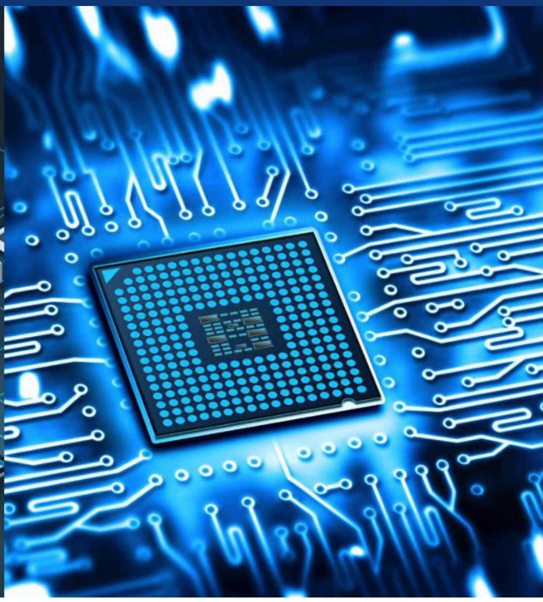
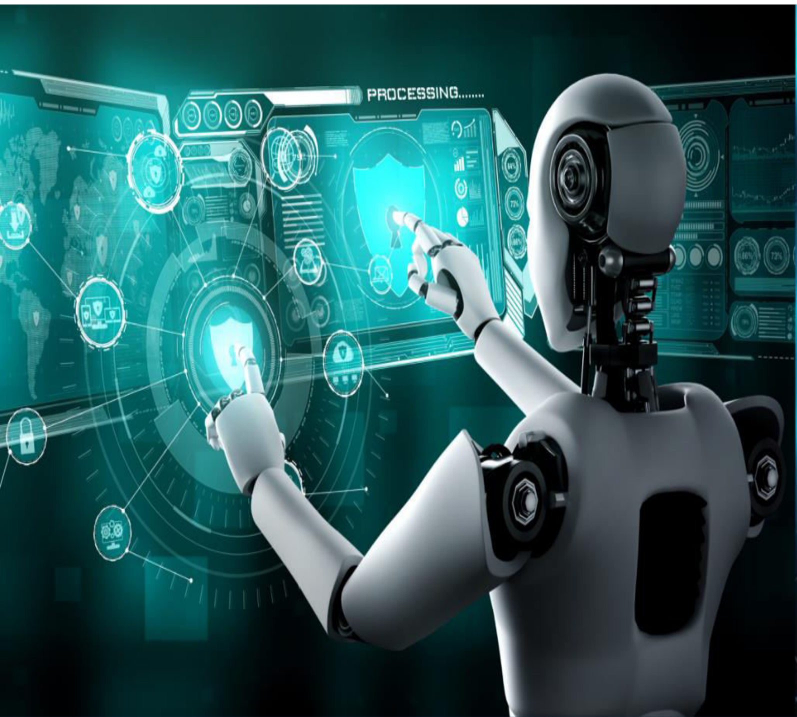
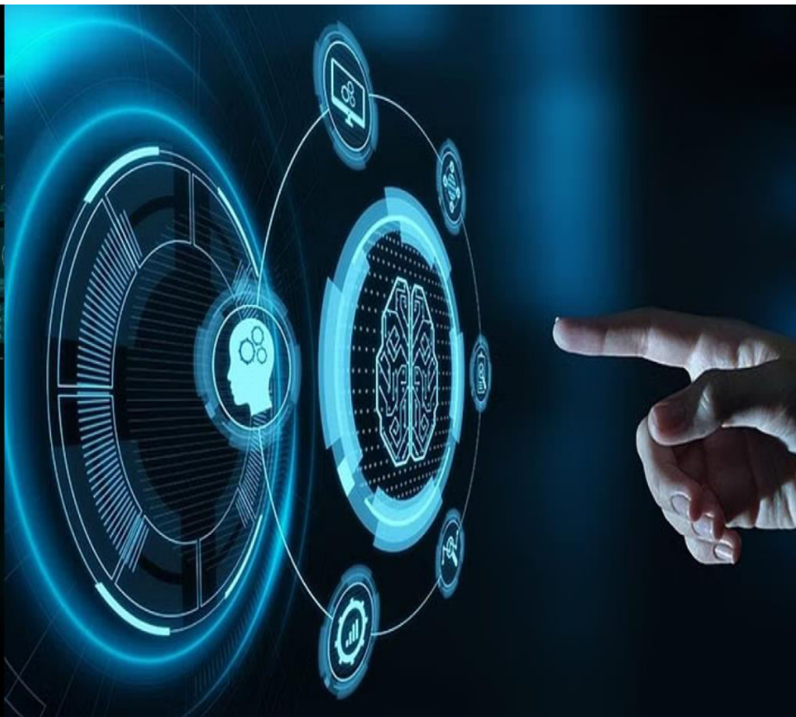
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.625

