Feature Learning for Handwritten Signature Verification Using Machine Learning

Ch.Srujana[1], K.Vaishanvi, M.Varun[1], G.Sai Prakyath[1]

1. M.V.S.R Engineering College , Hyderabad, Telangana, **India-501510**
Corresponding Author :Email: srujana13ch@gmail.com, 245117737013@mvsrec.edu.in, 245117737040@mvsrec.edu.in, 245121737045@mvsrec.edu.in

Abstract:

Signature forgery has become increasingly common, creating a strong need for reliable and automated signature recognition and verification systems. Since signatures are widely used for personal authentication and carry individual behavioural characteristics, accurate verification is essential to prevent unauthorized access and fraud. This paper presents an offline signature verification approach that treats handwritten signatures as images and applies computer vision and neural network–based techniques for recognition. A dataset of genuine signatures is used for training, where discriminative features are extracted and stored in a feature database. During testing, features from an input signature are computed and matched against the stored template features to obtain a similarity score. The system classifies a signature as genuine when the matching score exceeds a predefined threshold, set to 70% in this work. Experimental observations indicate that the proposed method can effectively distinguish genuine signatures from mismatches, demonstrating its usefulness for practical signature authentication applications.

Keywords : Signature Verification, Signature Recognition, Offline Handwritten Signatures, Forgery Detection, Machine Learning, Computer Vision.

Introduction:

Machine learning (ML) is a rapidly evolving subfield of artificial intelligence (AI) that focuses on the development of computer algorithms capable of improving automatically through experience and the use of data **[1], [2].** These algorithms construct predictive models based on training data—often referred to as learning from examples to make decisions or predictions without the need for explicit programming **[3].** ML methods are widely applied in diverse domains such as medicine, email filtering, computer vision, and natural language processing, where traditional algorithmic approaches often fail to deliver efficient or scalable solutions **[1], [4].**In machine learning, algorithms are typically categorized into supervised **and** unsupervised learning paradigms **[5**]. Supervised learning involves training models on labeled datasets, where both the input and output variables are known, enabling the model to learn the mapping function Y=f(X)Y = f(X)Y=f(X) **[6].** This approach underpins applications such as regression and classification, utilizing algorithms like linear regression, logistic regression, support vector machines (SVM), and decision trees **[7], [8].** In contrast, unsupervised learning operates on unlabelled data, allowing models to identify hidden patterns, clusters, and associations without predefined categories **[9].** Common examples include clustering algorithms and association rule mining techniques. Regression analysis remains a fundamental technique within supervised learning, aimed at modelling relationships between dependent and independent variables to predict continuous outcomes **[6], [7].** The linear regression model assumes a linear dependency between predictors and outcomes, while non-linear regression techniques—such as polynomial or spline regression—capture more complex relationships **[8].** Advanced regularization methods such as Lasso Regression (Least Absolute Shrinkage and Selection Operator) enhance model performance by penalizing irrelevant features and reducing over fitting **[10].**Furthermore, neural networks, particularly Convolutional Neural Networks (CNNs)**,** have revolutionized pattern recognition and computer vision tasks by automatically learning hierarchical feature representations from raw pixel data **[4], [11].** Unlike traditional methods that rely on handcrafted features like texture and shape, CNNs can autonomously extract discriminative patterns, enabling accurate image classification and recognition. This makes ML an effective framework for tasks such as signature recognition and verification**,** where human behavioural patterns—encoded in handwritten signatures—can be analysed and authenticated computationally.

Literature Survey:

Handwritten signatures remain a widely accepted biometric trait for authentication in banking and administrative workflows. Signature verification systems are generally categorized into online and offline methods. Online systems use dynamic information such as pen pressure and velocity, while offline signature verification operates on scanned images only, making it more challenging due to loss of temporal cues, background noise, and high intra-writer variation. Early offline approaches commonly follow a pipeline of pre-processing, handcrafted feature extraction, classification/matching. Survey work by Impedovo and Pirlo reported that classical systems rely on geometric and statistical descriptors (shape measures, stroke distribution, contour cues), and texture/directional features, followed by classifiers such as MLPs, SVMs, and distance-based verification, often evaluated using FAR/FRR/EER metrics. Public datasets have played a key role in standardized evaluation. The **GPDS-960** corpus is widely used and contains a large number of users with genuine signatures and forgeries, supporting both writer-dependent and writer-independent evaluation settings. The **CEDAR** signature resources and related benchmark setups have also been used extensively for offline verification experiments and feature-learning studies.

