International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 01 | Jan 2021 www.irjet.net p-ISSN: 2395-0072

HANDWRITTEN SIGNATURES FORGERY DETECTION

Kshitij Swapnil Jain[1], Udit Amit Patel[2], Rushab Kheni[3]

[1]*Student, School of Information Technology, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
[2]*Student, Dept. of Mechatronics, Mukesh Patel School of Technology Management and Engineering, Mumbai, Maharashtra, India*
[3]*Student, Dept. of Computer Science, MCT's Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India*

ABSTRACT: *Signature plays an important role in banking, financial, commercial etc. Signature may be unique for each person. However, with signatures comes many challenges since any two signatures may look very similar with little to no differences written by the same person. As there are unique and important variations in the feature elements of each signature, thus in order to match a particular signature with the database, the structural parameters of the signatures along with the local variations in the signature characteristics are used. In order to avoid any such identity crimes committed in banks and many other companies, forgery detection systems is a solution to this problem along with the help of the concepts of machine learning algorithms and CNN. This software can be used to validate signatures across many platforms like loans, legal document signing, application signing, applying and much more.*

Key words: CNN, ResNet, Fuzzy Logic, Backpropogation, ReLu, ANN

1. INTRODUCTION

The need for signature verification is very much important because unlike passwords signatures cannot be changed or forgotten because it is unique for everyone and it is considered as the important method for verification. The techniques and system used to solve signature verification is divided into offline signature and online signature methods. In offline signature verification method more number of hardware's were not used and images were captured using camera, whereas in online verification method more hardware's were used and the hardware's were directly connected to the computer. The features used for offline verification are simpler. The signatures from the database are preprocessed using various preprocessing techniques then the preprocessed database features were extracted. The automatic off-line signature verification solutions can be classified into two categories: handcrafted feature extraction algorithms and deep learning methods. The deep learning methods are especially considered to be the most promising approach for its great capability for image recognition and detection. Although studies of deep learning with small-scale data are getting considerable attention in recent years, most deep learning methods still need a large number of samples to train their system. In other words, most of the studies still need several (more than one) signature samples to accomplish their training process. In this paper, we propose an off-line handwritten signature verification method using convolution neural network (CNN). Signature forgery detection finds its application in the field of net banking, passport verification system, credit card transactions and bank checks. Therefore, with the growing demand for protection of individual identity, the design of an automatic signature system is needed.

2.0 OBJECTIVE

The objective of the software is:

- To verify if a given signature is original or forged.
- To understand the characteristics of a signature.
- To implement the system.

3.1 LITERATURE REVIEW

Hanmandlu et al. [1], (2005), have identified and proposed a new system for handwritten signature verification. Using the quadtree structure of the histogram template, the paper proposes a new descriptor. The methodology used for the verification of signatures includes usage of Artificial Immune Recognition System (AIRS). The classifier derives its implementation from a natural immune system. By using AIRS training, new cells are developed which are subsequently recognised by a KNN (k-nearest neighbour classifier). The paper shows 6 KNN classifiers can also be substituted by an SVM (support vector machine) in order to get robust and better classification. Three datasets that were used to perform experiments on: namely the MYCT75, GPDS-300 and GPDS-4000. Using the AIRSVM gives better results as compared to AIRS or SVM classifier.

Quek and Zhou [2], (2002), have proposed a system which is constructed on the basis of a novel fuzzy neural network called the POPFNN-TVR. The strengths of POPFNN-TVR are namely its learning ability, its generalization ability, and advanced computational ability, this all in combination make identifying forged images very powerful when verifying pretty skillfully forged images.. To prove the efficiency of POPFNN-TVR and its application in the antiforgery system, numerous experiments have been framed out and implemented in this paper. The experimental results and analysis for each of those experiments are presented at the end of the paper for further discussion.

Hanmandlu et al [3], (2003), have proposed a novel approach known as the box method which is used for feature extraction for the identification of handwritten characters. Under this methodology the input binary image is divided into a number of subimages known boxes. Vector distance is calculated based on some fixed point, and is added and normalized by the number of pixels in that particular box or subimage. Both of the methodology namely neural networks and fuzzy logic are used and they deliver very promising results which is around 97 percent using neural networks and 98 percent using fuzzy logic. The results are independent of various parameters like fonts, size etc.

Xiao et al [4], (2020), have explored a two-part splicing forgery detection method. The two parts consist of a coarse-to-refined convolutional neural network(C2RNet) and a diluted adaptive clustering network. In the proposed model the differences in the image are found by cascading a coarse CNN and refined CNN (C-CNN and R-CNN respectively). The cascading results in making scales where the image has been tampered finding difference in their properties. The computational complexity of the whole model is reduced by an imagelevel CNN rather than using a patch level CNN into C2RNet. Since the difference in properties is compared therefore it results in stabilised results. It was found that the proposed method produces better results than the already existent splicing techniques for forgery detection even in conditions of attack. However, the size of these datasets restricts training and hence optimal results are not yet obtained from this proposed model.

Ruiz et al [5], (2020), have proposed the use of Siamese Neural Networks for performing the task of forgery detection in offline signature verification process with a writer independent context. Additional training is not required for adding new signs for verification. Further, three types of data were analysed so as to increase the amount of samples and variability needed for training the deep neural network. The datasets used were the GAVAB dataset, a proposal of compositional synthetic signature generation from shape primitives and the GPDSSynthetic dataset. The first two approaches can be categorised as on-demand approaches and can produce infinitely many signatures which are synthetic in nature. In the given approach, initially training is done by Siamese Neural Networks using the first mentioned dataset. When original and synthetic signatures are mixed it leads to the best result for training. Further few other datasets such as GPSythetic, MCYT, Sigcomp11 etc, were also used to test and create a general model.

Abuhaiba and Ahmed [6], (1993), have proposed an automatic off-line character recognition system for completely unconstrained handwritten numerals. They used around 1763 unnormalized samples of data collected by U.S. Postal Services Department from letter envelopes. A realistic set of 105 Fuzzy Constrained Character Graph Models (FCCGMs) was developed through the training phase. In scale, shape and writing style, FCCGMs tolerate great variability. Characters were recognised by applying a set of rules to the FCCGM to match a representation of a character tree. By first translating the character skeleton into an approximate polygon and then transforming the polygon into a tree structure suitable for recognition purposes, a character tree is obtained. The model has been tested on 1812 unnormalized samples (not including the training set) and it proved to be effective for multi-writer, multi-pen, multi-textured paper, and multi-color ink in recognition rate and tolerance.

Walker et al [7], (1988), have proposed a novel recognition system that uses a smart thinning algorithm to produce stick figure images from scanned characters. The targeted symbols and criteria for accuracy from a mapping and charting environment presented problems that required new OCR techniques to be developed. In order to extract only topological, geometrical and local indicators that are needed to recognise the character or to reject the character as unrecognisable, the recognition logic interacts with the feature extraction algorithms. The substitution error rate is 0.3% and the rejection rate is 2.7%. There is also a discussion of preliminary findings for the identification of alphabetic characters.

