

Article

High-Performance Embedded System for Offline Signature Verification Problem Using Machine Learning

Umair Tariq ¹, Zonghai Hu ^{1,*}, Rokham Tariq ¹, Muhammad Shahid Iqbal ²  and Muhammad Sadiq ³¹ School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China² Department of Computer Science and Information Technology, Women University of AJ&K, Bagh 11100, Pakistan³ College of Software Engineering, Shenzhen Institute of Information Technology, Shenzhen 522646, China

* Correspondence: zhhu@bupt.edu.cn

Abstract: This paper proposes a high-performance embedded system for offline Urdu handwritten signature verification. Though many signature datasets are publicly available in languages such as English, Latin, Chinese, Persian, Arabic, Hindi, and Bengali, no Urdu handwritten datasets were available in the literature. So, in this work, an Urdu handwritten signature dataset is created. The proposed embedded system is then used to distinguish genuine and forged signatures based on various features, such as length, pattern, and edges. The system consists of five steps: data acquisition, pre-processing, feature extraction, signature registration, and signature verification. A majority voting (MV) algorithm is used for improved performance and accuracy of the proposed embedded system. In feature extraction, an improved sinusoidal signal multiplied by a Gaussian function at a specific frequency and orientation is used as a 2D Gabor filter. The proposed framework is tested and compared with existing handwritten signature verification methods. Our test results show accuracies of 66.8% for ensemble, 86.34% for k-nearest neighbor (KNN), 93.31% for support vector machine (SVM), and 95.05% for convolutional neural network (CNN). After applying the majority voting algorithm, the overall accuracy can be improved to 95.13%, with a false acceptance rate (FAR) of 0.2% and a false rejection rate (FRR) of 41.29% on private dataset. To test the generalization ability of the proposed model, we also test it on a public dataset of English handwritten signatures and achieve an overall accuracy of 97.46%.



Citation: Tariq, U.; Hu, Z.; Tariq, R.; Iqbal, M.S.; Sadiq, M. High-Performance Embedded System for Offline Signature Verification Problem Using Machine Learning. *Electronics* **2023**, *12*, 1243. <https://doi.org/10.3390/electronics12051243>

Academic Editors: Yohan Ko, George K. Adam and Gwanggil Jeon

Received: 18 January 2023

Revised: 15 February 2023

Accepted: 18 February 2023

Published: 4 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: Urdu handwritten signature verification (UHSV); false acceptance rate (FAR); false rejection rate (FRR); majority voting (MV); pattern recognition (PR); convolutional neural network (CNN); k-nearest neighbor (KNN)

1. Introduction

In machine learning, edge computing is becoming increasingly important, especially when potent devices, such as modern hybrid embedded systems, have shown a fantastic aptitude for high performance in their demanding processing requirements [1]. Offline signature verification is one of the state-of-the-art biometric recognizable proof methodologies. Many systems of falsifiers and verifiers have emerged in recent years [2]. Signature verification is still tough because no standard database is available for forgeries.

Though many script signature data sets, such as English, Latin, Chinese, Persian, Arabic, Hindi, and Bengali, are publicly available, no Urdu handwritten datasets are currently available in the literature. The purpose of this work is to solve the Urdu signature verification (USV) problem by utilizing cutting-edge machine learning methods. In the proposed framework, a majority voting (MV) algorithm is used to enhance accuracy. Majority voting needs an odd number of classifiers to vote accurately. The channel is split into two distinct parts—real and forged. The proposed model is evaluated using the public data set available in [3] and the results are compared with other methods. The FAR and

FRR equations are created using the MV algorithm. A 2D Gabor filter is improved as a sinusoidal signal modulated by a Gaussian function at a specific frequency and orientation. There are two directional components in the Gabor filter. A sinusoidal wave characterizes the Gabor filter's drive reaction, which is increased by a Gaussian function [4].

In this study, we consider Urdu, which is a mainstream South Asian dialect. "There are 250 million people who communicate in Urdu, around 130 million people live in Pakistan and India". Urdu has 37 basic characters with a couple of enlargements to key characters, which makes a total of 41. Persian has 32 characters, and the Arabic script contains 28 characters. The main objective of the offline handwritten signature verification system is to improve the accuracy and minimize the error rate of the signature verification system. In a real-time situation, only 30% of forgeries can be identified. This means that the remaining 70% of forgeries in offline signatures still need to be cleaned up [5]. This research is intended to propose an innovative, accurate, and efficient "offline signature verification system for Urdu handwritten signature". The USV will confine itself to unique and manufactured signatures by relying upon some exceptional parts, such as length, pattern, and instance.

The Kaggle dataset in [5] was used to compare and test for more public datasets. The Kaggle handwritten dataset contains the real and forged signatures of 30 people. Every person has five real signatures and five fake signatures from someone else. An image of an individual numbered 023's signature made by user 06 is NFI-00602023. Due to potential authentication in fields such as crime prevention and forensic identification, signature verification has been a study area of interest in recent years [6].

Recent advances in machine learning and artificial intelligence have enabled several intriguing insights into brain computation. Many scientists believe that deep neural networks will one day equip biological brains with theories of perception, cognition, and action [7]. According to [8], deep learning relies on the construction of multiple nonlinear functions modeling the intricate relationship between input characteristics and labels. While neural networks have been around for a while, improvements in the past several years have substantially enhanced their effectiveness in natural processing, computer vision, robotics, and decision-making [7,8].

A new hybrid model of two famous classifiers is presented in [9]. SVM and CNN are used for handwritten digit recognition. The SVM and CNN are integrated by replacing the last output layer of the CNN with the support vector machine. In the experimental methodology, they use the well-known MNIST handwritten dataset to evaluate the presented model's performance. The dataset contains 60,000 samples for the training set and 10,000 samples for the testing set. The KNN approach in [10] is used for offline signature verification, and they implement their proposed technique using a public data set. In [11], a strategy for directing a disconnected, manually written signature confirmation system is presented. In this paper, the creators apply a factual surface component. The conventional neural network is examined and used as an element. In [12], the author proposes an automated method for identifying the surface condition of concrete structures using vision-based techniques, such as deep convolutional neural networks (CNNs), transfer learning, and decision-level image fusion [12]. A comprehensive overview of recent deep-learning-based image colorization methods from several angles, such as color space, network structure, loss function, degree of automation, and application domains, is presented. Measures for gauging effectiveness are also covered [13]. Deep learning techniques are used to determine the strength of RC beams against torsional loads. Two-dimensional CNN models are established based on the collected data. When compared to other machine learning models, building codes, and empirical formulas for predicting the torsional strength of RC beams, the proposed 2D CNN with hyperparameter optimization shows marked improvement [14]. The handwritten verification dataset has been implemented in [14] to compare the accuracy of our proposed solutions. In the end, CNN is also included in the proposed solution.

In [15], the authors also use KNN for signature verification. For the training [16] and testing of their proposed methodology, they use MCYT-75 and GPDS public databases for their result. An offline Japanese signature verification has been proposed [16,17]. This paper proposes a transformed example matching strategy, free of stroke widths, and a suitable determination system. The trial results demonstrate that the proposed techniques enhance the confirmation execution and the exactness of the estimation under a 0.16 FRR. S. In [18], an Arabic signature oracle database for signature verification is presented with geometrical features for signature feature extraction, and in the verification phase, SVM is used for the results [19].