Volume 13, Issue 1, January 2025

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI:10.15680/IJIRCCE.2025.1301031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Signature Authenticity Detection using Enhanced Neural Networks and Visual Feature Analysis

Dr. M.V. Vijayakumar[1], Aman Kumar[2], Rajneesh Ravi Raj[3], Abhinav Kumar[4], Sahil Khursheed[5]

Professor, Department of Information Science & Engineering, Dr. Ambedkar Institute of Technology, Bangalore, Karnataka, India[1]

Undergraduate Students, Department of Information Science & Engineering, Dr. Ambedkar Institute of Technology, Bangalore, Karnataka, India[1,2,3,4]

ABSTRACT: In offline signature verification, the loss of dynamic signature information makes it difficult to design robust feature extractors capable of distinguishing genuine signatures from forgeries. To address these challenges, we propose a novel framework leveraging a Siamese neural network to learn discriminative features for signature verification. The proposed method integrates geometric and directional features into the learning process, capturing critical structural and spatial patterns that differentiate genuine signatures from forgeries. The Siamese architecture ensures that the network effectively compares pairs of signatures and learns their similarity in a Writer-Independent manner.

Additionally, we incorporate knowledge of skilled forgeries from a subset of users during training, enabling the network to identify subtle differences between genuine signatures and imitations. Furthermore, the learned features demonstrated strong generalization capabilities, surpassing state-of-the-art performance on other datasets without requiring fine-tuning for specific dataset.

KEYWORDS: Signature Verification, Siamese Neural Networks, Geometric Features, Directional Features, Deep Learning

I. INTRODUCTION

Handwritten signatures have long been a trusted method of personal authentication, widely used in legal, financial, and social contexts. Unlike other biometric systems such as fingerprints or iris scans, signatures are culturally ingrained and easily integrated into day-to-day transactions, making them a practical and widely accepted form of identity verification. However, the simplicity and accessibility of signatures also make them susceptible to forgery, posing significant challenges in ensuring their authenticity. Among the various types of forgeries, skilled forgeries—where an imitator deliberately mimics the handwriting style of an individual—are particularly challenging to detect and can lead to severe security vulnerabilities.

Signature verification methods are classified into online and offline systems. Online systems capture dynamic information when a person sign and it helps capturing the informations including pen pressure, stroke velocity, acceleration, and the sequence of strokes.

It leverages valuable time-based data to analyze how individuals create their signatures. Online techniques capture this dynamic information, such as the speed and pressure of pen strokes, using specialized hardware like digital pens or signature pads. However, this requirement limits their use to situations where such devices are readily available.

On the other hand, offline signature verification relies on static images of signatures, usually scanned from physical documents. This makes offline methods more accessible and practical for various scenarios, such as verifying signatures on checks, contracts, or historical records. However, these systems face challenges because they lack the dynamic details that online methods can provide, making it harder to detect subtle differences between genuine and forged signatures.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A critical factor influencing the performance of offline signature verification systems is the methodology used for feature extraction and classification. Traditionally, offline systems have relied on handcrafted features, such as texture, shape, contour, and edge orientation, to represent signatures. These features, though effective to some extent, often struggle to generalize across diverse writing styles and fail to capture subtle differences between genuine signatures and skilled forgeries. Moreover, offline signature verification systems are further categorized into writer- dependent (WD) and writer-independent (WI) approaches. WD systems are tailored to individual users, offering high accuracy by focusing on the specific characteristics of a user's signature. However, they require a large number of samples per user, making them impractical for large-scale applications. On the other hand, WI systems aim to verify signatures across a wide range of users without prior knowledge of individual characteristics. While more scalable, WI systems face greater challenges in achieving high accuracy due to the inherent variability in writing styles across individuals.