Blanco et al [8], (2014), observed that biometric verification is increasingly being used ever since the use of mobile has increased and handwritten signatures are perhaps the most common way to carry out this task. This study obtained multiple results from 43 users signing 60 times, which were divided in three sessions and captured in 8 specific devices out of which six were mobile devices and the remaining were made specifically for signature collection. Each session captured 20 signatures per user and stored it in ISO/IEC 19794–7 format. The algorithm applied is a DTW-based algorithm which is particularly useful for mobile environments. Thus, the results which were obtained were based on interoperability, feedback and modality tests. These conclusions will finally help in creating more accurate models for the purpose of signature verification.

Zheru et al [9], (1995), have proposed a novel recognition methodology which uses combined self organising maps and fuzzy rules to classify the data. The SOM algorithm is used in the learning phase to produce prototypes which are used to determine fuzzy regions and membership functions together with corresponding variations. By learning from training patterns, fuzzy rules are then created. An input pattern is categorised by a fuzzy rule based classifier in the recognition stage. Then an uncertain pattern is re-classified by a SOM classifier. Experiments on a 20,852 handwritten numeral database (10,426 used for training and a further 10,426 for testing) show that this combination technique produces satisfactory results in terms of accuracy.

Bertolini et al [10], (2010), highlighted two important issues of off-line signature verification. The first one regards feature extraction. They introduce a new graphometric feature set that considers the curvature of the most important segments, perceptually speaking, of the signature.

The idea is to simulate the shape of the signature by using Bezier curves and then extract features from these curves. The second important aspect is the use of an ensemble of classifiers based on graphometric features to improve the reliability of the classification, hence reducing the false acceptance. The ensemble was built using a standard genetic algorithm and different fitness functions were assessed to drive the search.

Yilmaz and Yanikoglu [11], (2016), presented a system that uses a score-level fusion of complementary classifiers that use different local features (histogram of oriented gradients, local binary patterns and scale invariant feature transform descriptors), where each classifier uses a feature-level fusion to represent local features at coarse-to-fine levels. For classifiers, two different approaches are investigated, namely global and user-dependent classifiers. User-dependent classifiers are trained separately for each user, to learn to differentiate that user's genuine signatures from other signatures; while a single global classifier is trained with difference vectors of query and reference signatures of all users in the training set, to learn the importance of different types of dissimilarities.

Sabourin et al. [12], (1997), have proposed a new formalism for signature representation based on visual perception. A signature image consists of 512×128 pixels and is centered on a grid of rectangular retinas which are excited by local portions of the signature. Granulometric size distributions are used for the definition of local shape descriptors in an attempt to characterize the amount of signal activity exciting each retina on the focus of the attention grid. Experimental evaluation of this scheme is made using a signature database of 800 genuine signatures from 20 individuals. Two types of classifiers, a nearest neighbor and a threshold classifier, show a total error rate below 0.02 percent and 1.0 percent, respectively, in the context of random forgeries

Shridhar et al [13], (1995), have proposed multiple experts to be the strategy for handwritten recognition. A multiple expert framework using neural networks is introduced in this paper. In the proposed framework, the authors developed: (1) an incremental neural network clustering algorithm with merging and suspending the process, (2) a changed method of extraction of directional histogram features, and (3) a subclass method with a neuron strategy for learning rejection. The performance and robustness of the proposed system are prominent from our experimental results on a large set of data.

Hong [14], (1994), has proposed a novel method for digit recognition using a nearest neighbor classifier. A set of prototypes is obtained from the training samples in this process and used to create a nearest neighbour classifier. The classifier is then mapped to a perceptron that is multi-layer. The neural network is mapped back to the closest neighbour classifier with new and improved prototypes after training. Using this strategy, from 93.7 percent with 100 prototypes to 96.2 percent with 500 prototypes for samples not used for testing, he was able to achieve high recognition accuracy.

Ching [15], (1982), has proposed a method of recognising the image using a traditional classification model. Typically, before hitting the function extractor, the input character is smoothed and cleaned by the preprocessor. The prerequisites for an effective character recognition system are good preprocessors and feature extractors. The different characteristics contained in the vast accumulation of handprint recognition literature are divided into two key categories after a description of preprocessing techniques: (1) global analysis and (2) structural analysis. Further subdivision of these classifications gives rise to six feature type families, viz. (a) point distribution, (b) transformation, (c) physical dimensions, (d) line segments and edges, (e) character outline, and (f) character centreline. With illustrative examples, each family is outlined and the results of the experiments are discussed at the end of the paper.

Toru [16], (1993), has proposed a solution to the problem of establishing a robust methodology for handwritten character recognition. First, to clarify the state of the art and remaining issues, conventional techniques in Japan are critically surveyed. Second, from the perspective of general pattern recognition methodology, new and promising approaches are enumerated. Third, using the concept of local affine transformation, a challenge to distortion-tolerant shape matching is defined. Finally, explored the interesting research issues relevant to the recognition of handwritten characters in mailing addresses.

Drouhard et al. [17], (1996), have proposed a neural network approach to build the first stage of an Automatic Handwritten Signature Verification System. The directional Probability Density Function was used as a global shape factor and its discriminating power was enhanced by reducing its cardinality via filtering. Various experimental protocols were used to implement the backpropagation network (BPN) classifier. A comparison, on the same database and with the same decision rule, shows that the BPN classifier is clearly better than the threshold classifier and compares favourably with the k-Nearest-Neighbour classifier.

Guerbai et al. [18], (2015), have proposed the use of One-Class Support Vector Machine (OC-SVM) based on writer-independent parameters, which takes into consideration only genuine signatures and when forgery signatures are lack as counterexamples for designing the HSVS. The OC-SVM is effective when large samples are available for providing an accurate classification. However, available handwritten signature samples are often reduced and therefore the OC-SVM generates inaccurate training and the classification is not well performed. In order to reduce the misclassification, they proposed a modification of the decision function used in the OC-SVM by adjusting carefully

the optimal threshold through combining different distances used into the OC-SVM kernel.

Sigari et al. [19], (2011), have proposed a new method for offline (static) handwritten signature identification and verification based on Gabor wavelet transform. The whole idea is offering a simple and robust method for extracting features based on Gabor Wavelet which the dependency of the method to the nationality of signer has been reduced to its minimal. The advantages of this system is its capability of signature identification and verification of different nationalities; thus it has been tested on four signature dataset with different nationalities including Iranian, Turkish, South African and Spanish signatures.

Hafemann et al. [20], (2017), have proposed a method to overcome the difficulty of obtaining good features as well as improve system performance by including the knowledge of skilled forgeries from a subset of users in the feature learning process. It aims to capture visual cues that distinguish genuine signatures and forgeries regardless of the user. This is achieved by learning the representations from signature images, in a Writer-Independent format, using Convolutional Neural Networks. There were four datasets on which experiments were conducted. The paper showed that features learned in a writer-independent way were effective for signature verification, improving performance on the task, compared to the methods that rely on hand-engineered features. However, it suffered from the drawback that the model learned with the GPDS dataset did not improve in all cases.

