International Journal for Multidisciplinary Research (IJFMR)

E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Signature Forgery Detection in Document Authentication Systems

Advancing Security: Machine Learning-Based

Abstract

Handwritten signatures play a vital role in our lives. From banks to institutions to organizations, signatures are a way of identifying a person. However, signings come with a lot challenges because any two signatures can look very similar with slight differences written the same person. Therefore, the identification of real and fake signatures is very difficult. To avoid similar identity related crimes committed in banks and many others companies, the counterfeit detection system is the solution to this problem along with the help concepts of machine learning and CNN. For better performance and time efficiency, Parallelization concepts are used in software implementation. This software can be used to verify signatures on many platforms such as loans, signing legal documents, applications signing, applications and much more.

Keywords: Crucial, Banks, Organization, Forgery, CNN, Forgery, Signature, Frauds.

Introduction

The aim of this project is to improve the detection of offline signature forgeries. In this project we analyzed and used machine learning concepts to classify, identify and also differentiate between the fake and the original signature. In this project implemented on Jupyter, we ensured the implementation of convolution neural network model using TensorFlow and created the model rather than traditional ways building a CNN model. We have uniquely implemented feature extraction based on various geometric factors of the signature image originating from the image dataset. Nowadays, a handwritten signature is one of the most widespread personal attributes proofs of identity, whether from the banking or business sector. People from lower society prefer to write their signatures in free handwriting due to lack of education and knowledge. Therefore, these types of signatures can be easily forged under certain circumstances. In this In this case, four types of fakes are possible.

Simulation Forgery

In which the forger has a sample of the signature to be forged. The quality of a simulation depends on how much the forger practices before attempting the actual forgery, the ability of the forger, and the forger's attention to detail in simulating the signature. Based on a forger's experience, known forgeries are classified as unskilled and skilled forgeries

Unknown/Random/Blind Forgery:

This is when the forger has no idea what the signature to be forged looks like. This is the easiest type of forgery to detect because it usually does not have the appearance of a genuine signature. This type of forgery will sometimes allow an examiner to identify who made the forgery based on the handwriting habits that are present in the forged signature.

Tracing:

The third type of forgery is tracing. Tracing can be done by holding the model document and the questioned document up to light and using a pen to trace the lines of the model signature onto the questioned document. A tracing can also be done by using a blunt stylus on the questioned document to create an impression of the model signature on the paper. This impression is then filled in with a pen to create the appearance of the model's signature. If the model signature used by the forger is not found, this type of forgery is sometimes difficult to detect from a photocopy.

Optical Transfer:

It is one in which a genuine signature is transferred onto a document by the use of a photocopier, scanner, facsimile machine, or photography. With this type of forgery, an examiner cannot positively identify a signature as genuine without having the Original for comparison.

Objective

The objective of the software is:

- to verify if a signature is forged or original.
- to ensure the authorized use of confidential information.
- to detect any impostor trying to access any important information.

Background:

For this system, the key concept will be the convolutional neural network (CNN). The CNN will be trained against a dataset containing many signatures such that it will have the skill to predict certain features and find out whether a forgery has been committed. or not. We aim to bring up software that verifies signatures and makes sure the software is more reliable, efficient, and 5% accurate than existing systems. Signatures vary with time when a person becomes old. There are certain factors which lead to changes in the signature, and these changes cannot be identified by ordinary people. One major concern that must be kept in mind is that the system should not be given to people randomly. Only those who have permission and are authorized to use this system should be taken care of since it is very confidential. This software can be used to validate signatures across many platforms, like loans, legal document signing, application signing, applying and a lot more. There are many organizations that have lost tremendous amounts of money due to a single forgery, and being able to detect even a single forgery can save money, time, and the reputation of the organization. The system will be run through a web page since it is more efficient and easier to use. Keeping in mind the people who have low technical skills or knowledge, the system should be easy to use and not require a lot of tasks to be done since it is timeconsuming.

Problem Description:

Online (dynamic) signature verification uses signatures that are captured by pressure-sensitive tablets that extract dynamic properties of a signature in addition to its shape. Dynamic features include the number and order of the strokes, Typical Signature Verification System, the overall speed of the signature, the pen pressure at each point, etc. and make the signature more unique and more difficult to forge. In an online signature verification system (Figure),