In recent years, advancements in machine learning, particularly the adoption of deep learning techniques, have revolutionized the field of offline signature verification. Convolutional Neural Networks (CNNs), in particular, have demonstrated remarkable success in extracting high-level features from images and have been widely applied to tasks such as handwriting recognition, word spotting, and document analysis. By learning hierarchical representations directly from data, CNNs eliminate the need for manual feature engineering and have achieved state-of-the-art performance in various pattern recognition tasks. Despite these advancements, the application of CNNs to offline signature verification still faces challenges, especially in addressing the variability of signatures in WI systems and distinguishing genuine signatures from skilled forgeries.

To address these challenges, we propose a novel offline signature verification framework based on a Siamese neural network architecture.

Unlike traditional CNNs, which focus on individual inputs, Siamese networks are designed to process pairs of inputs and learn a similarity metric that This makes Siamese networks particularly effective for verification tasks, where the goal is to determine whether two signatures belong to the same individual. By learning a similarity the Siamese network can effectively compare genuine signatures with forgeries, even in a WI setting.

To improve the performance of our Siamese network, we incorporated geometric and directional features into its learning process. Geometric features focus on the structure of the signature, like its shape, size, and alignment, while directional features analyze the orientation and flow of the pen strokes. These features complement each other, enabling the model to detect subtle differences between authentic signatures and forgeries. By blending the precision of handcrafted features with the power of deep learning, our approach offers a strong and versatile solution for offline signature verification.

Fig. 1. Some genuine signature samples (first column) and skilled forgeries (second column) .We evaluated the framework on ICDAR datasets. The result demonstrates the effectiveness of the Siamese network in learning discriminative features and achieving state-of-the-art performance in terms of Equal Error Rate (EER). Additionally, the integration of geometric and directional features ensures the model's adaptability to diverse signature styles, enabling strong generalization across datasets without requiring dataset-specific fine-tuning.

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI:10.15680/IJIRCCE.2025.1301031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Our approach tackles the main shortcomings of existing methods by bringing together the strengths of Siamese neural networks, geometric features, and directional features in a single, cohesive framework. This results in a scalable, reliable, and practical solution for offline signature verification, making it ideal for various applications, including document authentication and fraud prevention.

II. RELATED WORK

The study of Offline Signature Verification (OSV) and representation learning has been an area of significant research, particularly in addressing challenges related to security and identity verification. This section reviews the advancements in OSV and representation learning methods, summarizing relevant literature in these domains. Research into automatic OSV dates back to the 1970s and has evolved significantly over the decades. The fundamental objective in OSV is to distinguish genuine signatures from forgeries. Forgeries are typically categorized into Random Forgeries, where an imposter uses their own signature to impersonate someone else, and Skilled Forgeries, where the imposter carefully mimics the target's signature. The latter presents a more challenging problem due to the forger's effort in replicating nuances of the genuine signature.

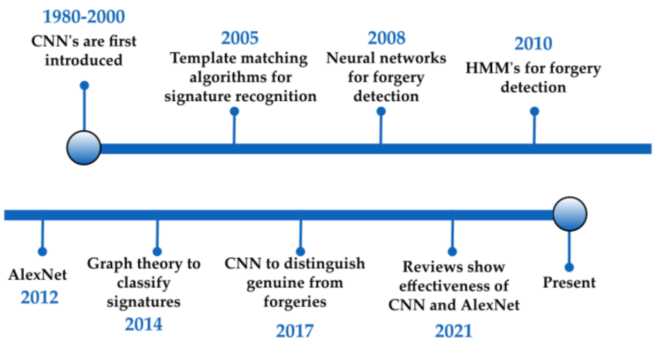


Fig. 2. Historical timeline of research of the problem of signature recognition.

Miguel A. Ferrer et al. [1] propose the use of a template matching algorithm based on the fact that the envelope of a signature contains features that can uniquely identify the signature. To compute the envelope of a signature, it is proposed to first apply the morphological dilation operation to ensure that the outline of the signature is unified and the variability of the signature is reduced. Then, a filling operation is used to simplify the process of extracting the outline. After those operations, the Euclidean distance can be used as a measurement function between the envelopes of the signatures.