In [20], a system for verification and authentication is presented. This particular system differentiates between individuals and recognizes them based on their style. The second goal is to narrow the search to identify the gender in the database.

Biometric signatures based on handwriting have been the subject of debate in the scientific community. There has been a lot of work conducted on expanding the use of handwritten signature analysis and processing systems. A comprehensive literature review of offline handwritten signatures published over the past decade [21] zeroed in on cutting-edge techniques for authenticating signatures using deep learning.

A comprehensive study on ensemble-based classifiers in [22] is presented, and the main idea of this study was to review existing methodologies. This study can be used as a tutorial for new researchers who are working on ensemble-based systems. In the system, four blocks are introduced [23]. The key factors are accuracy, computational cost, scalability, and flexibility. The ensemble is highly diverse, and the choice of combination also grants better performance and accuracy of the system [24].

An approach is proposed in [25] for the recognition of objects from datasets of images. The presented approach is a combination of two SVM and KNN [26]. To identify and categorize potential IoT devices in the future, the DeepClassifier deep neural network has been proposed. Initial work proposes a likelihood-based approach and demonstrates that it is optimal in the asymptotic limit of the observations, despite its high computational complexity [27]. Though feature-based classification methods claim to be simple, designing efficient features can be challenging [28]. The proposed DeepClassifier is divided into two stages: the first processes a single observation; the second combines the results of multiple classifications [29].

For signature verification problems, we propose a novel hybrid machine learning system for testing handwritten signature verification. A new Urdu handwritten signature database is created, and a framework is proposed for data acquisition, pre-processing, feature extraction, and hybrid classification. It achieves high accuracy for signature verification problems.

This paper is organized into four sections. Section 1 includes the Introduction and a background study. Section 2 includes the Material and Methods. Section 3 comprises the proposed methodology and Results. Section 4 describes this research's Conclusion and future work.

2. Materials and Methods

We proposed a group of classifiers for “offline Urdu handwritten signature verification” to enhance its prediction ability. This system comprises five steps: “Data Acquisition, Pre-processing, Feature Extraction, Signature Registration, and Signature Verification” [30]. A histogram was used in data acquisition [31]. Pre-processing includes noise removal, binarization, smoothing, and thinning. Signature verification has KNN, SVMs, and ensemble classifiers. Figure 1 describes the proposed system for UHSV.

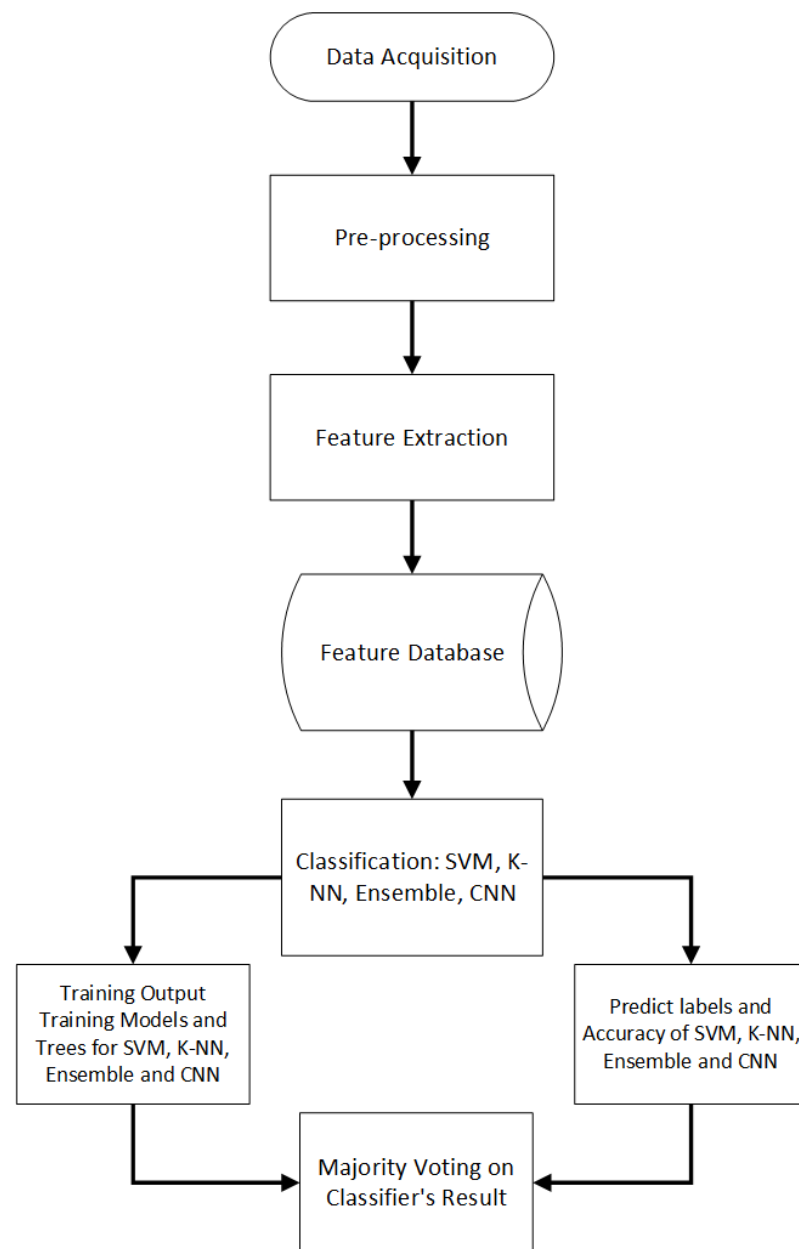


Figure 1. Embedded System for UHSV.

2.1. Classification Selection

This hybrid approach is based on the observation that when using KNN for handwritten character recognition, the correct class is almost always one of the two nearest neighbors of the KNN. An SVM is used to predict the accuracy of the class between two diverse classification assumptions. The embedded hybrid system of KNN, SVM, and ensemble classifiers results in significant changes in confirmation rates [32].

2.2. Hyperparameters of Classifiers

The convolutional neural network (CNN) uses two Conv2D layers followed by two MaxPooling2D layers, two Dense layers, and a Dropout layer. The CNN uses the ReLU activation function for the Conv2D and Dense layers, and the Softmax activation function for the output layer. The first Conv2D layer has 64 filters with a kernel size of (3, 3), which means that the layer will learn 64 different 3×3 convolutional filters to extract features from the input images. The second Conv2D layer has 32 filters with the same kernel size, which means it will learn 32 different filters to extract more abstract features from

the previous layer's output. The MaxPooling2D layers reduce the dimensionality of the feature maps produced by the Conv2D layers. The first MaxPooling2D layer has a pool size of (3, 3), which means it will reduce the feature map size by a factor of 3×3 , and the second MaxPooling2D layer has a pool size of (2, 2), which means it will reduce the feature map size by a factor of 2×2 [33]. The two Dense layers consist of 128 and 1 units and are connected to the flattened output of the second MaxPooling2D layer. The first Dense layer uses the ReLU activation function, which introduces non-linearity to the model and helps it to learn more complex representations. The second Dense layer has one unit and uses the Softmax activation function to produce a probability score for each of the two classes. The input images are resized to 224×224 with 3 color channels, which is a common image size used in CNNs [34].

