Shaik Nakarikanti Shabana, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Online Signature Verification Using Siamese Convolutional Neural Networks for Secure Digital Authentication

Shaik Nakarikanti Shabana[1], P. Anusha[2]

[1]Student, [2]Assistant Professor, Department of CSE,AI,ML, MAM
Women's Engineering College, Kesanupalli(Narasaraopet)

Abstract- Secure authentication in digital settings that electronically collect and verify handwritten signatures relies heavily on Online Signature Verification (OSV) technologies. These systems analyse not only the static structure of a signature but also its unique dynamic properties like pen pressure, pace, and stroke order using machine learning and deep learning methodologies. Hybrid models, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generator Adversarial Networks (GANs) are some of the current methods used to examine the distinct static and dynamic aspects of signatures. The above deep learning approaches provide a challenge when it comes to developing responsive and dependable real-time systems. This is because these models need to be trained every time a new user is added to the database. Our experiment's goal is to construct an OSV system that comprises a ReactJS-built website for signature uploading or database storage and a CNN-based Siamese Network for OSV integration. The model uses the spatial information extracted from the signatures to make judgements depending on how close the uploaded signature is to the user's original signature. Through ongoing model training, the system is prepared to deal with both genuine and fake signatures. This system is designed to provide a safe, dependable, and resilient way to verify identification in online transactions and sensitive digital applications. It does this by combining signature preprocessing methods, feature extraction, and classification models. Things like online signature verification, siamese networks, react-js, and feature extraction are all part of the index.

Keywords: RC framed structures, masonry infill, seismic analysis, ABAQUS, AAC blocks, retrofitting, CFRP, time history analysis, soft storey, earthquake resistant structures.

I. INTRODUCTION

Since most modern-day interactions and transactions take place online, trustworthy authentication mechanisms are more important than ever. Online signature verification is a common kind of identification that relies on the distinct characteristics of a person's signature to validate their identity. Signatures provide a very secure biometric alternative to readily guessed passwords or PINs. Among the many possible uses for OSV, some of the most promising are identity verification, access control, and financial transactions [1, 2] [3]. It is possible to verify documents using signature verification systems [4]. Furthermore, compared to conventional approaches, OSV systems improve both efficiency and the user's overall experience. These solutions

eliminate the need for complex passwords or physical tokens by use people's natural signing habit. This method guarantees strong security while simplifying the authentication procedure. With the rise of remote work and online services, OSV allows for safe authentication regardless of where you are.

The ability to record and analyse the dynamic features of a signature in real time is essential to some OSV systems [5] [6]. Online signatures are captured in real-time using digital input devices like digital pens, touch screens, or similar technology, as opposed to static signatures that are collected from printed images or documents. Despite its potential and importance, online signature verification is hindered by a number of issues that prevent it from being widely used and effective.

There may be inconsistencies and noise in the data due to differences in signature styles, ambient conditions, and input device features. Building trustworthy and precise verification methods becomes more challenging as a result. In order to get access to sensitive information, individuals imitate online signatures. The resilience of OSV on various systems may be compromised if users do not have access to devices that can capture dynamic signature features.

Researchers are looking at new approaches and technology developments in OSV to tackle these problems. Many algorithms developed in deep learning have shown to be more robust against noise and distortions than more conventional approaches. There are a lot of distinguishing characteristics of a signature, including the shape, size, velocity, pressure, and spatial arrangement of the letters. Conventional verification methods fail to make adequate use of these capabilities.

By constantly learning from fresh data, deep learning algorithms are able to automatically adapt to changes over time, detect minor variances in patterns, and capture complex information from pictures. This gives the algorithms, in comparison to more conventional methods, a better fit for the job of verifying signatures. The OSV systems' accuracy has been significantly enhanced by using deep learning technologies like CNN [7] [8]. The performance of these verification techniques has been significantly enhanced with the arrival of transformers [9] [10]. One adaptable approach to real-time identity verification is authentication systems that self-adjust in response to certain risk criteria.

Recent studies in deep learning and machine learning[11,12] have shown promising results in enhancing the robustness and precision of authentication systems, which might lead to the creation of a more streamlined and secure authentication process. Section II of this study outlines previous research in the area of signature verification. In Section III, the paper lays out its suggested system. The dataset used to train and assess the model is detailed in Section IV of the article. Details about the system's evaluation measures and their outcomes are included in Section V of the article. The article wraps up with a discussion on future efforts in the final part.