Hafemann et al. [2] propose a feature extraction approach that uses Convolutional Neural Networks (CNN) to distinguish between genuine signatures and forgeries. The proposed CNN architecture includes five convolutional layers, three max-pooling layers, and four fully connected layers. Batch normalization is suggested as a crucial step in the implementation of the model. After the CNN extraction, the authors propose a transfer learning approach to use the extracted features to train a support vector machine (SVM) with the radial basis function (RBF) kernel.

Other authors have proposed the use of traditional artificial neural networks to classify signatures. Al-Shoshan et al. [3] propose the use of Fourier descriptors and the contour of the signature as input to a multilayer perceptron (MLP). Ali Karouni et al. [4] use simple geometric features of the signature such as area, centroid, eccentricity, kurtosis, and skewness to train an artificial neural network (ANN).

Several approaches have been proposed in the literature using concepts from graph theory. Tomislav Fotak et al. [5] propose the use of graph connectivity and the existence of Eulerian and Hamiltonian paths as features to classify a signature, and also propose the computation of the minimum spanning tree of the signature to build the graph of the signature, using the concept of avoiding sets as a condition to measure the distance and similarity between signatures. That is a combinatorial necessary and sufficient condition for cluster consensus [6]. The method achieves 94.25% accuracy.

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI:10.15680/IJIRCCE.2025.1301031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Abhay Bansal et al. [7] also propose the use of graph theory concepts to compare signatures by first extracting the critical points on the signature contour and using a graph matching algorithm to compute the similarities between signatures.

The use of Hidden Markov Models (HMM) has been widely popular in the literature and is proving to be a good statistical model for solving the signature recognition problem. An example of this is the work of J. Coetzer et al. [8], who propose the use of discrete Radon transforms and a Hidden Markov Model with ring topology trained with the Viterbi algorithm to detect forgeries. Other authors [9,10] have proposed the use of Hidden Markov Models, making them a well-known method in the field of signature recognition.

In one particular study, an MLP network was used to classify the features, in the last layer of a CNN model used to extract the signature features. Similar to the present study, the final result is binary, but it allows to check for forgery. The results of the model showed high accuracy [11].

III. METHODOLOGY

3.1. Workflow of the Method

The method followed were based on the following steps :-

1. Prepare datasets of the signature extracted from ICDAR 2011 Signature Dataset.
2. Develop a deep classifier (CNN) to classify signatures as legitimate or not.
3. Visualization of the final inference and the percentage of the similarity.

The following figure shows a block diagram highlighting the steps of the project.

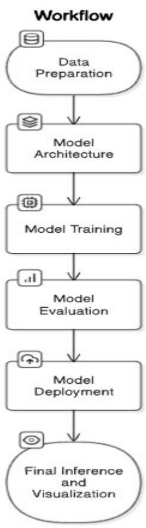


Fig.3. Block diagram showing the steps of the method followed.

1.2 Signature Data preparation

Data preparation is the first step in building a machine learning model. In this project we are working with signature image datasets like the ICDAR 2011 dataset . This process involves preparing the data so it is clean, consistent throughout the dataset, and is ready for model training and testing. The cleaning of datasets includes image preprocessing, feature extraction, and dataset creation, all of this helps in setting up a solid foundation for model learning.

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI:10.15680/IJIRCCE.2025.1301031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Image preprocessing begins by resizing all images to a uniform size, ensuring that they are neither too large nor too small for the model to analyze patterns consistently. Next, the images are converted to grayscale, stripping away color details while preserving key features like edges and textures. Finally, the pixel values are normalized—adjusted to a standard range, usually between 0 and 1. This step improves the model's stability and speeds up the learning process.

The next step is feature extraction, where important characteristics are identified from the images to help distinguish genuine signatures from forgeries. Geometric features examine shapes and contours, directional features focus on the flow and orientation of edges, and regional features highlight specific areas to capture unique patterns or textures.

Finally, the dataset is created by pairing genuine and forged images, labeling them accordingly, and then splitting the data into training and testing sets. These labeled samples are saved in files like train_data.csv and test_data.csv, ensuring the dataset is well-organized and ready for model training and evaluation.