Wen et al. [21], (2009), have proposed two models utilizing rotation invariant structure features to tackle the problem. In principle, the elaborately extracted ring-peripheral features are able to describe internal and external structure changes of signatures periodically. In order to evaluate match score quantitatively, discrete fast fourier transform is employed to eliminate phase shift and verification is conducted based on a distance model. In addition, the ring-hidden Markov model (HMM) is constructed to directly evaluate similar between test signature and training samples. With respect to the side effect of outlier training samples for stable statistical model and threshold estimation, they proposed a selection strategy to improve the performance of system. Experimental results demonstrated that the proposed methods were effective to improve verification accuracy.

Yingyong et al. [22], (1994) have proposed algorithms for extracting global geometric and local grid features of signature images were developed. These features were combined to build a multi-scale verification function. This multi-scale verification function was evaluated using statistical procedures. Results indicated that the multi-scale verification function yielded a lower verification error rate and higher reliability than the single-scale verification function using either global geometric or local grid feature representation. The correct verification rate of the multi-scale system was more than 90% in rejecting skilled forgeries and was perfect in rejecting simple forgeries based on a limited database.

Gideon et al. [23], (2018), have proposed a method for the classification of offline signatures as genuine or forged. The database of signatures is collected, image processing techniques such as RGB to grayscale, removal of noise, grayscale to bitmap and resizing are carried out on images as preprocessing techniques using MATLAB. The training of the model is used by Keras library in python with TensorFlow backend to implement Convolutional Neural Networks. The dataset used is a collection of 6000 signatures with 1000 genuine and 1000 forged signatures per subject. High accuracy was obtained on splitting the data as 8:2 ratio of training and test data which decreases on making the ratio as 7:3 and 6:4. The drawback was that the structure of the fully connected layer is not optimal. Future work includes deriving a custom loss function to predict the user to which signature belongs and to detect if it is genuine or forged.

Hadjadji et al. [24], (2017), have proposed an Open Handwritten Signature Identification System (OHSIS) for offline handwritten signature identification by using conjointly the Curvelet Transform (CT) and the One-Class classifier based on Principal Component Analysis (OCPCA). Binarization of acquired signature is done as a pre-processing method. CT is explored for feature generation due to its efficient characterization of curves contained into the local orientations within the signature image. While OC-PCA is used for its effectiveness to absorb the high feature size generated by the CT and allows achieving at the same time an open system new combination approach based on Choquet fuzzy integral is proposed to combine multiple individual OHSISs in order to improve the robustness of the OHSIS. Evaluation is done on the basis of Identification rate which is the number of instances correctly identified to the total number of instances in percentage.

Vargas et al. [25], (2011), have proposed a method for conducting off-line handwritten signature verification.. It works at the global image level and measures the grey level variations in the image using statistical texture features. The co-occurrence matrix and local binary pattern are analysed and used as features. This method begins with a proposed background removal. A histogram is also processed to reduce the influence of different writing ink pens used by signers. Genuine samples and random forgeries have been used to train an SVM model and random and skilled forgeries have been used for testing it. Results are reasonable according to the state-of-the-art and approaches that use the same two databases: MCYT-75 and GPDS-100 Corpuses. The combination of the proposed features and those proposed by other authors, based on geometric information, also promises improvements in performance.

Sabourin et al. [26], (1994), have proposed a new concept of representation and interpretation of handwritten signature images. The segmentation process breaks up the

signature into a collection of arbitrarily-shaped primitives. In the next step, a local interpretation process serves as a sophisticated template matching, permitting the labeling of all primitives from the test primitive set. This is followed by the global interpretation process, which permits the evaluation of a similarity measure between two structural graphs. Experimental results obtained from a database of 800 handwritten signature images from 20 writers showed good performance in the best strategy proposed using a minimum-distance classifier and two reference signatures

Fazli et al. [27], (2015), have stated that signature is a useful biological technique to recognize persons because of its ease of use and fast processing. This paper discusses offline signature recognition and verification systems using neural networks. This system enables the user to verify whether a signature is original or a fake. The proposed system includes three main steps: preprocessing of image, feature extraction and classification. This system can recognize 700 signatures with a recognition ratio of 93.5% and verification original signatures from fake ones with a ratio of 97%.

Ooi et al. [28], (2016), have proposed a method to compensate the lack of dynamic information from static signature images through the use of discrete Radon transform (DRT), principal component analysis (PCA) and probabilistic neural network (PNN). Median filter and grey-scaling of images is done to remove noise and minimize database storage of images. transforms the two dimensional images with lines into a domain of possible line parameters. PCA is utilized here for feature data compression. This paper uses a Probabilistic neural network (PNN) instead of similarity matching concept. A PNN has three layers – pattern layer which has one neuron for each input layer vector in the training set, summation layer which has one neuron for each user class and an output layer which holds the maximum value of summation of neurons to produce the probability score. The dataset used has 1000 genuine signatures, 500 casual forgeries and 500 skilled forgeries, which were collected from 100 writers and 10 forgers. Future works include using a large database of signatures with forgeries and a powerful specification of PC support to obtain a more reliable system.

Ghandhali et al. [29], (2012), have stated that biometric features have great importance in authentication systems nowadays. One of the most important and conventional biometrics is signature. In this paper, they have proposed a system which has two independent phases for offline signature identification and verification. The identification phase is based on Triangular Spatial Relationship (TSR) that is a rotation invariant feature extraction method. Also, a symbolic representation of signature has been employed to make using TSR possible. In the verification phase, a hybrid method is proposed that combines Discrete Wavelet Transform (DWT), Gabor filter, and image fusion methods. Experimental results on some benchmarks have confirmed the robustness and precision of the proposed method together with its robustness against translation, scaling, and rotation.

Alonzo et al. [30], (2005), have presented a set of geometric signature features for offline automatic signature verification based on the description of the signature envelope and the interior stroke distribution in polar and Cartesian coordinates. The features have been calculated using 16 bits fixed-point arithmetic and tested with different classifiers, such as hidden Markov models, support vector machines, and Euclidean distance classifiers. The experiments have shown promising results in the task of discriminating random and simple forgeries.

Hairong et al. [31], (2005), have proposed a novel off-line Chinese signature verification method based on support vector machines. The method used in this paper uses both static features and dynamic features. The static features include moment features and 16-direction distribution (an improvement on 4-direction distribution). The dynamic features include gray distribution and stroke width distribution. At last, a support vector machine is used to classify the signatures. The main steps of constructing a signature verification system are discussed and experiments on real data sets show that the average error rate can reach 5%, which is obviously satisfactory.

Ismail et al. [32], (2000), have proposed a system of two separate phases for signature recognition and verification is developed. A recognition technique is developed based on a multistage classifier and a combination of global and local features. New algorithms for signature verification based on fuzzy concepts are also described and tested. It is concluded from the experimental results that each of the proposed techniques performs well on different counts.