II. RELATED WORKS

The efficiency of signature verification systems has been enhanced by the use of several deep learning technologies in recent years. Hafemann et al. [13] presented a system that could detect faked or authentic signatures using deep convolutional neural networks (CNNs) trained on the properties of the original. Users' actual and phoney signatures were used to train the model. By combining depth-wise and point-wise convolutions, the Deep-Wise-Separable CNN was introduced by Chandra Sekhar Vorugunti et al. [14] as an attempt to enhance the discriminating capacity of the conventional CNN.

For the purpose of signature categorisation, this CNN was combined with LSTM architecture. Unfortunately, using fake user signatures to train a signature classification algorithm is an inevitable part of any such project. If there is insufficient data to train CNNs, their performance will suffer. If there is an imbalance in the dataset, they could provide biassed results. The training accuracy of the model is susceptible to shortages in the amount of falsified signatures, which occurs often. The

problem of the little dataset was addressed by introducing Generative Adversarial Networks (GANs). Vorugunti et al. [15] suggested a GAN that uses a discriminator to attempt to categorise pictures and a generator to create new false signatures. As a result, the discriminator may be trained to produce better classifying signatures over time.

Many GAN enhancements, such as the image-to-image translation-focused CycleGAN, have been suggested throughout the years. If a signature contains noise, such markings or stamps, CycleGAN may remove it from the signature verification context [4]. The use of GANs is not without its risks, such as overfitting and the potential inability to detect tiny but crucial characteristics of human-made forgeries.

Convolutional neural networks (CNNs) excel at processing picture data, but they struggle when faced with sequential input. Recurrent Neural Networks (RNNs) are suggested as a means to handle sequential data, which includes textual signatures. While researching the potential of RNNs for signature verification, Ruben Tolosana et al. [16] did some digging.

The Long Short Term Memory (LSTM) RNN comes into play as an alternative to RNNs due to its ability to keep the long-term dependencies necessary for effective prediction, something that RNNs are unable to do. Recent studies have focused on improving the models' prediction skills by combining transformers with convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [17]. Similarly to RNNs, transformers analyse sequential data; however, in order to build an optimal signature verification system, transformers may retain the context of the prior data.

III. PROPOSED SYSTEM ARCHITECTURE

Chapter One: Frontend and Backend

The website was developed with the help of JavaScript, CSS, and HTML. The system's user interface is seen in figure 1. There is a login/password prompt on the user interface for authenticating the user. The Python framework Flask, which is designed for lightweight applications, has been used to build the backend. Because of its support for GridFS image storage, MongoDB was chosen as the database for this project. Thanks to its horizontal scalability and schema flexibility, MongoDB has become a popular NoSQL database [18]. It uses documents, which are analogous to JSON objects, to store data. The documents containing the signatures are housed in collections, which are like entries in a table in a SQL database [19].

The user's name serves as the collection's identifier. In addition to the document's unique identification (ID), the signature of the corresponding user, and the username are all included in each document in the collection. As soon as a signature is uploaded to the database, GridFS divides the picture into smaller pieces and puts each one in its own document in the fs.chunks collection. Information about the signature, including the file type, size, and references to the chunks in fs.chunks, may be found in the fs.files collection. For each signature image saved in the fs.files and fs.chunks collections, GridFS utilises the distinct identifier included in the signature file id in the user collection to locate the specific picture.

Users are able to submit the necessary signature and input their username on the site. In order to determine whether the submitted picture is authentic or a fake, the user's signature is obtained from a database and compared with the uploaded signature. B. Siamese Architecture and CNN-based Systems Given that signatures are intricate visual patterns, Convolutional Neural Networks (CNNs) are essential for image-related tasks like extracting spatial feature pictures. Siamese Network, which is based on Convolutional Neural Networks (CNNs), uses two signals to calculate a similarity measure.

When solving problems that require determining the degree of similarity or difference between two inputs, Siamese Networks are ideal. Two convolutional neural networks (CNNs) in the model, which have identical architecture and network weights, take a pair of signatures as input and

perform sequential convolution and pooling operations to

Fig. 1: User Interface of the system: It provides the option for authentication and uploading signature. The register tab allows user to add signature to database while verify tab allows to verify the uploaded signature.

provide representations of features at a high level. By learning the similarity metric and calculating the distance between them using the Euclidean formula, a contrastive loss function is used to connect these feature representations from both CNNs. The loss function encourages the model to minimise the distance between similar pairs of data points and maximise the distance between different pairs.