The KNN classifier uses Models 5, 10, and 15 with seed = 7, 14, and 21, Kfold = 3, and hp-candidates = n-neighbor was (n, 2,3,4,5,6,7). The best hyperparameters for KNN were hp-candidate, (n, 3); seed, 21; and input model, 15. SVM uses 3 types of hyperparameters: learning rate = 0.2; subsample = above 7; hyperplane = 255×255 ; and SVM models = (5, 10, 15). The SVM best model hyperparameter was 15. The ensemble uses trees instead of models. A total of 200 trees were generated, such as (0, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200), and an ensemble uses these trees to generate different bags. The number of bags is similar to the number of trees. The total number of bags = (0, 20, 40, 60, 80, 100, 120, 140, 160, 180, 200). The best oob-loss was 0.65, and the reuse oob-loss at 200 trees was 0.00001. The best performance of the ensemble classifiers was achieved with the reuse oob-loss at 200 trees.

To create the hyperplane, SVM picks the most extreme points and vectors possible [35], while the parameters of SVM are kernels. The KNN method assigns an input sample to a category determined by which of the K closest samples it most closely resembles [36]. The KNN classifier uses the minimum distance to K neighbors to determine the output class [37] for a given input sample by performing the following steps: it finds the average distance $T_i \in T$ between both the input data set and all the training samples. Euclidean distance $d(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ is used as the parameter to find the nearest distance between the two points. Ensemble learning is an effective method of machine learning that combines numerous independent models (called base models) into a single superior prediction model. KNN is also used to generate the embedding of the system. In the proposed system, we generated user embedding and found items that are most relevant to it. In the testing phase, the most similar signature image was tested. This method is a computationally expensive operation by performing $O(N \times K)$, where N is the number of items, and K is the size of each embedding.

The ensemble classifier is used to improve accuracy over a typical classifier by combining the predictions of multiple classifiers. Bagging, often called bootstrap aggregating, is the process of combining results from several iterations of a predictive model. The independent training of models is followed by an averaging step. Using bagging, one hopes to reduce variability more than would be possible using a single model. If you want to estimate a parameter of a population, you can use a technique called "bootstrapping," which involves drawing samples at random from the population and replacing some of the data with new data. Initially, bootstrap samples are generated in bagging. Each sample is then processed by a regression or classification technique. For classification, the result is either the class with the most votes (hard voting) or the class with the highest average probability. This is where aggregation has a role in this context. Bagging mathematically can be represented as: $\widetilde{f}_{bag} = \widetilde{f}_1(x) + \widetilde{f}_1(x) + \dots + \widetilde{f}_b(x)$.

2.3. Pre-Processing

For experiments, signatures were taken on white papers from signers. Every single signer signed their signature five times. Three different forged signatures were generated by professional writers: random, skilled, and straightforward. Eight signatures from one signer were collected at the end of one sheet of paper. A total of 21 people took part in the sign-up for database creation. The first task was to convert all hardcopy text

signatures into softcopy images. The viability of the feature extraction stage relies on image scanning quality. Four steps of pre-processing were used in this proposed solution: (i) Noise removal—the noise removal function converts it to grayscale. The noise is removed using the wiener2 function; (ii) binarization—image binarization is the conversion of an image into two black-and-white colors; (iii) smoothing—Gaussian smoothing filters are commonly used to reduce noise; and (iv) thinning—a thinning procedure makes signatures, lines, and curves thinner and more precise. To prepare signature images from K, the image is binarized, and the noise from the signature image is removed. In the next step, the signature image is smoothed, and then, the image is thinned. These steps will also be repeated for images K and r until pre-processed image data are clear.

Pixel p_0 must be explored; when pixel p_0 is a bit of a skeleton, pixel p_0 will be deleted or not eradicated by conditions. Regardless, in a one-way handle, the estimation of pixel p_0 should be set to another concern, for instance, -1 , as shown by Hilditch's calculation [37], set to 0 until all pixels in the photo are examined along these lines. At this point, the procedure is rehased until no advancements are made. Figure 2 contains the handwritten signature images of Urdu script which are used in the pre-processing phase of the proposed system.

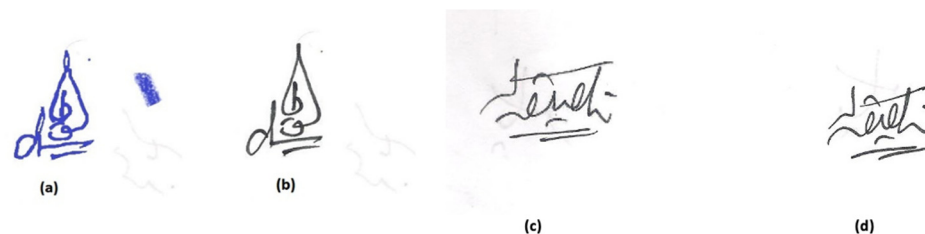


Figure 2. (a,c) after and (b,d) before pre-processing the image.

2.4. Feature Extraction

One of the important parts of this study is feature extraction. Feature extraction is an essential part of any image processing to increase the performance of embedded systems [38]. Many researchers use different feature extraction techniques to extract features from images. The standard feature extraction methods were already discussed above. We used the Gabor wavelet method for feature extraction [9,36]. The method, which is used for generating embeddings in the proposed system, is called principal component analysis (PCA), which was used in feature extraction. PCA reduces the dimensionality of an entity by compressing variables into a smaller subset. This allows the model to behave more effectively. The novelty of the feature extraction phase is that it extracts features from multiple signature images from a single signer and saves them into a single file. This will be used to compare the testing signature at the verification phase. This strategy helps reduce the verification complexity of classifiers. This research uses a feature of interest that improves the proposed system's performance. A single image's extracted features are 25,000. We used a feature of interest to overcome this massive number of features, making the system more reliable and accurate. The Gabor filter function drive reaction is characterized by a sinusoidal wave (a plane wave for 2D Gabor channels) increased by a Gaussian function. The channel has an imaginary and a real segment in orthogonal directions. The two parts may be framed into a complex number or be utilized separately.

$$gf(a, b; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{a'^2 + \lambda^2 b'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{a'}{\lambda} + \psi\right)\right) \quad (1)$$

$$gf(a, b; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{a'^2 + \lambda^2 b'^2}{2\sigma^2}\right) \cos\left(i\left(2\pi\frac{a'}{\lambda} + \psi\right)\right) \quad (2)$$

$$gf(a, b; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{a'^2 + \lambda^2 b'^2}{2\sigma^2}\right) \cos\left(i\left(2\pi\frac{a'}{\lambda} + \psi\right)\right) \quad (3)$$

Here, the pre-processed image from K is read, and then, the Gabor wavelet function for feature extraction with the parameter c to extract the features of interest from the extracted features is applied. The c from column to row is converted. All these processes will be repeated for images K and R until all of the features are extracted. In this mathematical statement, λ expresses the wavelength of the sinusoidal component, θ speaks to the introduction of the ordinary to the corresponding lines of a Gabor capacity, ψ is the stage balance, σ is the sigma/average deviation in the Gaussian cover, and γ is the 3D angle proportion, which indicates the sphericity of the backing of the Gabor filter. The Gabor filter is a process for every image to extract features. By utilizing a filter bank for analysis and a decimation process, DWT can be used to examine a picture. Each decomposition level is represented in the analysis filter bank by a pair of low- and high-pass filters. While the high-pass filter is responsible for extracting finer features such as edges, the low-pass filter is responsible for obtaining the rougher, more general information in the image. DWT has two levels: one for edge detection and one for image denoising. The Gabor filter function is used as a mother wavelet function in DWT. The use of a linear filter known as a Gabor filter makes feature extraction in image processing tasks easier. It is a bandpass filter, so it only lets through signals within a certain frequency range and mutes everything else.