1.3 Model Architecture

The model used in this research is a Siamese network, designed to compare two inputs and measure how similar they are. It processes two images side by side, learning to identify the relationships or differences between them. In this study, the network is used to determine whether pairs of images are genuine signatures or forgeries.

The network has several key components. At its core are convolutional neural network (CNN) layers, which extract essential features from the images. These layers are adept at detecting patterns like edges, textures, and shapes—critical elements for distinguishing authentic signatures from forgeries.After extracting features, the network uses fully connected layers to process these features and create representations that can be used for comparison.

A unique feature of the Siamese network is that it uses shared weights for both image streams. Both input images pass through the same CNN layers, which use identical parameters to ensure consistent feature extraction and avoid any bias between the two images. In the final step, the network compares the extracted features using a distance function, such as Euclidean or cosine distance, to calculate a similarity score. This score indicates how closely the two images match.

1.4 Model Training

The offline signature verification model is trained using a Siamese neural network, an architecture designed for comparing pairs of inputs. This makes it ideal for signature verification, where the task is to assess the similarity between two images. The network consists of two identical branches with shared weights, ensuring that both images in a pair are processed in the same way for consistent feature extraction.

During training, pairs of signature images are fed into the model. Each branch processes one image, extracting a feature embedding that represents the signature's unique traits. These embeddings are then compared using a Euclidean distance metric, which calculates the similarity between the two. Based on this similarity score, the model classifies the pair as either a genuine match or a forgery.

To train the model, the Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) is employed. This loss function evaluates the difference between the predicted similarity score and the actual label (1 for genuine pairs and 0 for forged pairs). By penalizing incorrect predictions, the loss function guides the network to refine its parameters, improving its ability to correctly classify signature pairs. This step is crucial in teaching the model to distinguish between subtle variations in genuine and forged signatures.

The Adam optimizer, a popular choice for deep learning models, is used for optimization. It adjusts the model’s weights and biases during training based on the computed gradients, leveraging its adaptive learning rate for efficient convergence. Training is conducted in mini-batches, each containing 32 pairs of signature images. This batching approach ensures that the model processes diverse examples in each epoch, improving its generalization capabilities. The dataset is also shuffled before each epoch to prevent overfitting to specific patterns.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. EXPERIMENT AND RESULT

When evaluating an offline signature verification model, it's crucial to ensure that it can reliably and accurately tell genuine signatures apart from forgeries. To achieve this, we asses the model using a variety of techniques, which not only highlight its strengths but also reveal areas where it could improve.

A key part of this evaluation is the classification report. Classification report is performance measure which handles metrics like precision, recall, F1-score, and accuracy. Precision tells us how often the signatures identified as genuine are actually authentic. Recall, on the other hand, measures how well the model catches all the genuine signatures. The F1-score balances these two metrics, making it especially handy when dealing with imbalanced datasets. Accuracy, as the name suggests, gives a straightforward percentage of how often the model gets things right overall.

Another useful tool is the confusion matrix. It breaks down the output prediction of the model into four different categories which are true positives (correctly identified genuine signatures), true negatives (correctly flagged forgeries), false positives (forgeries wrongly marked as genuine), and false negatives (genuine signatures mistaken for forgeries). This breakdown paints a clearer picture of where the model might struggle, such as detecting small but meaningful differences between similar signatures.

Since the model generates similarity scores rather than simple yes-or-no answers, it uses a threshold to decide how close is "close enough." Adjusting this threshold lets us find the right balance between sensitivity (catching all genuine signatures) and specificity (avoiding false alarms with forgeries). Fine-tuning this balance helps the model perform better in real-world scenarios, where such decisions carry real consequences.

By the combination of these evaluation techniques, we get a detailed view of the model's performance. This comprehensive approach ensures the model is not only accurate but also reliable enough to handle the nuanced challenges of offline signature verification.