We couple the input signatures based on whether they are authentic or a mix of real and fake. Figure 2 shows the layout of a Siamese network. The experimental model's architecture is based on the work of [20]. It uses a sequence of convolutional and fully connected layers to handle two input pictures with dimensions 224 × 224 × 3. The process starts with a series of convolutional layers. The first convolution uses 96 filters and a 7 × 7 kernel. Then, using max pooling, the spatial dimensions are reduced to 112 × 112, allowing the most essential characteristics to be focused on.

Next comes a second convolution that uses 256 filters and 5 x 5 kernels, followed by yet another pooling layer that reduces the dimensions to 56 × 56. Following this, two further convolutional layers with 384 and 256 filters and 3 x 3 kernels are added. Then, after max pooling, the spatial dimensions are reduced to 27 × 27, and regularization is accomplished via intermediate dropout layers. Afterwards, the output is compressed into an 186,624 element vector. The vector is transformed into a 128-dimensional vector after passing through two completely linked layers. The first layer contains 1024 units and a 50% dropout. You may compare the two feature embeddings that this network produces after processing the input pictures separately to see how close they are.

IV. DATASET

All of the data utilised in the experiments came from the ICDAR 2011 Signature Dataset, which is a dataset that was collected via kaggle1. There are a total of 2,149 photos in the dataset, with 69 fictitious classes and 69 real classes in the training set and 21 fictitious classes and 21 real classes in the testing set.

A grand number of 23,205 signature pairs make up the training set, while 5747 pairings make up the testing set. There are two types of signature pairs: genuine-genuine and genuine-forged. Figure 3 depicts the real signature of a dataset sample user, whereas Figure 4 shows the mock signature of the same individual.

V. EXPERIMENTATION AND RESULTS

Particularly in activities such as signature verification, evaluation measures are used to evaluate the efficacy and performance of a model. The objective of the system dictates the metrics that should be used. It summarizes the model's performance in general. The following definitions of key concepts are provided before moving on to a discussion of the various metrics:

1. When a real signature is accurately recognised as such, it is called a true positive (TP).
2. Incorrectly identifying a bogus signature as real is known as a false positive (FP).
3. In a True Negative (TN) situation, the forgery of a signature is successfully detected as forgery.
4. False Negative (FN): This occurs when an authentic signature is mistakenly deemed to be counterfeit.

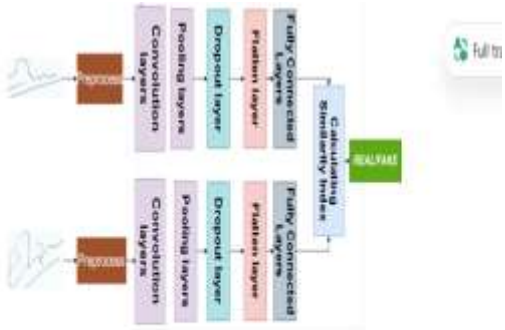

Fig. 2: The general architecture of the model: the model implemented uses two signatures as inputs. These signatures are preprocessed before providing them to the model. The model outputs feature vectors that are compared to obtain similarity index.

Fig. 3: **Real signature of sample user of the dataset**

Fig. 4: **Forged signature of the same sample user**

A. Precision What constitutes precision in signature verification is the ratio of the number of authentic signatures that are accurately recognised to the total number of signatures that are authenticated (incorrectly included). When the model's accuracy is high, it means it does an excellent job of reducing the likelihood of false positives.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

B. Remember Recall, often called True Positive Rate (TPR), measures how many real signatures were successfully detected as compared to the overall number of valid classifications. The likelihood of the model rejecting real signatures decreases as recall increases. It is critical to avoid incorrectly categorising real signatures.

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

C. Precision The percentage of right categorisations is called accuracy. Taking into account both authentic and fake signatures, accuracy paints a general image of the model's prediction accuracy. This is provided as

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

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

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

D. The size of the dataset used to test the model is shown by TotalClassifications, and the number of signatures that were properly categorised (i.e., accepted as authentic and forged) is denoted by CorrectClassifications. Level D. Formula One Performance By comparing the model's recall and accuracy, the F1 score determines how well it can categorise each class. Accuracy and memory work hand in hand.