2.5. Training

The extracted features are saved in the directory for classifier training and testing as calculated features. The reason for saving these features is to reduce the time required for feature extraction. All calculated features are saved with the same name as their labeled classes, A to U, which means 1 to 21 people. The training and testing algorithm read the calculated features c and labels l from k and divide the calculated features and labels into three datasets. The training time is more than the existing methods. A challenging task is creating a system that is effective and efficient for data collection. The databases' volume increases exponentially and quickly becomes unmanageable; therefore, the core of intelligent solutions is to reduce the raw data to sparse forms without losing crucial information. Hybrid classifiers for training are applied: SVM, KNN, and ensemble. Now, the system will read the calculated features c from r , then divide the calculated features c from r into three data sets. Handwritten classification and verification are significant and challenging problems due to two main factors. First, because signature consistency is frequently lacking, intra-personal variations in speed, pressure, and inclination can be significant. Second, there are not any forgeries, and we can only obtain a limited number of samples from a single person. Therefore, it is challenging to evaluate the dependability of retrieved features. The limitation of this framework is pre-processing computational time and cost.

2.6. Hybrid Classifier for Training and Testing

We proposed a novel hybrid classifier combining SVM, KNN, ensemble, and CNN. At the result stage, we took all the results and combined them. In addition, we implemented a new approach to obtain the actual and highly rated results. The MV algorithm was employed to achieve this task. The hybrid classifier helps achieve better performance and more accurate results. For the training phase, we labeled the target classes with every member's signature that we had taken before. These labels with extracted features from the Gabor filter were given an arrangement of training samples generally denoted by extracted features or "vectors". Every training set was named with a target class, an individual's signature from a limited and typically small arrangement of labeled classes. The objective of supervised learning is to correctly predict the class labels of samples in the testing phase. This method was repeated for every classifier in the hybrid approach, and the MV method is applied to achieve the accuracy of the proposed system. All the data sets are used to make a base classifier, and the "majority voting" of these base classifiers decides the final prediction of the character.

3. Results

For experiments, we created an Urdu handwritten signature dataset for testing the performance of our proposed embedded system. The data were collected from 21 individuals. Each person was asked to sign his or her signature six times correctly, the same as he or she could. This dataset was saved as a genuine signature database. We asked some writing experts to sign these signatures as a random, skilled, and straightforward forgery of forged handwritten signatures and saved them in the forged handwritten signature database. We had six genuine and three forged handwritten signatures for each person. So, nine from one person, and $9 \times 21 = 189$. This dataset was divided into 70% and 30% for training and testing, respectively.

For the training of “KNN, SVM, ensemble, and CNN classifiers”, the training set consisted of 70% of the dataset and was divided into three sub-models for better training. All handwritten signatures were labeled correctly. All labeled and analyzed features had inputs for “KNN”, “SVM”, and “ensemble classifier’s training”. The training models were saved separately.

3.1. Testing

Individual testing was conducted on each classifier, 30% of the entire dataset was used as the testing set and 70% as the training set. In the KNN and SVM classifiers, the training was done in three sub-models called Model 5, Model 10, and Model 15. Model 5 used 33% of the training set, Model 10 used 66% of the training set, and Model 15 used the whole training set. The purpose was for comparison and increase of the accuracy of the verification process. The overall accuracy includes testing results from all training and testing data.

Figure 3 demonstrates the accuracy of KNN in various models. This methodology required additional time but reduced the overall computational cost. The accuracy of KNN Model 5 is the highest, at 89.23%. The overall accuracy is 86.34%.

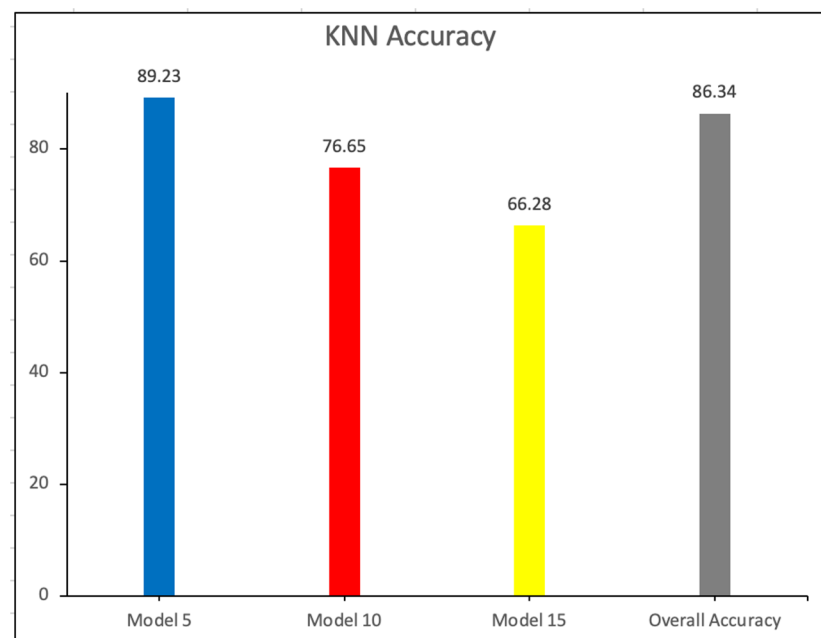


Figure 3. Accuracy of k-nearest neighbor (KNN) in different models.

The testing was also implemented with an SVM classifier. Figure 4 shows the accuracy of the SVM classifier in different models. The accuracy of the SVM classifier is 93.31%.

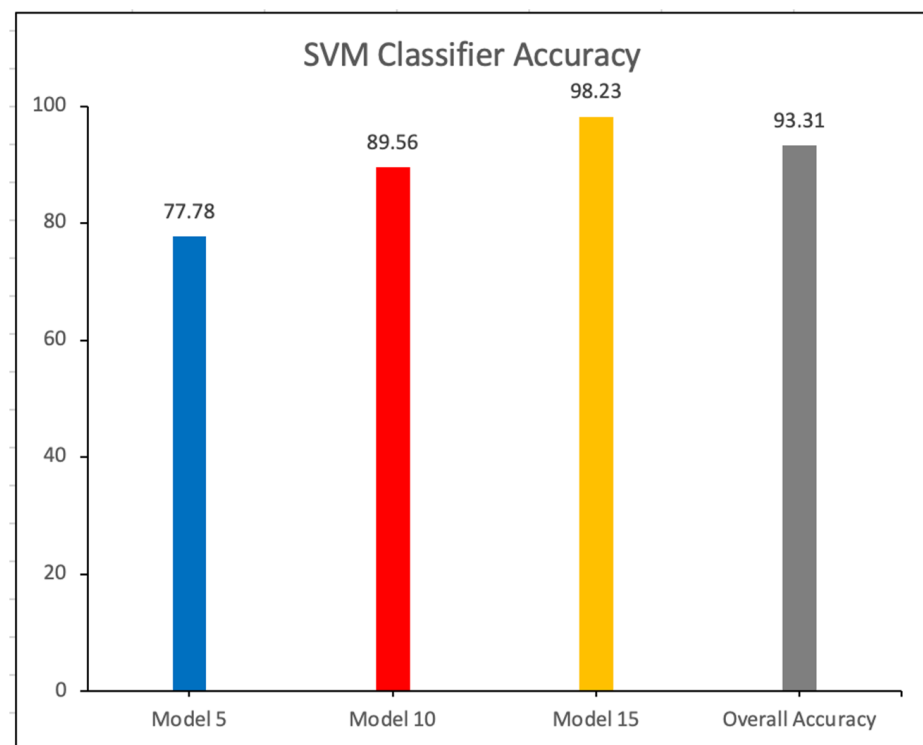


Figure 4. Accuracy of support vector machine (SVM) classifier in different models.