Guler et al. [33], (2008), have proposed a method for the automatic handwritten signature verification (AHSV). The method relies on global features that summarize different aspects of signature shape and dynamics of signature production. For designing the algorithm, they have tried to detect the signature without paying any attention to the thickness and size of it. The results have shown that the correctness of our algorithm detecting the signature is more acceptable. In this method, first the signature is pre-processed and the noise of sample signature is removed. Then, the signature is analyzed and specification of it is extracted and saved in a string for the comparison. At the end, using an adapted version of the dynamic time warping algorithm, signature is classified as an original or a forgery one.

Bhatia et al. [34], (2013), have proposed a signature verification system using Error Back Propagation Training Algorithm designed using Neural Network Toolbox of MATLAB to verify the signatures. The attractive features of this system are its low cost, low intrusion, good performance and use of an acceptable and natural biometric (the signature). A two-step method is proposed, which involves

identification of the signature in the first step followed by individual verification. Both the steps are carried out by Neural Networks trained using Error Back-Propagation Training Algorithm.

Baba [35], (1989), has stated that back-propagation may be the most widely-used method to adapt artificial neural networks for pattern classification. However, an important limitation of this method is that it sometimes falls into a local minimum of the error function. In this paper, the random optimization method of Matyas and its modified algorithm are used to learn the weights and parameters in a neural network. Research indicates that these algorithms can be successfully utilized in order to find the global minimum of error function of neural networks.

Muramatsu et al. [36], (2006), have proposed a new algorithm that performs a Monte Carlo based Bayesian scheme for digital signature verification. It has been divided into two phases namely learning phase and the testing phase, in the learning phase, semi-parametric models are trained using the Markov Chain Monte Carlo (MCMC) technique to draw posterior samples of the parameters involved, while in the testing phase the signatures are tested against the test data to validate the model accuracy. The proposed algorithm achieved an EER of 1.2% against the MCYT signature corpus where random forgeries are used for learning and skilled forgeries are used for evaluation

Armand et al. [37], (2007), have proposed a novel methodology by testing and validating the efficiency of the enhanced version of the MDF feature extractor for signature verification. Six-fold cross validation was performed to investigate new feature values of MDF, tests were performed, assessing the impact on the verification rate of the signatures. Two different neural classifiers were used and two methodologies for verification were applied. The merging of the MDF technology with the new features gave very encouraging results as the result of the study.

Agam [38],(2007), has proposed a curve warping approach that is used for reducing the variability associated with matching signatures. Existing techniques, uses methodology which uses 1-D parameterisation in the novel approach, is not just limited to 1-D approach. In this paper, they propose a novel approach for solving the curve correspondence problem that is not limited by the requirement of 1-D parametrization. The proposed approach utilizes particle dynamics and minimizes a cost function through an iterative solution of a system of first-order ordinary differential equations. The proposed approach is, therefore, capable of handling complex curves which was not possible by the previous approaches using the 1-D parameterisation. The proposed approach is evaluated by using the real world signed documents.

Batista et al. [39], (2010), have proposed that the neural and statistical classifiers employed in off-line signature verification (SV) systems are often designed from limited and unbalanced training data. In this article, an approach based on the combination of discrete Hidden Markov Models (HMMs) in the ROC space is proposed to improve the performance of these systems. By training an ensemble of user-specific HMMs with different numbers of states and different codebook sizes, and then combining these models in the ROC space, it is possible to construct a composite ROC curve that provides a more accurate estimation of system performance. Moreover, in testing mode, the corresponding operating points—which may be selected dynamically according to the risk associated with input samples—can significantly reduce the error rates. Experiments performed by using a real-world off-line SV database, with random, simple and skilled forgeries, indicate that the multi-hypothesis approach can reduce the average error rates by more than 17%, as well as the number of HMM states by 48%.

Nanni et. al. [40], (2008), have proposed an on-line signature verification system based on local information and on a one-class classifier, the Linear Programming Descriptor classifier (LPD), is presented. The information is extracted as time functions of various dynamic properties of the signatures, then the discrete 1-D wavelet transform (WT) is performed on these features. The Discrete Cosine Transform (DCT) is used to reduce the approximation coefficient vector obtained by WT to a feature vector of a given dimension. The Linear Programming Descriptor classifier is trained using the DCT coefficients. Results using all the 5000 signatures from the 100 subjects of the SUBCORPUS-100 MCYT Bimodal Biometric Database are presented, yielding remarkable performance improvement both with Random and Skilled Forgeries.

3.2 BACKGROUND

Signature forgery in legal documents, bank checks, doctors prescriptions can lead to huge consequences. In this respect signature verification is an important application in the field of biometrics. Bio-metrics measure human behaviors and are used in constructing recognition systems. Such recognition systems are useful for authorization and achieving a high degree of security.

3.3 MOTIVATION

Although signatures forgery are often manually detected by experts, still high accuracy is not always achieved. There are many difficulties in the manual forgery detection due to variations in handwriting style and professionalism of forgers. Recognizing dissimilarity between genuine and forged signatures needs skilled professionals. Automatic recognition systems can play an effective role in verifying signatures with high accuracy and in differentiating between genuine and forged signatures.

4.0 METHODOLOGY

Convolution Neural Network

A convolutional neural network (CNN) may be a multi-layer neural network with a deep supervised learning architecture that is proverbial to possess the potential of extracting features for classification by itself. CNN is composed of two parts: an automatic feature extractor and a trainable classifier. The feature extractor extracts feature from input data via two operations: convolution filtering and downsampling.

Convolution filtering: A picture will be seen as a matrix *I*, where *I*(*x*, *y*) is the brightness of the pixel located at coordinates (*x*, *y*). A convolution product is computed between the matrix *I* and a kernel matrix *K* which represents the sort of filter. *K* can be of size three by three or five by five. The result of this product will be the new brightness of the pixel (*x*, *y*).

The convolution product of product * for a kernel of size 3 × 3 is defined by:-

$$I * K = \begin{pmatrix} I(1,1) & I(1,2) & \dots & I(1,n) \\ \vdots & & I(x,y) & \vdots \\ I(m,1) & I(m,2) & \cdots & I(m,n) \end{pmatrix} * \begin{pmatrix} K(1,1) & K(1,2) & K(1,3) \\ K(2,1) & K(2,2) & K(2,3) \\ K(3,1) & K(3,2) & K(3,3) \end{pmatrix}$$

..... (1)

where,

$$I * K_{x,y} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I(x+i, y+j) * K(2+i, 2+j)$$

..... (2)

The choice of filter depends on the value of K.

Based on these features, the trainable classifier is trained using a back-propagation algorithm with a fully connected layer, and it produces the classification results.

The backpropagation algorithm performs learning on a *multilayer feed-forward* neural network. It iteratively learns a set of weights for prediction of the class label of tuples. A multilayer feed-forward neural network consists of an *input layer*, one or more *hidden layers*, and an *output layer*.