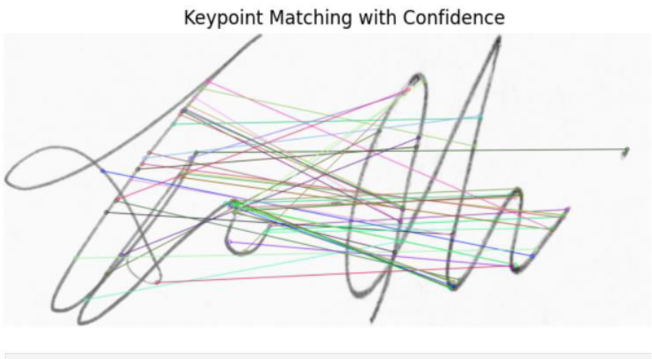


Fig.4. Feature Matching with Confidence

Fig.4. refers to the process of identifying and comparing distinctive features (keypoints) in signature images to assess their similarity. In the context of the given model, the Siamese neural network extracts feature embeddings for each signature image, capturing unique patterns and characteristics. The Euclidean distance between these embeddings is calculated to quantify their similarity.

The confidence in matching is reflected in the similarity score output by the model. A low score indicates a high confidence match, suggesting the signatures are likely from the same individual, while a high score indicates a mismatch with low confidence in their similarity. By applying a carefully chosen threshold, the model translates these scores into binary decisions (genuine or forged).

Confidence in keypoint matching improves with robust feature extraction and diverse training data, ensuring the model reliably distinguishes subtle differences across signature pairs.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

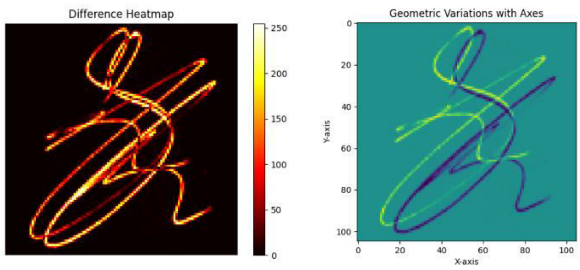


Fig.5. Difference Heatmap(5.1.) and Geometric Variations with Axes(5.2.)

Fig 5.1. visualizes the pixel-wise or feature-level differences between two signature images. This technique highlights regions of significant dissimilarity, aiding in the identification of forged characteristics. In the given model, a heatmap could be generated by subtracting the feature maps extracted by the Siamese network branches. Bright regions in the heatmap indicate areas where the two signatures differ significantly, such as altered strokes or missing details, while darker regions represent similar features. Heatmaps are often overlaid on the original images to provide interpretable visual feedback, enhancing the understanding of where mismatches occur.

Fig 5.2. refer to differences in structural aspects of the signatures, such as alignment, rotation, scaling, or aspect ratio. These variations are analyzed using axes to quantify spatial relationships. For instance:

- **X-Axis**: Represents horizontal alignment, showing shifts or discrepancies in the stroke layout.
- **Y-Axis**: Captures vertical alignment and height variations, indicating inconsistencies in letter formations or overall proportions.

By plotting these variations along the axes, the model can identify geometric distortions that are typical in forgeries, such as unnatural scaling or misplaced elements. These methods complement feature-level analysis by providing a geometric context for the differences observed in the signature pairs.

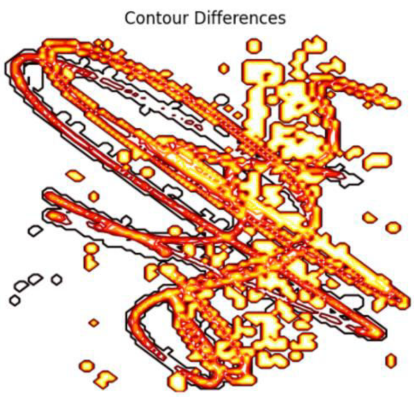


Fig.6. Contour Difference

Analyzing contour differences involves studying the shapes and outlines of a signature to determine its authenticity. This approach focuses on both the outer and inner contours, which are the curves and edges that define the overall structure of a signature. By carefully examining the directional flow of the strokes, the shape of the curves, and the lengths of the contours, the system identifies subtle variations between a questioned signature and reference signatures.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

These contour features help reveal inconsistencies that may indicate forgery, such as unnatural stroke directions, altered shapes, or discrepancies in structural details. Since contours capture the essence of a signature's unique form, they are difficult to replicate precisely, making this method particularly effective in detecting forgeries.

