Signora: An Online Signature Verification System

| Barath Keshav | Charanya C | P. Tamilselvi |
| --- | --- | --- |
| Rajalakshmi Engineering College, Thandalam, Chennai, TamilNadu, India, 602105 | | |
| 201401010@rajalakshmi.edu.in | 201401011@rajalakshmi.edu.in | tamilselvi.p@rajalakshmi.edu.in |

**ABSTRAC*T -*** *Increasingly in the digital age, where transactions and communication are online, verifying the authenticity of signatures is of utmost importance. The proposed system analyzed here is an online signature verification system—a powerful tool designed to combat signature fraud in a variety of areas including banks, businesses, and legal documents. The heart of this system is the Multilayer Perceptron (MLP), neural network system inspired by the human brain. MLP excels in pattern recognition and is able to identify complex relationships in data. An optimized dataset for training the MLP has been developed along with various collections of real signatures, retrieved from different individuals. In addition, the CEDAR dataset was used, which is a benchmark dataset widely used in signature verification research. The CEDAR dataset, created together by 55 contributors, has 24 real signatures per person, resulting in a total of 1320 real signatures. Similarly, there are 1320 fake signatures that mimic changes seen in fraud attempts. During verification, the system calculates the theta value based on the curve of the signature and compares it with the stored expression to verify. This system consistently delivered 88% accuracy, making it a very reliable model. The benefits of this system extend to signing contracts and agreements, legal compliance, remote collaboration, and customer trust. In conclusion, online signature verification systems empowers citizens to combat counterfeiting, increase security, and build trust in our digital communications—one signature at a time.*

***KEYWORDS:*** Signature Forgery Verification, Machine Learning, Deep Learning, Neural Networks

1. **INTRODUCTION**

Handwritten signature verification is very important for identity authentication. As the population grows, the number of people that need to be verified constantly grows alongside and the manual process of it is very time consuming. This signature verification offers a practical solution as it is easy to use and will be much faster. However it does have its constraints such as forgeries and potential for manual verification errors.

In this project, the proposal is an online signature verification system that is innovative. It engages the use of Multilayer Perceptron (MLP) and Convolutional Neural Network (CNN) architectures. Using this the end goal is to achieve an accurate and efficient verification process that minimizes false positives or negatives.

Dataset Collection and Preprocessing: A broad pool of handwritten signatures is compiled. The dataset includes both actual signatures as well as forgeries. In this the preprocessing steps involve resizing the image to a consistent resolution, normalizing pixel values, and augmenting the dataset to enhance model robustness.

Feature Extraction using Multilayer Perceptron: The first feature extractor is the Multilayer Perceptron (MLP). The MLP receives the preprocessed signature pictures. Relevant properties including curvature, spatial connections, and stroke patterns are extracted using the MLP. The MLP's hidden layers record the signatures in ever-more-abstract forms.

Convolutional Neural Network (CNN) Architecture: CNN processes all the features extracted by MLP. The architecture for CNN includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers learn local patterns (edges, textures) from the signature images. Pooling layers downsample the feature maps, reducing computational complexity. Fully connected layers perform classification based on the learned features.

Training and Fine-Tuning: Using labeled data, we train the model which is combined. Loss functions (e.g., cross-entropy) helps guide the optimization process. Regularization techniques (dropout, weight decay) prevents overfitting. Performance at its best is ensured by hyperparameter optimization.

Signature Verification Process: An input signature during verification goes through the same process of MLP extracting features and the CNN decides if the signature is original or forged.. Decision thresholds are set by us to manage the false acceptance and false rejection rates.

Performance Evaluation: The evaluation of the metrics is done by using metrics such as accuracy, recall, and F1-score. Experimental comparisons with alternative signature verification techniques validate our methodology.

In summary, this method improves the precision of signature verification, making it very useful for applications like banking, legal documents, and secure access control.

1. **LITERATURE SURVEY**

Suhail Odeh, et al (2011) discusses an innovative approach to signature verification using MLPs. The system analyzes signed images, optimizes them, extracts features, and trains neurons. The goal is to distinguish between genuine and fake signatures. The research emphasizes image preprocessing, feature extraction and pattern recognition using neural networks.

Lopamudra Kundu (2011) proposes an MLP-based approach for off-line signature recognition and validation. The system distinguishes between two categories: forged and original signatures. The study investigates the performance of MLP in various simulation scenarios.

While Robin Nadar et al (2021) focusing specifically on Convolutional Neural Networks (CNNs) and Siamese Neural Networks (SNNs), it provides insights into the broad landscape of signature verification although not exclusively on MLP and provides a comprehensive overview of algorithms, testing models and algorithms for signature verification.