The testing was implemented with an embedded ensemble classifier, but the ensemble classifier had a different procedure. The ensemble classifier uses distinctive trees instead of models. Two further losses were calculated while testing with the embedded ensemble classifier. Figure 5 shows the loss of ensemble training, and Figure 6 shows the accuracy of the ensemble

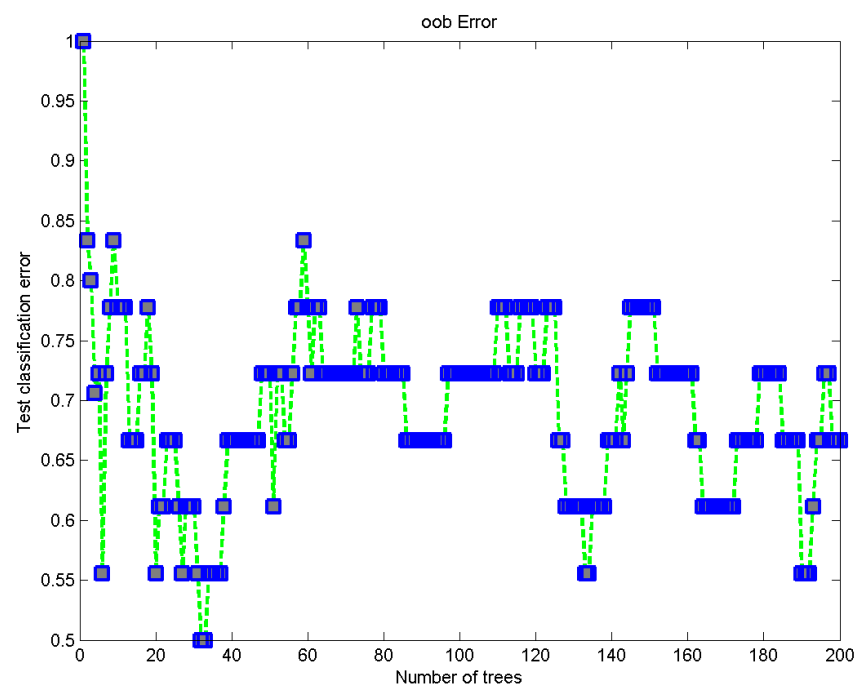


Figure 5. Bagging Loss of Training.

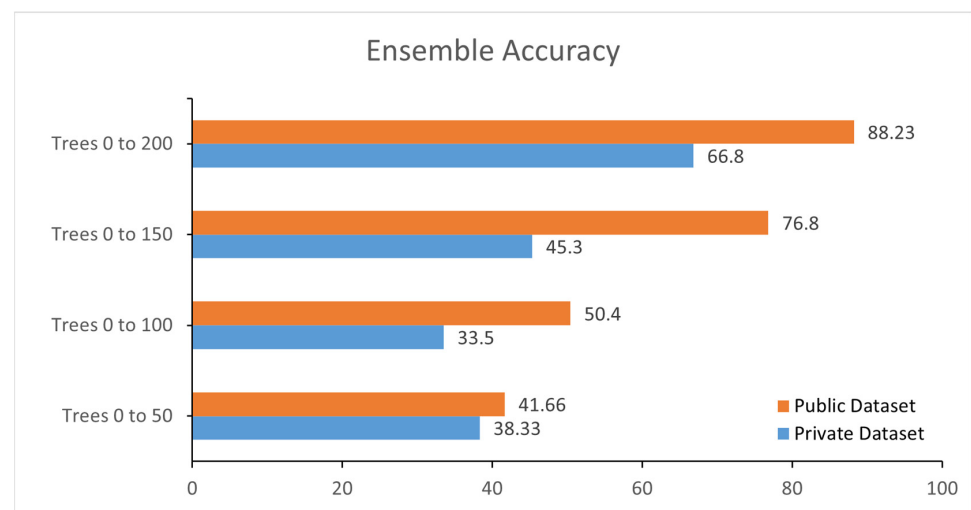


Figure 6. Overall accuracy of the ensemble classifier.

3.2. CNN

The proposed model now includes a sizable dataset that can be utilized for the training and testing and the CNN model used in offline signature verification is presented in Figure 7. CNN has been trained and tested as presented in Figure 8. CNN achieved the accuracy of 95.05% on private dataset and 95.27% on public dataset.

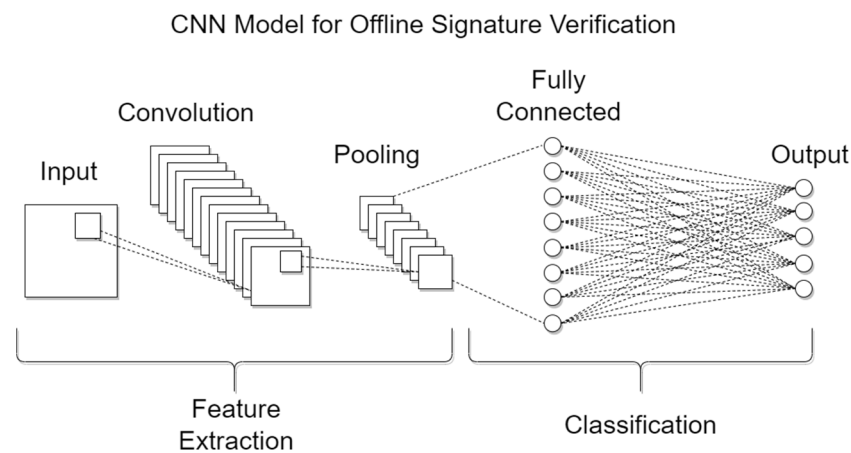


Figure 7. CNN Model for Offline Signature Verification.