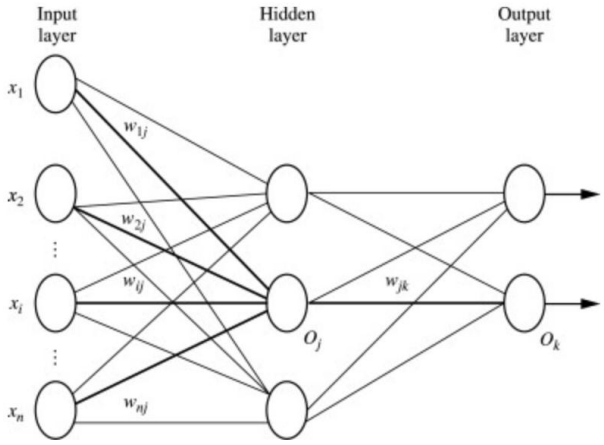


Fig 1: Multilayer feed-forward neural network

The Backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called the delta rule or gradient descent. The weights that minimize the error function is then considered to be a solution to the learning problem.

Working of Back-propagation algorithm:

- We first initialized some random value to 'W' and propagated forward.
- Then, we noticed that there is some error. To reduce that error, we propagated backwards and increased the value of 'W'.
- After that, we also noticed that the error has increased. We came to know that we can't increase the 'W' value.
- So, we again propagated backwards and we decreased 'W' value.
- Now, we noticed that the error has reduced.

The proposed method uses a CNN as a feature extractor and as a classifier. The Figure below shows the architecture of the proposed CNN-AE model. Similar to other deep neural networks, in CNN, the extraction process is black box and the exact characteristics of the features remain unknown. We assume that if a CNN is trained for classifying forged and genuine signatures, the trained CNN can extract effective features for distinguishing behavior characteristics of forgery, such as hesitation and delay before drawing the complicated part of a signature. Therefore, the output of the CNN feature extractor is used as a feature vector, defined as the S-vector (denoted by S in figures and equations). The S-vector is used as the input to an auto encoder for building the subject mode.

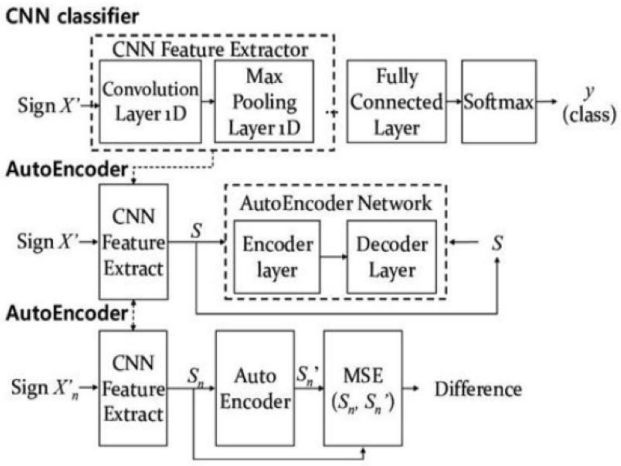


Fig 2: CNN classifier with AutoEncoder

An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise"

In the CNN feature extractor, the convolution layer uses X' as the input value. Each node ci of the convolution layer is defined in Equation below, where k is a kernel (also called filter) and l is the number of kernels used

$$c_i = \sum_{j=0}^{l} k_j \times x'_{i+j} \quad \text{..... (3)}$$

The output of ci is input to an activation function ReLU (Rectified Linear Activation function) defined in the equation below. The activation results constitute a convolution map. These procedures of the convolution layer are shown below:

Drawbacks of Convoluted Neural Networks

The very benefit of deep networks is that it represents multiple complex functions and learns features at different levels of abstraction, going to complex features that exist in the deepest of the layers from the edges which are a part of the much lower layers in general. But a huge barrier while using deep networks comes into play which is that the gradient decreases exponentially and quickly to zero as we back propagate from the final layer, back to the first layer. The gradient is a numeric calculation allowing us to know how to adjust the parameters of a network in such a way that its output deviation is minimized. In very rare cases it might explode or grow into very large values, suddenly and quickly.

Solution to the Problem

To overcome this, we end up requiring ResNet which acts as a shortcut or basically skips certain steps or connections which in turn allows the gradient to be directly back propagated thus reducing the chances that it will quickly fall to a very small value.

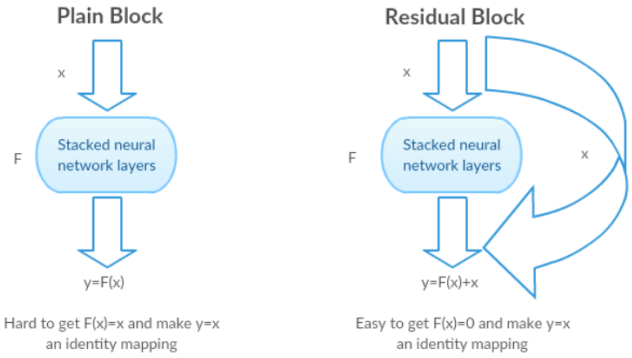


Fig 3: (left) Plain block and (right) Residual Block

Layers of CNN

The deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernels), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.

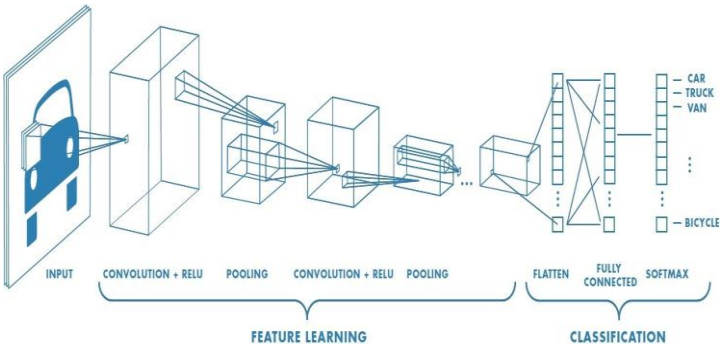


Fig 4: Neural network with many convolution layer

Convolution Layer: Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

Stride: Stride is the number of pixels shifted over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

Padding: Sometimes filters do not perfectly fit the input image. We have two options:

- Pad the picture with zeros (zero-padding) so that it fits
- Drop the part of the image where the filter did not fit. This is called valid padding which keeps only the valid part of the image.

Non Linearity (ReLU): ReLU stands for Rectified Linear Unit for a non-linear operation. The output is ***f(x) = max(0,x).*** Why ReLU is important : ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values

Pooling Layer: Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling is also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types:

- Max Pooling
- Average Pooling
- Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

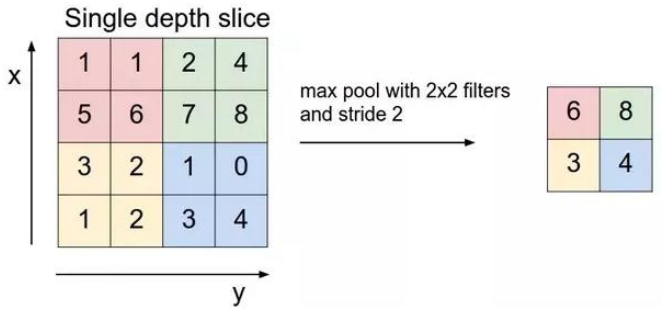


Fig 5: Max pooling

Fully Connected Layer: The layer we call as FC layer, we flattened our matrix into vectors and fed it into a fully connected layer like a neural network. With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax to classify the outputs.