Research has demonstrated that contour-based analysis significantly enhances the accuracy of signature verification systems. By highlighting differences in the shape and flow of signatures, this method provides valuable insights that aid in distinguishing genuine signatures from forgeries in a reliable and interpretable way.

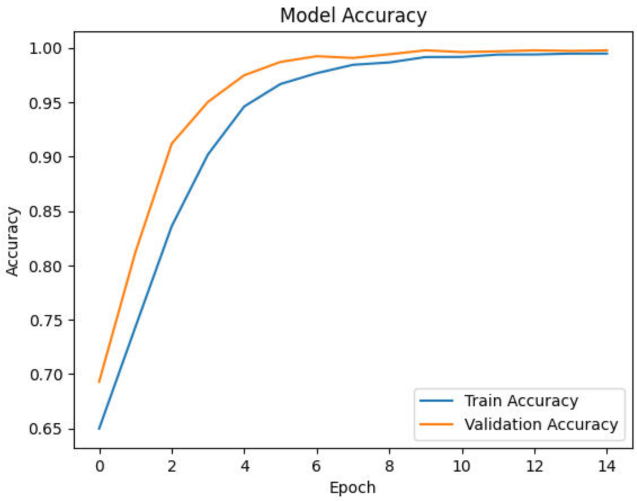


Fig.7. Model Accuracy

Fig.7. shows the model's accuracy over training epochs providing a clear picture of how well the system is learning. The x-axis represents the number of training epochs, while the y-axis shows the accuracy levels. Typically, there are two lines on the graph—one for training accuracy and another for validation accuracy.

At the start of training, both accuracies are usually low because the model is just beginning to learn how to differentiate between genuine and forged signatures. As training progresses, the model improves, and both accuracy lines rise, reflecting its growing ability to correctly identify signatures.

If the training accuracy becomes much higher than the validation accuracy, it may indicate overfitting. This means the model has learned the training data very well but struggles with unseen data. On the other hand, if both lines stay close together and reach high accuracy, it shows that the model is learning effectively and can generalize well to new samples.

A graph with closely aligned, high accuracy lines suggests the model is robust and reliable, capable of accurately verifying signatures across different scenarios. This makes it a valuable tool for practical signature verification tasks.

© 2025 IJIRCCE | Volume 13, Issue 1, January 2025| DOI:10.15680/IJIRCCE.2025.1301031

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

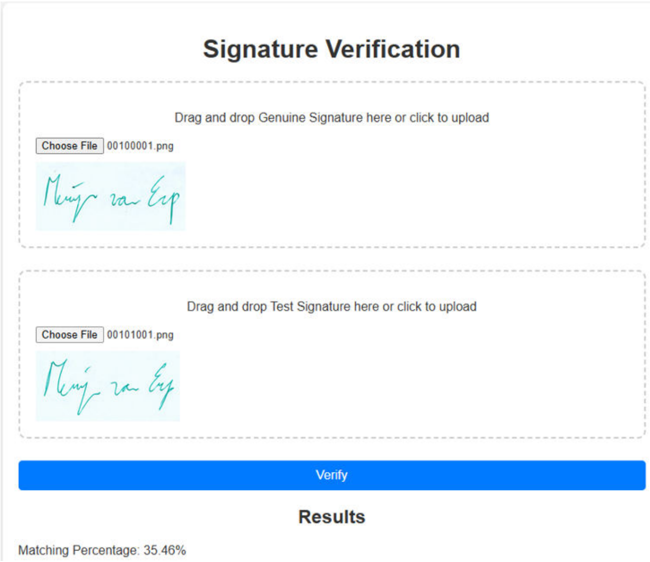


Fig.8. Front End Website

Fig.8. depicts a signature verification system that combines a Flask-based backend with a web-based frontend interface. Flask is a lightweight and versatile Python web framework, powers the backend by acting as a bridge between the user interface and the underlying machine learning model. The system revolves around a robust Model Inference Endpoint within the Flask backend, which processes user-submitted signature images. Users upload two images—a "Genuine Signature" and a "Test Signature"—via the web interface. These images are then passed through a Data Preprocessing Pipeline, where they are resized, normalized, and transformed to align with the trained model's input requirements.