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

E. Specificity The fraction of real forged signatures that the model wrongly identifies as genuine is represented by this measure. Another name for it is the False Positive Rate (FPR).

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

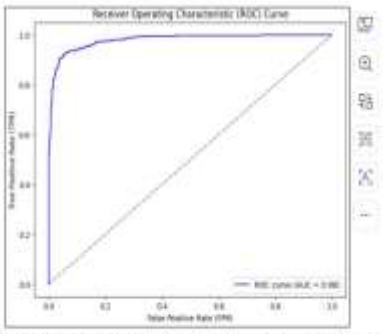

Fig. 5: The ROC curve is plotted with FPR on x-axis and TPR on y-axis. It also displays AUC value which denotes general performance of the model. The AUC value for the model is 0.98

F. Findings and Interpretation The dataset's signatures were scaled to (224,224,3) dimensions before being fed into the model. To assess the system, the model was initially trained on a subset of the training dataset containing approximately 10,000 signature pairs. Subsequently, it was trained to make predictions on the testing dataset, which contains 5747 signature pairs, with 2772 pairs of genuine-genuine signatures and 2975 pairs of

genuine-forged signatures. The Keras library has been used to construct our framework. The model's end result is a pair of input vectors.

A threshold is used to compare the vectors' Euclidean distances. A binary classifier's performance over several thresholds is shown by the ROC curve, a graph. By comparing TPR with FPR at different threshold values, the ROC curve seen in figure 5 is generated. At each node on the curve, you can find the associated threshold. One way to see the compromise between specificity and sensitivity is via the ROC curve. One way to measure the model's efficacy is by looking at its area under the receiver operating characteristic (ROC) curve (AUC).

A large area under the curve (AUC), which may take values between 0 and 1, indicates that the model performs better than the competition in classifying data at all threshold levels. Figure 5 displays the model's AUC value, which is 0.98. Finding the sweet spot between TPR and FPR is the goal of optimum threshold optimisation. The results of the experiment show that a threshold of around 0.24 is ideal. Table I I shows the model's confusion matrix, which was created after assessing the test dataset using the given threshold value. The experimental findings, including the model's accuracy, precision, recall, and f1 score, are shown in Table II II.

TABLE I: **The confusion matrix shows the number of True Positives, True Negatives, False Positives and False Negatives obtained using the model.**

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

TABLE II: **The table shows the different metrics that assess performance of the model.**

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

This impressive accuracy rate of 92.85% for the signature classification test shows that the model effectively detects real signatures with few false positives. With a recall score of 92.78%, it is clear that the model is successfully detecting real signatures while producing fewer false negatives. At 92.81%, the f1 Score is 0.9281. In this scenario, a high f1 score indicates that the model manages to accurately identify real signatures while avoiding misclassifications. The model is dependable for real-world applications due to its high overall accuracy of 93.17%, which demonstrates a great capacity to accurately categorise signatures in most circumstances.

VI. CONCLUSION

We presented a Siamese network for online signature verification that is based on convolutional neural networks (CNNs) in this study. Using the publicly accessible Kaggle dataset, the article demonstrated the experiments and evaluated the model. Accuracy, recall, f1-score, and precision are the metrics used to assess the model. An accuracy of around 93% is attained. The model's strength lies in its Siamese-based architecture, which manages to achieve remarkable results with little data training.

You can make it even better. A portion of the training dataset was used to train the model. So, it's possible that the model's performance may improve if the subparts were larger. It is possible that the performance might be much improved by increasing the number of epochs on the dataset and adding additional convolutional and pooling layers. We want to improve our model's efficiency, reliability, and generalizability in our future endeavours.

REFERENCES

1. M. C. Fairhust and S. Ng, "Management of access through biometriccontrol: 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 techniquefor the online and offline representation of electronic signatures inbiometric systems," Processes, vol. 11, no. 1, p. 190, 2023.
6. X. Song, X. Xia, and F. Luan, "Online signature verification based onstable 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 signatureverification," 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 Scienceand 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. Sekhar Vorugunti, 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. [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.