, R. Gy ¨ or˝ odi, G. Pecherle, and A. Olah, “A comparative ¨ study: Mongodb vs. mysql,” in 2015 13th international conference on engineering of modern electric systems (EMES), pp. 1–6, IEEE, 2015.
20. S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Llados, and U. Pal, ´ “Signet: Convolutional siamese network for writer independent offline signature verification,” arXiv preprint arXiv:1707.02131, 2017.