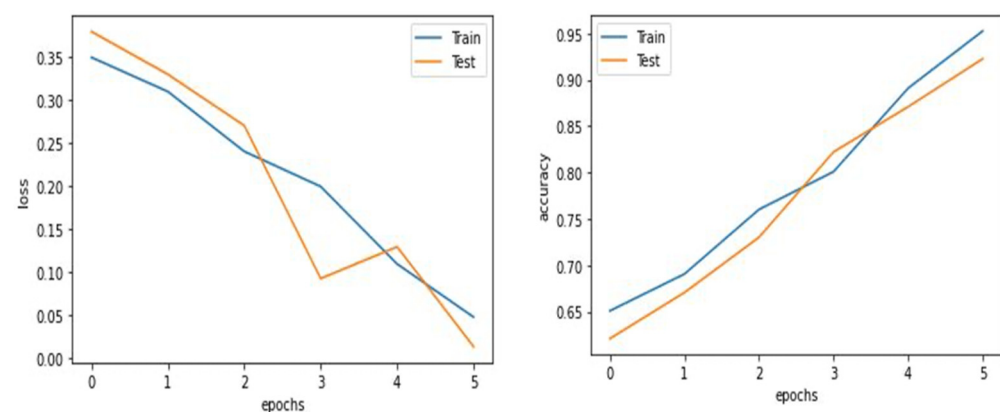


Figure 8. Training and Testing of CNN Model.

3.3. Majority Voting (MV)

Many voting algorithms have been used in [37] biometric authentication applications. It is used when we need to find the majority element (a value predicted by the classifier) from the predictions of multiple classifiers.

Figure 9 represents the comparison of the public dataset and private dataset. After applying majority voting on hybrid classifiers, the accuracy is increased. We used the MV algorithm in our model to ensure that the proposed embedded system has high performance. We needed an odd number of classifiers to use the MV algorithm.

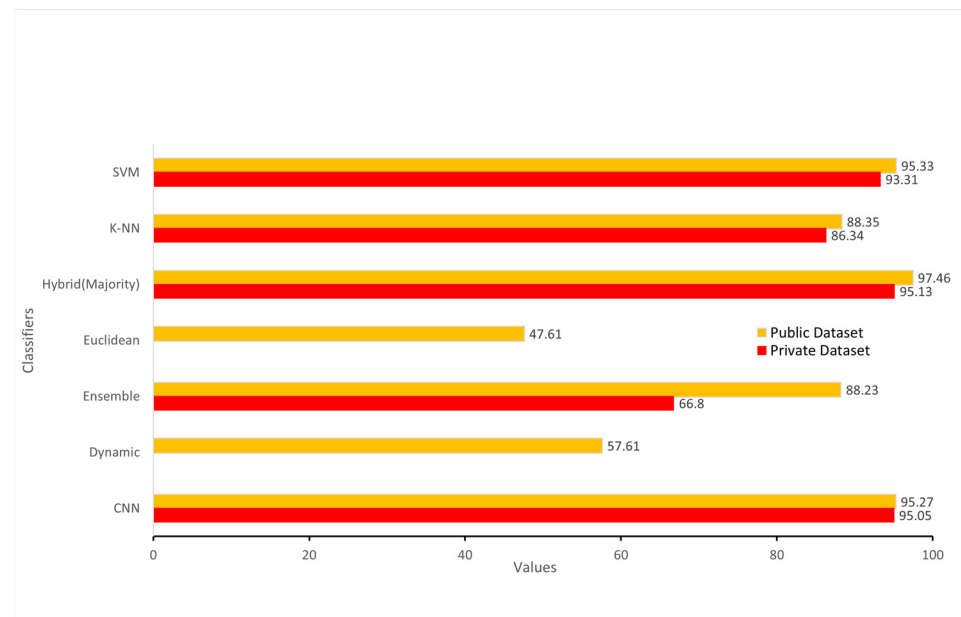


Figure 9. Comparison of the proposed method with existing published methods.

The confusion matrix in Figure 10 represents the performance comparison of the machine learning models (KNN, CNN, and SVM) as described in [38]. As this is a binary classification problem, the matrix was separated into true labels and false labels. Before using the MV, accuracy of the ensemble, KNN, and SVM classifier was 66.8%, 86.34% and 93.31%, respectively on private data set. We combined the results of all these classifiers and applied MV algorithm to achieve the highest accuracy of the proposed system. Table 1 shows the actual percentage of classifiers in MV algorithms for true labels. All three SVM, KNN, and ensemble classifiers voted 88.23% for true labels, which is indicated as “3-0 voting,” where 3 indicates the number of classifiers that predicted true labels. Both SVM and KNN classifiers, voted 97.46% for true labels, which is considered 2-1 voting, and SVM alone voted 98.23% for true labels, which is 1-2 voting.

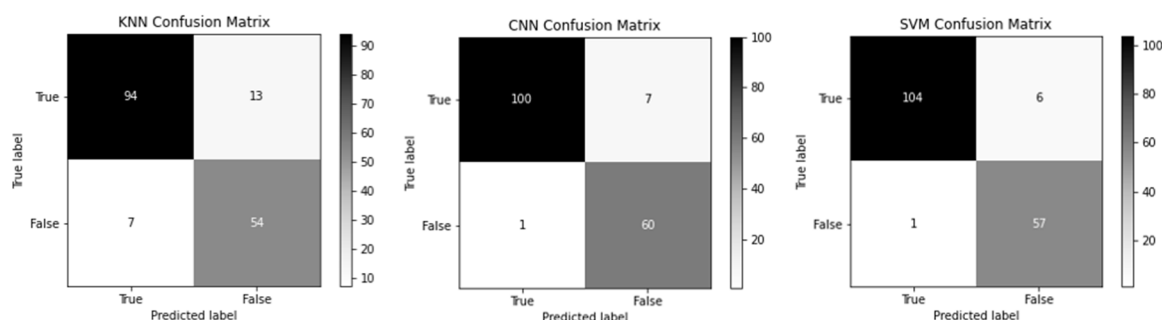


Figure 10. Confusion Matrix of KNN, CNN, and SVM Classifiers.

Table 1. Actual individual percentage of classifiers in MV algorithms for true labels.

No. of Classifiers Vote for True Labels	Classifiers	Voting Percentage %
3-0 voting	SVM, KNN, ensemble	88.23%
2-1 voting	SVM, KNN	97.46%
1-2 voting	SVM	98.23%

In this case, the majority of classifiers voted for 97.46%, which is considered the accuracy of our proposed model. This level of accuracy has never been achieved for Urdu signature verification.

Table 2 shows the voting percentage of classifiers that cannot vote for true labels. SVM, KNN, and ensemble classifiers could not vote for true labels with 1.76%, which is indicated as 0-3 voting, where 0 represents the number of classifiers that cannot predict true labels. KNN and ensemble were incapable of voting for true labels, which are considered 1-2 voting at 1.77% and 2-1 voting at 2.54%, respectively, whereas only ensemble classifiers could not vote for true labels. This is examined in Figure 7. The existing Arabic and Persian signature verification methods need to be revised for Urdu signature verification. We used Arabic and Persian offline signature verification systems to test our Urdu data set. The results show that both methods could not perform better on the Urdu signature data set. In comparison, the proposed method had much better performance. For the proposed embedded system to be generally useful, we tested our method on a public dataset of handwritten signatures, which can be found on the English “Kaggle” dataset of handwritten signatures [5]. It has 1800 signature, 860 genuine signature, and 840 forged signature datasets. The results on the public dataset are 88.23% with the ensemble, 88.35% with KNN, and 95.33% with SVM. When the MV algorithm was used, the accuracy on this public dataset after MV algorithm is 97.46%.

Table 2. Actual individual percentages of classifiers in MV algorithms that are unable to vote for true labels.