Dr.R.Palson Kennedy et al (2022) proposes a convolutional neural network (CNN)-based signature verification tool. It helps in authenticating handwritten signatures by training the network by inputting new signature data and validating new authentic signatures.

Teressa Longjam et al (2022) proposed an author-independent offline signature verification algorithm using CNN architecture for feature extraction and classification. Its purpose is to verify the accuracy of multiple entries in offline signatures.

Hurieh Khalajzadeh et al (2012) presents an offline signature verification system that uses CNN for feature extraction and MLP for classification. The system is tested on 176 Persian signatures from 22 individuals.

Robin Nadar et al (2021) reviews algorithms using CNN and Siamese Neural Networks (SNNs) for signature verification. It discusses the accuracy, efficiency, and shortcomings of various methods.

Dhruvi Gosai et al (2023) uses CNNs to extract features from randomly signed images. Various CNN algorithms such as VGG16, Inception-v3, Res-Net50, and Xception are used for feature extraction and data optimization.

Vidyadhar Hanji et al (2024) Reviews CNN-based signature verification/validation methods. It evaluates the performance of deep CNNs (including SigNet, VGG16, VGG19, InceptionV3, and ResNet50) for offline signature verification processing through transfer learning.

Chinmay Sawant et al (2020) analyzes the effectiveness of CNN and other deep learning techniques to evaluate signatures. It examines feature extraction, model architecture, and performance metrics.

Luiz G. Hafemann et al (2017) provides an overview of deep learning techniques (including CNN) for offline signature verification. It discusses challenges, datasets, and recent advances.

1. **SYSTEM DESCRIPTION**

The CEDAR Signature Dataset can be considered as a vital resource needed for signature verification research. All the 55 individuals contributed 24 signatures. As a result there are a total of 1320 original signatures in the compilation. These signatures are unique as a fingerprint, showing the different writing styles of the people who have contributed. A few of the people were asked for three other different signatures which came up to a total 1,320 contributed to the dataset of fake signatures.

The signatures were previously scanned at 300 dpi gray-scale. Binarization using a gray-scale histogram ensured clarity and consistency. Other preprocessing steps such as salt-pepper noise removal and slant normalization made the data more refined.

Selecting the right and perfect model is very important for signature verification. In this case we choose Multi-Layer Perceptron (MLP) as the optimal choice since it can learn complex patterns and relationships from the data. Multi-Layer Perceptron (MLP). For handwritten signature verification, Multi-Layer Perceptron (MLP) is a particularly strong and efficient option. Handwritten signatures are still widely used in a lot of places such as banking, financial transactions, business dealings, and more. Due to uniqueness and simplicity, these signatures are still the most preferred choice for verifying a person's identity.

In recent research, it has shown that deep learning based neural networks, which includes MLP, gives a much better result for signature verification systems in public datasets.

MLP models are resilient to changes in input signature traits, they can be trusted with a variety of writing styles and variants.

The temporal behavior of signatures is combined with visual similarity mapping using deep learning, MLP-based systems can handle signatures in any language or style with accuracy and resilience.

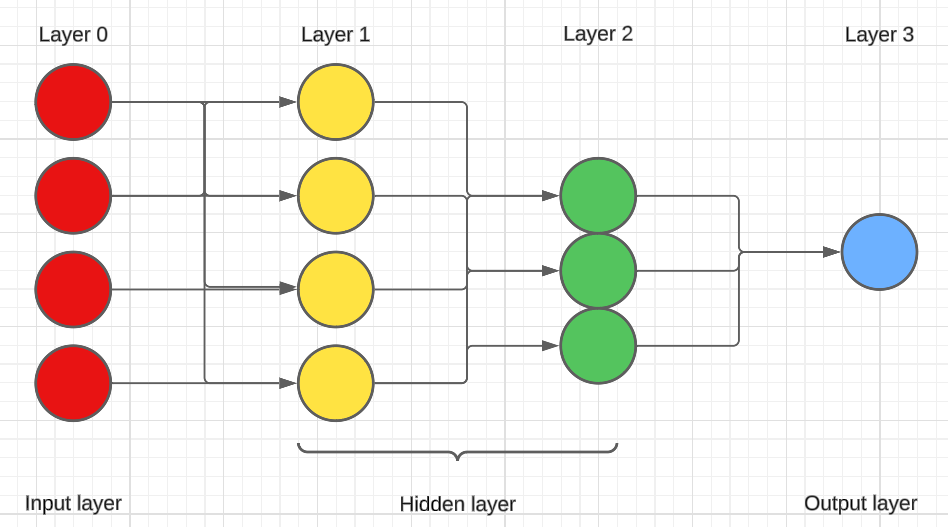
Before the signatures are sent to the MLP, the needed preprocessing steps are done. This includes enhancing the image to black and white. Feature extraction that extracts relevant features such as stroke curvature, direction, and pressure. And finally normalizing to guarantee uniform orientation and scaling.