Residual Network (ResNet)

Due to the phenomena of gradient decrease, the system functionality can be increased by implementing ResNet which is a type of neural network. The ReLU activation solves the problem of vanishing gradient that is due to sigmoid-like non-linearities (the gradient vanishes because of the flat regions of the sigmoid).

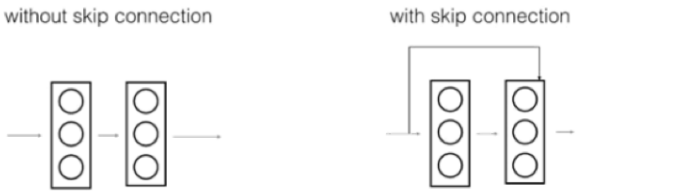


Figure 1(left) shows the working of deep Networks and (right) shows how residual network works.

Fig 9: (left) shows the working of deep Networks and (right) shows how residual network works

Residual networks help build deeper neural networks which is important because it helps in avoiding the degradation of the accuracy and the error rate of the handwritten signatures.

The weights of the matrix keep multiplying as you go through the layers. Batch normalization is a method used to make artificial neural networks faster and more stable through normalization of the input layer by re-centering and re-scaling, it effectively improves the speed and performance thus allowing the layer to train better. On the basis of how different or similar are the ResNet blocks, they are divided into two types:

I. The Identity Block

II. The Convolution Block

When input activation is of the same or similar dimension as the output activation, the standard block i.e. identity block is used.

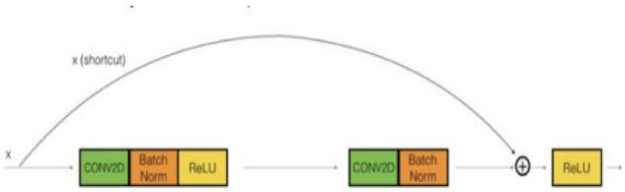


Figure 2 Identity Block

Fig 10: Identity block

Convolution block is used when there is a difference in the dimensions of the input and output activations and it is used to match that

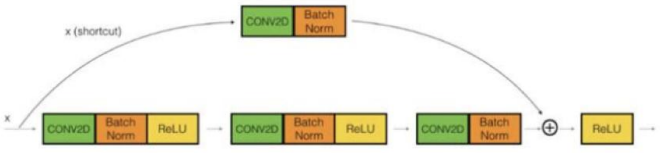


Fig 6: Convolutional block

ReLU is like a mathematical function and along with norm batch it will be able to skip over double-triple layers of the convoluted neural network. Deep neural network is good as it gives really good features like how a specific S was made or how the curve is. However, with deep neural networks, you lose gradient descent and it doesn't give good results. Due to this very reason we use the Resnet layer and it carries the weight of the matrix from the first to last without actually going through the layers in between. The activation function of a node defines the output of the node given an input or set of inputs.

ResNet-50 is a convolutional neural network that is 50 layers deep. We can load a pre trained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

In simple terms, the authors of the ResNet propose that fitting a residual mapping is much easier than fitting the actual mapping and thus apply it in all the layers. Another interesting point to note is the authors of ResNet are of the opinion that the more layers we stack, the model should not perform worse.

Phases of the system

I. Setting up the system

Importing libraries which are required

II. Preparing the dataset

We will first prepare the dataset and separate out the images:

We first divide the folder contents into the train and test directories.

Then, in each of the directories, create a separate directory for original signatures that contains only original signatures, and a separate directory for forged signatures. We'll do this manually for our project.

III. Building up the model

Data augmentation and adding data-augmentation parameters to generators.

Import the base model i.e. ResNet50 from tensorflow.keras.

Build and compile the model.

Fitting the model

IV. Testing the model

Now we'll test the model on the testing dataset to determine the testing loss and testing accuracy of the model.

V. Prediction

Now we'll run the model against two test cases, test case 1 being an original signature and test case 2 being a forged signature.

5.0 PLATFORM

Front end : Python

Python is used for the implementation of forgery detection in handwritten signatures. The abundance of python libraries and frameworks facilitates coding. Its readability is another boon. It reduces development time and makes collaboration easier.

Libraries used are :

I) Keras

II) Numpy

III) Pandas

IV) Scikit-learn

V) Matplot library

VI) Scipy

Back end : .png (images dataset)

Dataset source: https://www.kaggle.com/divyanshrai/handwritten-signatures?

This dataset consists of English signatures both forged and original which are already present in the labelled folder. From the above dataset, we chose 2149 images for the training set, 237 images for validation set and 274 images for testing set.

Sample Dataset :

6.0 SAMPLE CODE

```
import numpy as np

from keras import layers

from keras.layers import Input, Add, Dense, Activation, ZeroPadding2D, BatchNormalization, Flatten, Conv2D, AveragePooling2D, MaxPooling2D, GlobalMaxPooling2D

from keras.models import Model, load_model

from keras.preprocessing import image

from keras.preprocessing.image import ImageDataGenerator

from keras.utils import layer_utils

from keras.utils.data_utils import get_file

from keras.applications.imagenet_utils import preprocess_input

from keras.layers.convolutional import *

import pydot

from IPython.display import SVG

from keras.utils.vis_utils import model_to_dot

from keras.utils import plot_model

#from resnet_utils import *

from keras.initializers import glorot_uniform

import scipy.misc

from matplotlib.pyplot import imshow

from matplotlib import pyplot as plt

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

import tensorflow as tf

import itertools

import matplotlib.pyplot as plt

import cv2

# %matplotlib inline

import keras.backend as K

K.set_image_data_format('channels_last')

K.set_learning_phase(1)

from google.colab import drive

drive.mount('/content/drive')

from keras.applications.resnet50 import ResNet50

from keras.models import Sequential

from keras import optimizers

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

from keras.layers import Dropout , GlobalAveragePooling2D

base_model = ResNet50(weights="imagenet", include_top=False , input_shape=(64, 64,3))

x = base_model.output

x = Dropout(0.5)(x)

x = GlobalAveragePooling2D()(x)

x = Dense(128, activation='relu')(x)

x = BatchNormalization()(x)

predictions = Dense(2, activation='sigmoid')(x)
```