No. of Classifiers Unable to Vote for True Labels	Classifiers	Voting Percentage %
0-3 voting	-	1.76%
1-2 voting	SVM	1.77%
2-1 voting	SVM, KNN	2.54%

3.4. False Acceptance Rate (FAR) and False Rejection Rate (FRR)

An FAR (false acceptance rate) in Equation (4) indicates how often a biometric authentication system accepts a false value. In our case, it is the ratio at which our USV accepts forged signatures. FRR (false rejection rate) in Equation (5) means that the percentage of a biometric system rejecting false values in our system should reject a maximum number of forged signatures. A good biometric authentication system should have a low FAR and a high FRR. Table 3 show the comparison of proposed method with existing published methods.

$$FAR = \frac{\text{False Accepted}}{\text{Total Forged Matched}} \quad (4)$$

$$FRR = \frac{\text{False Rejected}}{\text{Total Genuine Matched}} \quad (5)$$

Researchers have used different methods for offline handwritten signature verification as compared to others our offline handwritten signature verification system is able to work better and have a low error rate. Table 1 illustrates that the proposed method better, with an accuracy of 97.46%. Table 2 show that the proposed hybrid classifier system has a very low

“FAR” of 0.02% and an FRR of 43.75%. Figure 11 shows FAR and FRR with other methods. Table 4 shows the error rate comparison of the proposed and other published methods.

Table 3. Comparison of the proposed method with other published methods.

No.	Classifiers/Features Used	FAR%	FRR%
1	Geometrical features [9]	0.1667	0.26667
2	SVMs [10]	7.25	11.1
3	Zernike moment feature [11]	17.96	20.10
4	DWT [13]	0.3	0.6
5	Local and global matching [15]	14.2	16.4
6	Fuzzy hybrid system [19]	9.04	6.27
7	HMM [22]	10.4	7.84
8	One-class SVM [24]	0.9	40.65
9	Fuzzy approach [29]	0.61	22.16
10	ANN [30]	0.033	4.7
11	Grid based [33]	11.21	19.33
12	Proposed high-performance embedded system	0.02	43.75

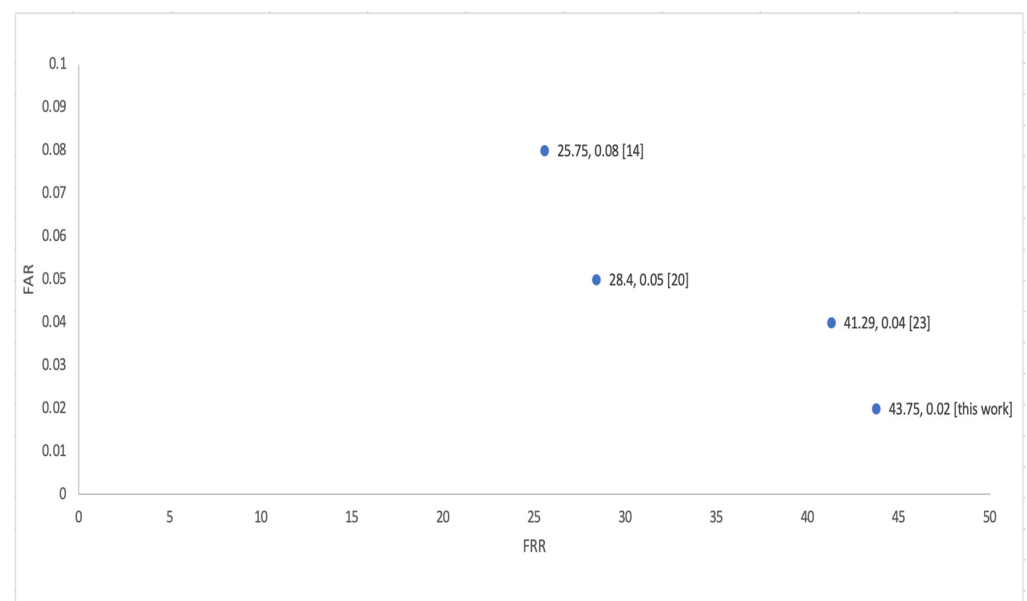


Figure 11. Comparison of FAR and FRR.

Table 4. Comparison of existing published methods with the proposed method.

Method	FAR	FRR	Accuracy
Arabic [14]	0.08	25.75	57.1%
Persian [20]	0.05	28.4	46.1%
Arabic [23]	0.04	41.29	85.1%
Proposed Urdu	0.02	43.75	95.13%

4. Conclusions

In this study, we developed an innovative method for offline systems to validate handwritten signatures. A dataset of handwritten signatures in Urdu was produced to train

and test the classifiers. A high-performance embedded system of KNN, SVM and ensemble classifiers with MV was used to categorize handwritten signature images, particularly for Urdu's handwritten signature verification. The CNN model also used in this work. The results show that the proposed embedded system has high performance on both the private dataset and the public datasets. In future work, other methods such as the generative adversarial network (GAN) can be compared with the proposed model. This model can be also used for retinal verification, offline fingerprint verification, computer vision recognition and verification in future work.

Author Contributions: For this research work, U.T. and Z.H. contributed to the conceptualization of this project. U.T. contributed to the methodology and experiment validation. R.T. contributed to database creation and writing. M.S.I. and M.S. contributed to the formal analysis, review, and editing. The whole work was supervised by Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: We thank the National Key R&D Program of China (2022YFB3605601 and 2017YFB0403602) for financial supports.