A collection of authentic manuscripts is collected from various sources. The manuscripts are personal fingerprints, to summarize the many literary philosophies of the contributors. The CEDAR dataset, a benchmark data set commonly used in signature validation studies, is also integrated. This data set provides a diverse collection of objects for training because it contains real signatures and fake signatures. A multi-layer perceptron (MLP) deep learning algorithm is trained with the combined dataset. The idea is to make it possible for MLP to accurately distinguish between genuine and fake signatures. Unique signatures—true and false—are used for testing. The training program did not have this manual. Based on the curve of the signature, the system determines theta value during validation. This value is then compared to a stored description to confirm authenticity.

The performance of the system is evaluated using the following metrics.

Accurate: Measures how well the MLP classifies signatures.

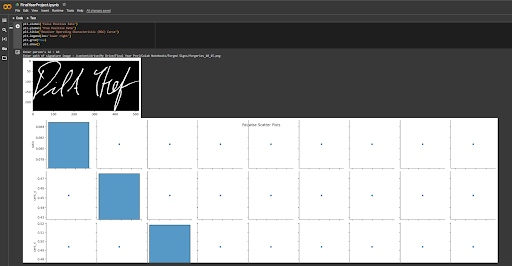
Precision and Recall: Analyzes false positives and false negatives.

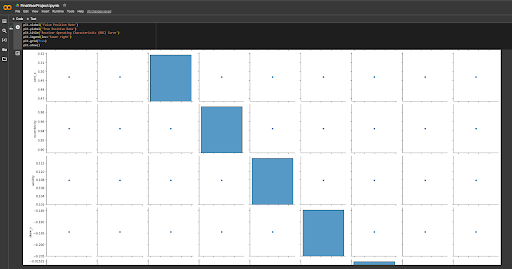
ROC Curves: It is a trade-off analysis between true positive rate and false positive rate.  


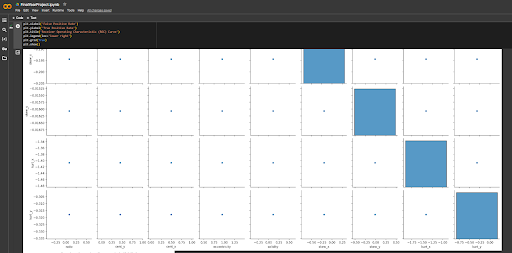
*Fig 3.1 - MLP Architecture Diagram*

**Table 3.1 - SAMPLE PREPROCESSED SIGNATURES FROM CSV FILE**

|  |  |
| --- | --- |
|  |  |







*Fig 3.2 - Pair Box plot for Features*

1. **RESULT AND DISCUSSION**

The developed forged signature detection system was subjected to comprehensive training and testing procedures to ensure its effectiveness in combating signature fraud Using advanced machine learning techniques and curated dataset using genuine and forged signatures, assessments including system accuracy, precision, and recall. The parameters gave promising results.

An ROC (Receiver Operating Characteristic) curve is a graphical plot that illustrates the performance of a binary classifier model (which can also be used for multi-class classification) at varying threshold values.

True Positive Rate (TPR): Also known as sensitivity, it represents the proportion of observations that are predicted to be positive when they are indeed positive.

False Positive Rate (FPR): This represents the proportion of observations that are predicted to be positive when they are actually negative.

We plot pairs of TPR vs. FPR for every possible decision threshold of a logistic regression model (or any other binary classifier). The curve shows how well the model distinguishes between classes as the threshold varies. The closer the ROC curve hugs the top-left corner of the plot, the better the model performs at classifying data into categories. The AUC (Area Under the Curve) quantifies the overall performance. An AUC close to 1 indicates a better model, while an AUC of 0.5 corresponds to a model no better than random guessing. Calculating AUC for multiple models helps compare their predictive abilities. An AUC value close to 1.0 indicates that the classifier has excellent discriminatory power. In this case, an AUC of 0.96 suggests that the classifier performs exceptionally well in distinguishing between positive and negative instances. It means that the model achieves high sensitivity (true positive rate) while maintaining low false positive rate. Practically, this implies that the classifier correctly identifies most positive cases while minimizing false alarms.