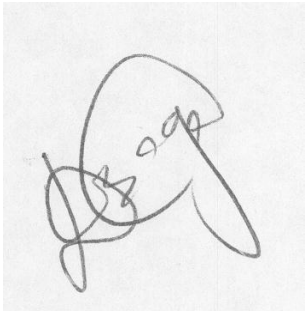
```
model = Model(inputs=base_model.input, outputs=predictions)

model.summary()

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

train_path = '/content/drive/My Drive/Signature Forgery Detection/EnglishTrain'

test_path = '/content/drive/My Drive/Signature Forgery Detection/EnglishTest'

train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, validation_split=0.1)

train_batches = train_datagen.flow_from_directory(train_path, target_size=(64,64), classes = ['full_forg','full_org'], class_mode = 'categorical', batch_size=32, subset = 'training',)

valid_batches = train_datagen.flow_from_directory(train_path, target_size=(64,64), classes = ['full_forg','full_org'], class_mode = 'categorical', batch_size=32, subset = 'validation')

test_batches = train_datagen.flow_from_directory(test_path, target_size=(64,64), classes = ['full_forg','full_org'], class_mode = 'categorical', batch_size=32)

model.fit(train_batches,steps_per_epoch=66, epochs=25, validation_data = valid_batches, validation_steps=1, verbose=1)

from tensorflow import keras

model = keras.models.load_model('/content/drive/My Drive/Signature Forgery Detection')

loss = model.evaluate_generator(test_batches, steps=8, verbose=1)

print ("Loss = " + str(loss[0]))

print ("Test Accuracy = " + str(loss[1]))

model.save('/content/drive/My Drive/Signature Forgery Detection')

test_batches.class_indices

import PIL

from PIL import Image

img_path = '/content/drive/My Drive/Signature Forgery Detection/EnglishTest/full_org/original_43_20.png'

img = image.load_img(img_path, target_size=(64, 64))

x = image.img_to_array(img)

x = x/255

x = np.expand_dims(x, axis=0)

result = model.predict(x)
```

7.0 TEST CASES

7.1 Test Case 1:

Input:

img_path = '/content/drive/My Drive/Signature Forgery Detection/EnglishTest/full_org/original_43_20.png'

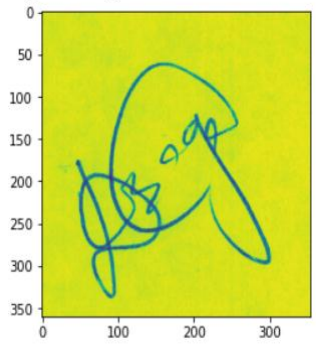


Output:

Image with its classification vector showing the sign is genuine not a forgery

```
Choose the signature image file:

Choose Files original_43_20.png
• original_43_20.png(image/png) - 71385 bytes, last modified: 6/18/2019 - 100% done
Saving original_43_20.png to original_43_20.png
class prediction vector [p(0), p(1)] =
[[3.0592675e-04 9.8420966e-01]]
```

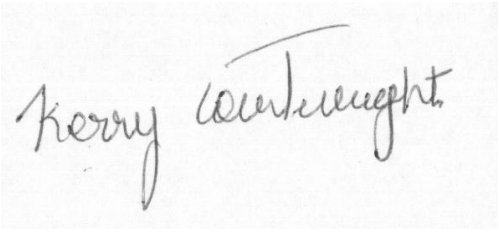
Original Signature



7.2 Test Case 2:

Input:

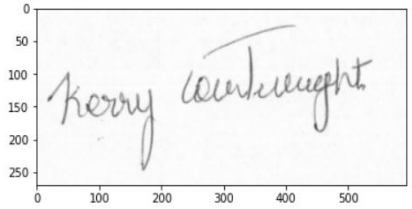
img_path = '/content/drive/My Drive/Signature Forgery Detection/EnglishTest/full_forg/forgeries_55_4.png'



Output:

Image with its classification vector showing the sign is a forgery

```
Choose the signature image file:

Choose Files  forgeries_55_4.png
• forgeries_55_4.png(image/png) - 126768 bytes, last modified: 6/18/2019 - 100% done
Saving forgeries_55_4.png to forgeries_55_4.png
class prediction vector [p(0), p(1)] =
[[0.9799446  0.00149396]]

Forged Signature
```



8.0 RESULT AND DISCUSSION

We train our model by using a dataset which consists of English signatures both forged and original which are already labelled and present in the respective folder.

We train our model for 25 epochs and the model thus generated as a loss function of 0.04492969438433647 which is negligible and an accuracy of 0.9921875

```
print ("Loss = " + str(loss[0]))
print ("Test Accuracy = " + str(loss[1]))
```

```
Loss = 0.04492969438433647
Test Accuracy = 0.9921875
```

As shown in the test cases input and output, we were correctly able to predict if a signature is original or forged. This makes the system efficient for signature verification. Thus our objective is fulfilled.

9.0 CONCLUSION AND FUTURE WORK

Handwritten signatures are very important in our social and legal life for verification and authentication. A signature can be accepted only if it is from the intended person. The probability of two signatures made by the same person being the same is very less. Many properties of the signature may vary even when two signatures are made by the same person. So, detecting a forgery becomes a challenging task.

Efficient methods of user verification are necessary in growing digitalization of various aspects of everyday life as well as new issues in offices and agencies. Parallel to new technology that is giving new possibilities a need for new and improved methods and algorithms is visible. The proposed method can be used as an effective signature verification system. The proposed method successfully made the offline signature verification with improvement in the efficiency and accuracy and easily detected the skilled forgeries. We have used python and its libraries, in conjunction with a solution based on Convolutional Neural Network (CNN) successfully for signature forgery detection. Future works include improving the model by lowering the Fault Rejection rate. Another promising work can be to combine offline and online signature verification systems which will make the system more robust as it will take both speed of execution and genuine visual signature into consideration so it will become harder to forge signatures. This can be developed in to apps or web page or can be used in security systems in public places such as ATMs, official government institutions, colleges, legal institutions, etc

10.0 REFERENCES

1. Hanmandlu, M., Yusof, M. H. M., & Madasu, V. K. (2005). Off-line Signature Verification And Forgery Detection Using Fuzzy Modelling. Pattern Recognition. 38(3):341–356.

2. Chai Quek & R.W. Zhou (2002). Antiforgery: A novel pseudo-outer product based fuzzy neural network driven signature verification system. Pattern Recognition Letters. 23(14): 1795-1816.

3. Madasu Hanmandlu, K.R. Murali Mohan, Sourav Chakraborty and Sumeer Goyal (2003). Unconstrained handwritten character recognition based on fuzzy logic. Pattern Recognition. 36(3):603-623.

4. Bin Xiao, Yang Wei, Xiuli Bi and Weisheng Li (2020). Image Splicing Forgery Detection Combining Coarse to Refined Convolutional Neural Network and Adaptive Clustering. Information Sciences. 511:172–191.

5. Ruiz, V., Linares, I., Sanchez, A., and Velez, J. F. (2020). Off-line Handwritten Signature Verification Using Compositional Synthetic Generation Of Signatures And Siamese Neural Networks. Neurocomputing. 374:30–41.