At the core of our backend we use the Signature Verification Logic, employing a pre-trained machine learning model which compares the two signatures. The model extracts the key features, such as contours in the signature, key points, and geometric variations computing a similarity score. This score is translated into a matching percentage, which quantifies how closely the test signature resembles the genuine one. For an instance, in one example, the system calculated a match percentage of 35.46%, indicating low similarity and suggesting that the test signature does not match the genuine signature. This output is then formatted as a JSON response and sent back to the frontend.

The frontend interface is designed keeping user convenience in mind, allowing users to upload signature images through an intuitive interface and displays the verification results, including the matching percentage, after processing. The interface also includes mechanisms to handle errors gracefully, ensuring a smooth user experience.

This integration of Flask and a web-based frontend exemplifies how machine learning models can be deployed effectively in real-world applications. Flask's flexibility supports complex backend operations like preprocessing and model inference, while the frontend ensures seamless communication of results to the user. This system highlights the potential of Flask for implementing practical solutions like offline signature verification.

V. CONCLUSION

In this research we demonstrated an offline signature verification system's potential of combining machine learning and web technologies for real-world applications.

- Flask-based backend and a web-based frontend, the system provides a seamless and user-friendly solution for verifying signatures.
- At its core, the model relies on a Siamese neural network, which compares pairs of signature images to compute similarity scores and determine whether the signatures match.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- The model analyzes key features, such as contours, keypoints, and geometric variations, to make its decisions.

- The data preprocessing pipeline standardizes input images by resizing and normalizing them, ensuring they are compatible with the trained model. However, the system's limited use of data augmentation techniques, such as distortions or rotations, restricts its ability to handle variations in handwriting styles or image noise effectively.

- The Flask-based backend facilitates efficient communication between the frontend and the verification model, managing tasks such as preprocessing, inference, and result generation. The frontend interface enhances usability by allowing users to upload images easily and view real-time results, such as matching percentages. For example, a 35.46% match score in one instance indicated low similarity, effectively signaling a potential mismatch.

REFERENCES

1. Ferrer, M.A.; Alonso, J.B.; Travieso, C.M. Offline geometric parameters for automatic signature verification using fixed-point arithmetic. IEEE Trans. Pattern Anal. Mach. Intell. 2005, 27, 993–997. [CrossRef]
2. Hafemann, L.G.; Sabourin, R.; Oliveira, L.S. Learning features for offline handwritten signature verification using deep convolutional neural networks. Pattern Recognit. 2017, 70, 163–176. [CrossRef]
3. Al-Shoshan, A.I. Handwritten signature verification using image invariants and dynamic features. In Proceedings of the International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06), Sydney, Australia, 26–28 July 2006; pp. 173–176.
4. Karouni, A.; Daya, B.; Bahlak, S. Offline signature recognition using neural networks approach. Procedia Comput. Sci. 2011, 3, 155–161.
5. Fotak, T.; Bacˇa, M.; Koruga, P. Handwritten signature identification using basic concepts of graph theory. WSEAS Trans. Signal Process. 2011, 7, 117–129.
6. Shang, Y. A combinatorial necessary and sufficient condition for cluster consensus. Neurocomputing 2016, 216, 611–616. [CrossRef]
7. Bansal, A.; Nemmikanti, P.; Kumar, P. Offline signature verification using critical region matching. In Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia, Sanya, China, 13–15 December 2008; Volume 3, pp. 115–120.
8. Coetzer, J.; Herbst, B.M.; du Preez, J.A. Offline signature verification using the discrete radon transform and a hidden Markov model. EURASIP J. Adv. Signal Process. 2004, 2004, 1–13. [CrossRef]
9. Justino, E.J.; El Yacoubi, A.; Bortolozzi, F.; Sabourin, R. An off-line signature verification system using HMM and graphometric features. In Proceedings of the 4th International Workshop on Document Analysis Systems, Boston, MA, USA, 9–11 June 2010; pp. 211–222.
10. Daramola, S.A.; Ibiyemi, T.S. Offline signature recognition using hidden markov model (HMM). Int. J. Comput. Appl. 2010, 10, 17–22. [CrossRef]
11. Khalajzadeh, H.; Mansouri, M.; Teshnehlab, M. Persian signature verification using convolutional neural networks. Int. J. Eng. Res. Technol. 2012, 1, 7–12

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com