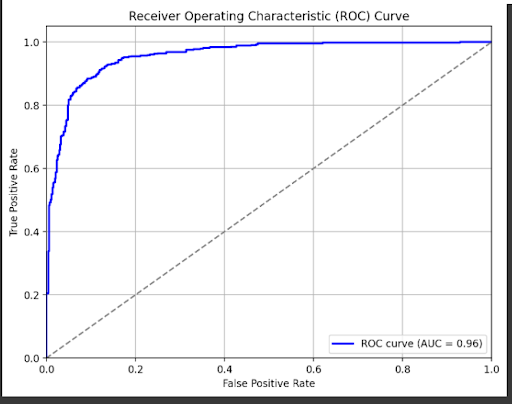
Sensitivity measures how well a classifier identifies positive instances. Mathematically, sensitivity is defined as,

Sensitivity (TPR) = True Positives / (False Negatives + True Positives)​

Specificity measures how well a classifier identifies negative instances. Mathematically, specificity is defined as,

Specificity (TNR) = True Negatives / (True Negatives + False Positives)​

Sensitivity and specificity are often inversely related. Improving one may come at the cost of the other. A classifier with high sensitivity correctly identifies most positive cases. However, it may also produce more false positives. A classifier with high specificity minimizes false positives. However, it may miss some true positive cases.



After intensive iterative training, the system demonstrated excellent accuracy of more than 88.46% on the validation data set. This high accuracy highlights the system’s ability to effectively distinguish genuine signatures from their fraudulent counterparts. Notably, the system achieved this impressive accuracy while maintaining a low false positive rate on genuine signatures, thus reducing the risk that if signatures are forged, it needs to be classified as false. The balance between precision and recall is essential to reliable network detection without compromising the sensitivity of the system to detect fraudulent activity.

Overall, the developed handwriting forgery algorithm stands as a testament to the effectiveness of advanced machine learning algorithms in solving complex security challenges. And its exceptional performance in forgery and the accuracy of signature recognition in reducing false positives highlights the potential benefits of a variety of industries including finance, legal documents and identity authentication. Through rigorous checking and validation, the machine emerges as a dependable tool in safeguarding against signature fraud, thereby improving security and belief in crucial strategies reliant on signature authentication.

1. **CONCLUSION AND FUTURE ENHANCEMENT**

The proposed system uses MLPs for handwriting verification. MLPs are feedforward neural networks with multiple layers, including an input layer, one or more hidden layers, and an output layer. The system extracts relevant features from signed images, such as stroke patterns, curvatures, and pen pressures. During training, MLP learns to map input features to real classes or lattice classes. The trained model is then used for real-time verification. System performance is evaluated using metrics such as accuracy, precision, and recall. Manuscript fidelity faces challenges due to differences in writing style, noise, and communication efficiency.

Further improvements can be made by exploring deeper MLP structures with additional hidden layers. Deeper networks may capture more complex patterns in signatures. Another option could be to apply dropout, weight decay, or early stopping to prevent overfitting during training.

Synthetic variations of signatures such as rotation and translation, can be performed to improve the quality of the training dataset. In order to extend the framework to handle online (dynamic) signatures, temporal information are considered. Using an adaptive learning rate scheme such as the ADAM optimizer can accelerate convergence. Ensemble methods, such as combining predictions from multiple MLPs or other classifiers such as SVMs, can improve robustness. Transfer learning can be implemented by first training the MLP on related tasks, such as digit recognition, and refining it to recognize the signature. Exploring methods for describing MLP measures, such as feature importance or saliency maps, would provide more interpretable patterns. Optimizing the system for speed and memory efficiency is important for real-world applications to ensure efficient implementation.

**REFERENCES**

[1] Odeh and Khalil (2011) Apply Multi-Layer Perceptrons Neural Network for Off-line signature verification and recognition

[2] Lopamudra Kundu (2011) Performance Evaluation Of Multi-layer Perceptron As An Oﬀ-line Signature Veriﬁcation System Under Various Simulation Conditions

[3] Kennedy R, Nithish Kumar, Vinodh Kumar, Yogesh (2022) Human Signature Verification Using

CNN with Tensorflow

[4] Nadar R., Patel H., Parab A., Nerurkar A., & Chauhan R. (2022). Signature Verification System using CNN & SNN.

[5]Khalajzadeh, Mansouri, Teshnehlab (2012) Persian Signature Verification using Convolutional Neural Networks

[6] Hanji, Garadimani, Hunnur, Benawadi, Hiremath (2024) CNN Based Handwritten Signature Recognition and Verification

[7] Sawant (2020) Signature Verification using Deep Learning

[8] Hafemann, Sabourin, Oliveira (2017) Learning Features for Offline Handwritten Signature Verification

using Deep Convolutional Neural Networks

[9] Kao, H.-H., & Wen, C.-Y. (2020). An Offline Signature Verification and Forgery Detection Method Based on a Single Known Sample and an Explainable Deep Learning Approach.

[10] Alajrami, E., Ashqar, B. A., Abu-Nasser, B. S., Khalil, A. J., Musleh, M. M., Barhoom, A. M., & Abu-Naser, S. S. (2020). Handwritten signature verification using deep learning