6. I.S.I Abuhaiba and Pervez Ahmed (1993). A fuzzy graph theoretic approach to recognize the totally unconstrained handwritten numerals. Pattern Recognition. 26(9):1335-1350.

7. C. L. Walker, M. Brown and Temple H. Fay (1988). Handprinted symbol recognition system. Pattern Recognition. 21(2):91-118.

8. Blanco-Gonzalo R., Sanchez-Reillo R., Liu-Jimenez, J. and Miguel-Hurtado O. (2014). Performance Evaluation Of Handwritten Signature Recognition In Mobile Environments. IET Biometrics. 3(3):139–146.

9. Zheru Chi, Jing Wu and Hong Yan (1995). Handwritten numeral recognition using self-organizing maps and fuzzy rules. Pattern Recognition. 28(1):59-66.

10. Diego Bertolini, Luiz Soares de Oliveira, Edson Justino, and Robert Sabourin (2010). Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. Pattern Recognition. 43(1):387-396.

11. Mustafa Berkay Yilmaz and Berrin Yanikoglu (2016). Score Level Fusion of Classifiers in Off-line Signature Verification. Information Fusion. 32: 109-119.

12. Robert Sabourin, Ginette Genest and F.J.Preteux,(1997). Off-line signature verification by local granulometric size distributions. Pattern Analysis and Machine Intelligence. 19(9):976-988.

13. Jun Cao, Majid Ahmadi and M. Shridhar (1995). Recognition of handwritten numerals with multiple feature and multistage classifier. Pattern Recognition. 28(2):153-160.

14. Hong Yan (1994). Handwritten digit recognition using an optimized nearest neighbor classifier. Pattern Recognition Letters. 15(2):207-211.

15. Ching Suen (1982). Distinctive features in automatic recognition of handprinted characters. Signal Processing. 4(2-3):193-207.

16. Toru Wakahara (1993). Toward robust handwritten character recognition. Pattern Recognition Letters. 14(4):345-354.

17. Drouhard, J.P Sabourin, Robert Godbout and Mario (1996). A neural network approach to off-line signature verification using directional PDF. Pattern Recognition. 29(3):415-424.

18. Guerbai, Yasmine Chibani, Youcef Hadjadji and Bilal (2015). The effective use of the one-class SVM classifier for handwritten signature verification based on writer-independent parameters. Pattern Recognition. 48(1):103–113.

19. Sigari, Mohamad-Hoseyn, Pourshahabi, Muhammad Pourreza and Hamid (2011). Offline Handwritten Signature Identification and Verification Using Multi-Resolution Gabor Wavelet. International Journal of Biometrics and Bioinformatics. 5(4):234-248.

20. Luiz G. Hafemann , Robert Sabourin and Luiz S. Oliveira (2017). Learning Features for Offline Handwritten Signature Verification Using Deep Convolutional Neural Networks. Pattern Recognition. 70:163-176.

21. Wen, Jing Fang, Bin Tang, Yuan Zhang and TaiPing (2009). Model-based signature verification with rotation invariant features. Pattern Recognition. 42(7):1458-1466.

22. Qi, Yingyong Hunt and Bobby (1994). Signature verification using global and grid features. Pattern Recognition. 27(12):1621-1629.

23. S Jerome Gideon, Anurag Kandulna, Aron Abhishek Kujur, A Diana and Kumudha Raimond (2018). Handwritten Signature Forgery Detection Using Convolutional Neural Networks. Procedia Computer Science. 143:978-987.

24. Hadjadji, Bilal Chibani, Youcef Nemmour and Hassiba (2017). An Efficient Open System for Offline Handwritten Signature Identification based on Curvelet Transform and One-Class Principal Component Analysis. Neurocomputing. 265(12):66-77.

25. Vargas-Bonilla, J.Ferrer-Ballester, Miguel Travieso, Carlos Alonso and Jesús (2011). Off-line signature verification based on grey level information using texture features. Pattern Recognition. 44(2):375-385.

26. Sabourin, Robert Plamondon, Réjean Beaumier and Louis (1994). Structural Interpretation of Handwritten Signature Images. International Journal of Pattern Recognition and Artificial Intelligence. 8(03):709-748.

27. Saeid Fazli & Shima Pouyan,(2015). High Performance Offline Signature Verification and Recognition Method using Neural Network. International Journal of advanced studies in Computer Science and Engineering. 4(6):9-13.

28. Shih Yin Ooi, Andrew Beng JinTeoh, Ying HanPang and Bee Yan Hiew (2016). Image-Based Handwritten Signature Verification Using Hybrid Methods of Discrete Radon Transform, Principal Component Analysis and Probabilistic Neural Network. Applied Soft Computing. 40(7): 274-282.

29. Ghandali, S., M. Moghaddam and M. Khosravi (2012). A new system for offline signature identification and verification. International Journal of Signal and Imaging Systems Engineering. 5(4):123-127.

30. Ferrer-Ballester, M. A., J. Alonso and C. Travieso-González (2005). Offline geometric parameters for automatic signature verification using fixed-point arithmetic. IEEE Transactions on Pattern Analysis and Machine Intelligence. 27(7):993-997.

31. Hairong Lv, Wenyuan Wang, Chong Wang, and Qing Zhuo (2005). Off-line Chinese Signature Verification Based on Support Vector Machines. Pattern Recognition Letter. 26(15):2390–2399.

32. Mohamed Ismail, Samia Omar, and Gad, S (2000). Off-line Arabic Signature Recognition and Verification. Pattern Recognition. 33(10):1727-1740.

33. Guler, Inan, Meghdadi and Majid (2008). A different approach to off-line handwritten signature verification using the optimal dynamic time warping algorithm. Digital Signal Processing. 18(6):940-950.

34. Bhatia, Shalini Bhatia, Pratik Nagpal, Dheeraj Nayak and Sandhya (2013). Online Signature Forgery Prevention. International Journal of Computer Applications. 75(13):21-29.

35. Baba N (1989). A new approach for finding the global minimum of error function of neural networks. Neural Networks. 2(5):367-373.

36. D. Muramatsu, M. Kondo, M. Sasaki, S. Tachibana and T. Matsumoto (2006). A Markov Chain Monte Carlo Algorithm for Bayesian Dynamic Signature Verification. IEEE Trans. on Information Security. 1(1):22-44.

37. Armand, S., Blumenstein, M. and Muthukkumarasamy, V. (2007). Off-line Signature Verification Using an Enhanced Modified Direction Feature with Single and Multi-classifier Approaches. IEEE Computational Intelligence Magazine. 2(2):18-25.

38. Gady Agam (2007). Warping-Based Offline Signature Recognition. IEEE Trans. on Information Security. 2(3): 430-437.

39. Batista, L., Granger, E. & Sabourin, R (2010). Improving performance of HMM-based off-line signature verification systems through a multi-hypothesis approach. International Journal on Document Analysis and Recognition.13(3): 33–47.

40. Nanni, Loris Lumini and Alessandra (2008). A Novel Local Online Signature Verification System. Pattern Recognition Letters. 29(4): 559-568.