Data Availability Statement: The Data presented in this research is available on request from the corresponding author. The public data which is used for comparison of existing publish method can be access from: <https://www.kaggle.com/datasets/divyanshrai/handwritten-signatures>, accessed on 18 December 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ruiz-Barroso, P.; Castro, F.M.; Delgado-Escano, R.; Ramos-Cozar, J.; Guil, N. High performance inference of gait recognition models on embedded systems. *Sustain. Comput. Inform. Syst.* **2022**, *36*, 100814. [CrossRef]
2. Karrar, N.; Mohamad, D.; Saba, T.; Rehman, A. *Discriminative Features Mining for Offline and Written Signature Verification*; 3D Research Center, Kwangwoon University: Seoul, Korea; Springer: Berlin/Heidelberg, Germany, 2014; Volume 4, p. 7.
3. Guerbai, Y.; Chinabi, Y.; Hadjadj, B. The Effective Use of the One-Class SVM Classifier for Reduced Training Samples and Its Application To Handwritten Signature Verification. In Proceedings of the 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, 14–16 April 2014.
4. Ferrer, M.A.; Vargas, J.F.; Morales, A.; Ordonez, A. Robustness of Offline Signature Verification Based on Gray Level Features. *IEEE Trans. Inf. Forensic Secur.* **2012**, *7*, 966–977. [CrossRef]
5. Kaggle, K. 2018. Available online: <https://www.kaggle.com/divyanshrai/handwritten-signatures> (accessed on 10 February 2021).
6. Ghosh, R. A Recurrent Neural Network based deep learning model for offline signature verification and recognition system. *Expert Syst. Appl.* **2021**, *168*, 114249. [CrossRef]
7. Saxe, A.; Nelli, S.; Summerfield, C. If deep learning is the answer, what is the question? *Nat. Rev. Neurosci.* **2021**, *22*, 55–67. [CrossRef]
8. Fan, J.; Ma, C.; Zhong, Y. A selective overview of deep learning. *Stat. Sci. A Rev. J. Inst. Math. Stat.* **2021**, *3*, 264. [CrossRef]
9. Niu, X.X.; Suen, C.Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits. *Pattern Recognit.* **2011**, *45*, 1318–1325. [CrossRef]
10. Shah, A.S.; Khan, M.N.A.; Subhan, F.; Fayaz, M.; Shah, A. An offline Signature Verification Technique Using Pixels Intensity Level. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2016**, *9*, 205–222. [CrossRef]
11. Vargas, J.; Ferrer, M.; Travieso, C.; Alonso, J. Off-line signature verification based on grey level information using texture features. *Pattern Recognit.* **2011**, *44*, 375–385. [CrossRef]
12. Yu, Y.; Samali, B.; Rashidi, M.; Mohammadi, M.; Nguyen, T.N.; Zhang, G. Vision-based concrete crack detection using a hybrid framework considering noise effect. *J. Build. Eng.* **2022**, *61*, 105246. [CrossRef]
13. Huang, S.; Jin, X.; Jiang, Q.; Liu, L. Deep learning for image colorization: Current and future prospects. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105006. [CrossRef]
14. Yu, Y.; Liang, S.; Samali, B.; Nguyen, T.N.; Zhai, C.; Li, J.; Xie, X. Torsional capacity evaluation of RC beams using an improved bird swarm algorithm optimised 2D convolutional neural network. *Eng. Struct.* **2022**, *273*, 115066. [CrossRef]
15. Hezil, H.; Djemili, R.; Bourouba, H. Signature recognition using binary features and KNN. *Int. J. Biom.* **2018**, *10*, 1–15. [CrossRef]
16. Kumar, R.; Singhal, P. Review on Offline Signature Verification by SVM. *Int. Res. J. Eng. Technol. (IRJET)* **2017**, *4*, 1771–1773.
17. Ueda, K. Investigation of Off-Line Japanese Signature Verification Using a Pattern Matching. In Proceedings of the Seventh International Conference on Document Analysis and Recognition IEEE, Edinburgh, Scotland, 3–6 August 2003.
18. Bulacu, M.; Schomaker, L.; Brink, A. Text-Independent Writer Identification and Verification on Offline Arabic Handwriting. In Proceedings of the Ninth International Conference Document Analysis and Recognition, Curitiba, Brazil, 23–26 September 2007.

19. Ghandali, M. Offline Persain signature Identification and Verification based on image registration and fusion. *J. Multimedia* **2009**, *4*, 41–49.
20. Kim, H.; Moon, H.; Ahn, H. A weight-adjusted voting algorithm for ensembles of classifiers. *J. Korean Stat. Soc.* **2011**, *40*, 437–449. [[CrossRef](#)]
21. Zhou, P.; Zhao, Z.; Zhang, K.; Li, C.; Wang, C. An end-to-end model for chinese calligraphy generation. *Multimed. Tools Appl.* **2021**, *80*, 6737–6754. [[CrossRef](#)]
22. Nandanwar, L.; Shivakumara, P.; Jalab, H.A.; Ibrahim, R.W.; Raghavendra, R.; Pal, U.; Lu, T.; Blumenstein, M. A Conformable Moments-Based Deep Learning System for Forged Handwriting Detection. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *1*, 1–14. [[CrossRef](#)]
23. Almisreb, A.A.; Tahir, N.; Turaev, S.; Saleh, M.A.; Al Junid, S.A.M. Arabic Handwriting Classification using Deep Transfer Learning Techniques. *Pertanika J. Sci. Technol.* **2022**, *30*, 641–654. [[CrossRef](#)]
24. Sudha, L.R.; Bhavani, R. A combined classifier kNN-SVM in gait-based Biometric authentication system. *Int. J. Comput. Appl. Technol.* **2014**, *49*, 113–121. [[CrossRef](#)]
25. Hadjadj, I.; Gattal, A.; Djeddi, C.; Ayad, M.; Siddiqi, I.; Abass, F. Offline Signature Verification Using Textural Descriptors. *Iber. Conf. Pattern Recognit. Image Anal.* **2019**, *3*, 177–188.
26. Muhtar, Y. A Survey of Offline Handwritten Signature Verification Based on Deep Learning. In Proceedings of the 2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML), Changdu, China, 22–24 July 2022.
27. Roach, L. *Ensemble-Based Classifiers*; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2009.
28. Muralidharan, R.; Chandrasekar, C. Object Recognition using SVM-KNN based on Geometric Moment Invariant. *Int. J. Comput. Trends Technol.* **2011**, *5*, 9.
29. Batool, F.E.; Attique, M.; Sharif, M.; Javed, K.; Nazir, M.; Abbasi, A.A.; Iqbal, Z.; Riaz, N. Offline signature verification system: A novel technique of fusion of GLCM and geometric features using SVM. *Multimedia Tools Appl.* **2020**, *3*, 1–20. [[CrossRef](#)]
30. Pan, J.; Ye, N.; Yu, H.; Hong, T.; Al-Rubaye, S.; Mumtaz, S.; Al-Dulaimi, A.; Chih-Lin, I. AI-Driven Blind Signature Classification for IoT Connectivity: A Deep Learning Approach. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 6033–6047. [[CrossRef](#)]
31. Angadi, S.; Gour, S. Euclidean Distance Based Offline Signature Recognition System Using Global and Local Wavelet Features. In Proceedings of the Fifth International Conference on Signals and Image Processing, Bangalore, India, 8–10 January 2014.
32. Azami, A.N.; Nasien, D.; Omer, F.S.D. Biometric signature verification system based on freeman chain code and k-nearest neighbor. *Multimed. Tools Appl.* **2017**, *10*, 15341–15355. [[CrossRef](#)]
33. Ahmed, S.M.A.R. Off-Line Arabic Signature Verification Using Geometrical Features. In *Computer & Information Systems*; Department of Computer Science Jazan University: Jazan, Saudi Arabia, 2012.
34. Yu, J.; Li, Y. Improving Hilditch Thinning Algorithms for Text Image. In Proceedings of the 2009 International Conference on E-Learning, E-Business, Enterprise Information Systems, and E-Governmen, Hong Kong, China, 5–6 December 2009.
35. Udendhran, R.; Balamurugan, M.; Suresh, A.; Varatharajan, R. Enhancing image processing architecture using deep learning for embedded vision systems. *Microprocess. Microsystems* **2020**, *76*, 103094. [[CrossRef](#)]
36. Ahmed, S.; Frikha, M.; Hussein, T.D.H.; Rahebi, J. Optimum Feature Selection with Particle Swarm Optimization to Face Recognition System Using Gabor Wavelet Transform and Deep Learning. *BioMed Res. Int.* **2021**, *2021*, 6621540. [[CrossRef](#)] [[PubMed](#)]
37. Pourdarbani, R.; Sabzi, S.; Kalantari, D.; Hernández-Hernández, J.L.; Arribas, J.I. A Computer Vision System Based on Majority-Voting Ensemble Neural Network for the Automatic Classification of Three Chickpea Varieties. *Foods* **2020**, *9*, 113. [[CrossRef](#)] [[PubMed](#)]
38. Philip, L. *Locally Linear Support Vector Machines*; University of Oxford, Department of Engineering Science: Oxford, UK, 2011.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.