Methodology:

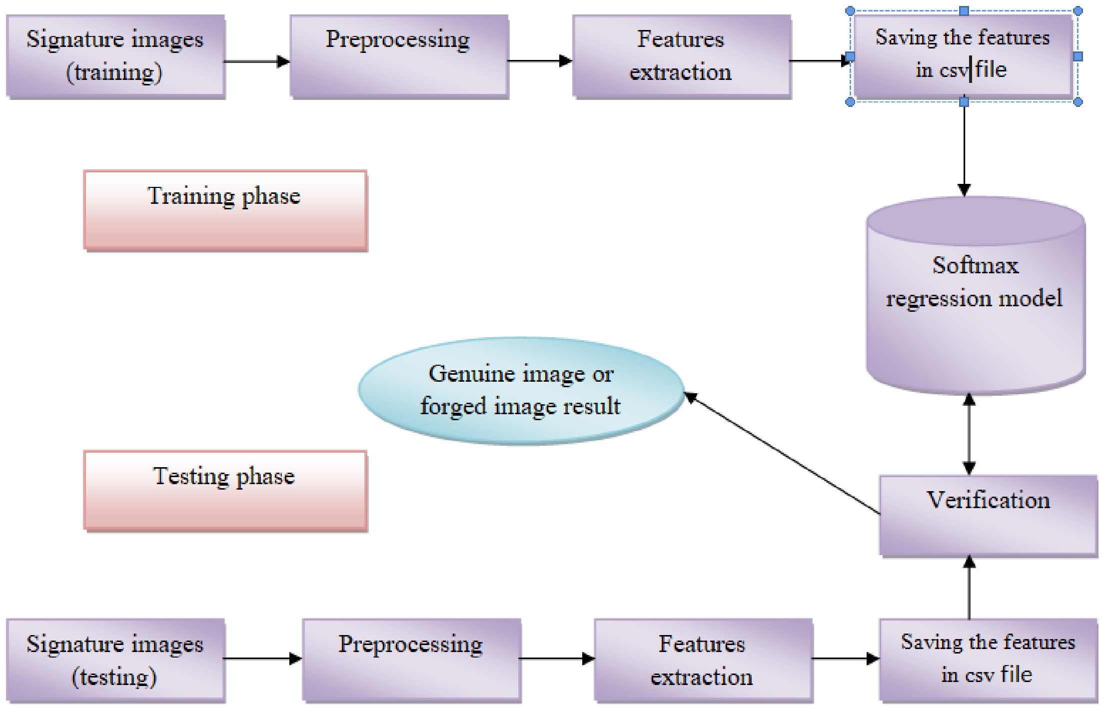

Fig 4: Proposed model

The end-to-end workflow of the proposed system is shown in **Fig. 4** In the training phase, signature images are pre-processed and transformed into feature vectors. These features are stored (CSV format) for reproducibility and faster experimentation. A Softmax-based classifier is trained using the extracted features. During testing, the same pre-processing and feature extraction steps are applied, followed by verification, producing a genuine/forged decision.

Feature Extraction and Storage

Each signature image is converted into a discriminative feature representation. The extracted feature vector for each sample is stored in a CSV file, forming a structured feature database that supports:

1. Reproducible experiments,
2. Efficient training without repeated image processing, and

3. Streamlined comparison during testing.

SOFTMAX REGRESSION CLASSIFIER

Softmax regression (multinomial logistic regression) converts network outputs (logits) into a normalized probability distribution over classes. For an input feature vector xxx, the Softmax probability for class iii is

P(y=i∣x)=ezi∑j=1Kezj,where z=Wx+bP(y=i \mid x)=\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \quad \text{where } z = Wx + bP(y=i∣x)=∑j=1Kezjezi,where z=Wx+b

where WWW is the weight matrix and bbb is the bias vector. The output lies in [0,1][0,1][0,1], and the sum of probabilities across all classes equals 1. Softmax is especially suitable for classification because it produces interpretable confidence values and supports training via cross-entropy minimization.

Phase I: Training

1. **Initialize** paths for genuine and forged training folders.
2. **Load** all training images Ii∈DtrainI_i \in \mathcal{D}_{train}Ii∈Dtrain.
3. **For each** training image IiI_iIi:
   3.1. **Preprocess** IiI_iIi: convert to grayscale, binarize (e.g., Otsu threshold), remove noise, normalize size, and center the signature.
   3.2. **Extract Features** using CNN feature extractor:

   fi=Φ(Ii)\mathbf{f}_i = \Phi(I_i)fi=Φ(Ii)

   3.3. **Store** (fi,yi)(\mathbf{f}_i, y_i)(fi,yi) into a CSV feature file FtrainF_{train}Ftrain.

4. **One-hot encode** labels yiy_iyi to [1,0][1,0][1,0] (genuine) and [0,1][0,1][0,1] (forged).
5. **Train** a Softmax regression classifier on FtrainF_{train}Ftrain to learn parameters (W,b)(W,b)(W,b).
6. **Save** the trained model parameters and pre-processing configuration.

Phase II: Testing (Verification and Decision)

For each query image Iq∈DtestI_q \in \mathcal{D}_{test}Iq∈Detest:
7.1. **Pre-process** IqI_qIq using the same pre-processing steps applied during training.
7.2. **Extract Features**:

fq=Φ(Iq)\mathbf{f}_q = \Phi(I_q)fq=Φ(Iq)

7.3. **Compute class probabilities** using Softmax:

p=Softmax (Wfq+b)\mathbf {p} = \text {softmax}(W\mathbf{f}_q + b)p=softmax(Wfq+b)

where p=[pgenuine,pforged]\mathbf{p} = [p_{genuine}, p_{forged}]p=[pgenuine,pforged].
7.4. **Set matching score** M=pgenuineM = p_{genuine}M=pgenuine.
7.5. **Decision Rule:**

- If M≥τM \ge \tauM≥τ, classify as **Genuine**
- Else classify as **Forged**

8. **Return** predicted class and score MMM for each query.

Results:

This work presents the results of the Signature recognition and verification using Machine Learning..

When the filename is 001001_001.png the first three letters and the next three letter are same therefore the output should be Genuine Signature.

```
Enter person's id : 001
Enter path of signature image : 001001_001.png
signature of the person is 'Bharath'
Genuine Image

Out[42]: True
```

Fig 1. Output of Genuine Signature.

When the filename is 021001_001.png the first three letters and the next three letter are not same therefore the output should be Forged Signature.

```
Enter person's id : 001
Enter path of signature image : 021001_001.png
signature of the person is 'Bharath'
Forged Image

Out[43]: False
```

Fig 2. Output of the Forged Signature

Accuracy of our Model:

```
In [13]: runfile('D:/varun/major/sigrecogtf.py', wdir='D:/varun/major')
Reloaded modules: preprocessor
OpenCV version 4.5.1
0.955555
```

Fig 3. Accuracy of our Model

Conclusion:

This work presented an offline handwritten signature recognition and verification system using a machine learning pipeline based on image pre-processing, CNN-based feature extraction, and Softmax classification. A Kaggle dataset of approximately 4000 signature images, containing both genuine and forged samples from multiple individuals, was used to train and evaluate the model. The proposed workflow first pre-processes the signature images to reduce noise and normalize writing variations, then extracts discriminative features which are stored in a CSV-based feature database for efficient training and testing. During verification, features from a query signature are processed through the trained classifier to obtain a confidence score, and a decision is made using a fixed threshold of 70% to classify the signature as genuine or forged.

References :

[1] . IBM, "What is Machine Learning?," *IBM Think*, 2025. [Online]. Available: https://www.ibm.com/think/topics/machine-learning
[2] . G. Piatetsky, "Machine Learning: The Future of Data-Driven AI," *KDnuggets*, 2023.
[3] . T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
[4] . Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015.
[5]. S. Raschka, "An Overview of Machine Learning Algorithms," *arXiv preprint arXiv:2106.04764*, 2021.
[6]. J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, Springer, 2017.
[7] . C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
[8] . I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
[9].Wikipedia, "Unsupervised Learning," 2025. [Online]. Available: https://en.wikipedia.org/wiki/Unsupervised_learning
[10]. R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
[11]. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.