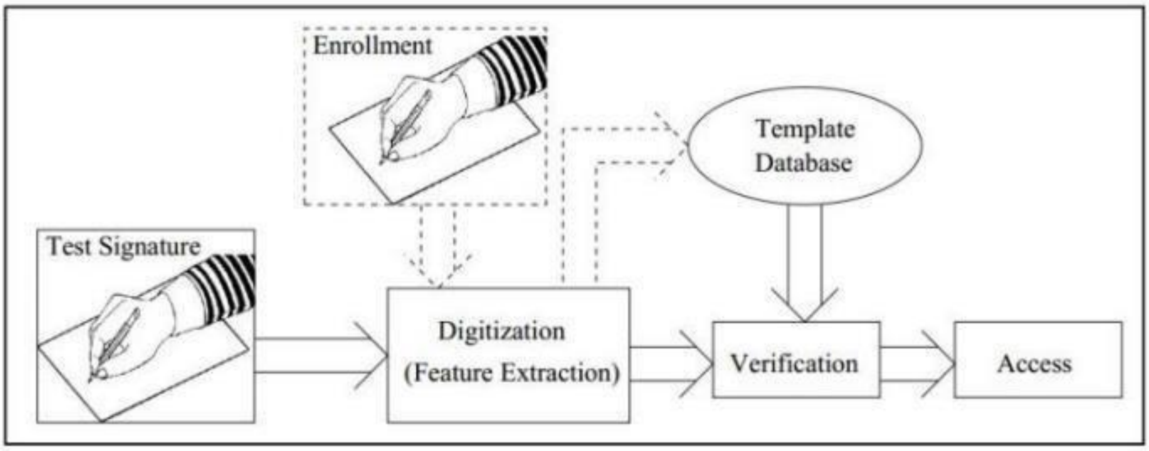


Figure : Typical Signature Verification System

Fig 1: Typical signature Verification System

The users are first enrolled by providing signature samples (reference signatures). When a user presents a test signature claiming to be an individual, this test signature is compared with the reference signatures for that individual. If the dissimilarity is above a certain threshold, the user is rejected. During verification, the test signature is compared to all the signatures in the reference set, resulting in several distance values. One must choose a method to combine these distance values into a single value representing the dissimilarity of the test signature to the reference set, and compare it to a threshold to decide. The single dissimilarity value can be obtained from the 6 minimum, maximum, or average of all the distance values. Typically, a verification system chooses one of these and discards the others. In evaluating the performance of a signature verification system, there are two important factors: the FRR of genuine signatures and the FAR of forgery signatures. As these two errors are inversely related, the EER where FAR equals FRR is often reported.

Section to describe or introduce new terms to the readers:

convolution neural network (CNN): A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. equal error rate (EER): The EER is the location on a ROC or DET curve where the false acceptance rate and false rejection rate are equal. Tensorflow: TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

LITERARY REVIEW:

According to recent studies, check fraud alone costs banks about $900 million annually, with 22% of all check fraud attributed to signature fraud. Clearly, with more than 27.5 billion checks written each year in the United States, visually comparing signatures with manual effort on the hundreds of millions of checks processed daily proves impractical.Myth: Authentic signatures of the same person will be exactly alike in all transactions

Reality: The physical act of signing requires brain, eye, arm, finger, muscle and nerve coordination. With all the factors at play, it's no wonder people don't sign exactly the same every time: some elements can be left out or altered. Personality, emotional state, health, age, conditions under which individual characters, space for signature and many other factors all affect the variation between signatures.

Types of signature forgery:

In real life, signature forgery is an event in which the forger focuses primarily on accuracy rather than fluency. The range of signature forgeries falls into the following three categories:

1. Random/Blind Forgery — Usually bears little or no resemblance to genuine signatures. This type of forgery occurs when the forger does not have access to an authentic signature.
2. Unqualified (tracing) forgery: The signature is traced and appears as a faint indentation on the sheet of paper below. This indentation can then be used as a guide for the signature.
3. Skilled forgery — Made by an offender who has access to one or more specimens of a genuine signature and can imitate it after much practice. A qualified forgery is the most difficult to verify of all forgeries.

The goal of an accurate verification system is to minimize both types of errors. Characteristic features: Let's understand the signature features for a human examiner to distinguish fraud from genuine. The following is a non-exhaustive list of static and dynamic characteristics used for signature verification:

- Shaky handwriting (static)
- Pen lift (dynamic)
- Retouch marks (static and dynamic)
- Letter proportions (static)
- Signature shape/size (static)
- Slope/Angle (static)
- Very close similarity between two or more signatures (static)
- Speed (dynamic)
- Pen pressure (dynamic)
- Patterns of pressure change (dynamic)
- Acceleration pattern (dynamic)
- Smoothness of curves (static)

Based on the verification environment and sampling conditions, not all features are available for analysis. After reading these articles, we were clear about the following topics and how to implement an optimal signature forgery detection model.

Our studies included:

Techniques such as four different CNN feature extractors have been created. Each CNN consisted of 2 convolution layers and 2 maximum pooling layers. Convolutional Neural Network (CNN) for implementing offline authentication methods, CEDAR Signature dataset for neural network training Convolutional Siamese Network (CSN), pooling layer, triple loss classification. I personally liked the CSN algorithm because it is the most optimal and efficient. Another important aspect was the assessment methods The number of hidden layers was 3 for the first paper and the number of nodes was 30, 20 and 30 for each layer. The training epoch was 500. The equal error rate (EER) is used as an evaluation metric for signature verification. All machine learning models were implemented using Theano library, a wellknown open-source machine learning library. MLP revealed a weakness in the qualified forgery test in the case of the second paper due to the fact that it is a 2-class model. AE and CNN_B-AE did not work well. This result shows that the S-vector generated by the trained CNN is more efficient than the raw signature data. CNN_D-AE showed the best result and CNN_AC- AE showed the second best accuracy. However, note that CNN_D-AE is not a practical model. In the third document, the Dataset contains 30 users, all of which have 15 signatures, the test starts with learning 0.00000 infrequently, and the margin is set to 0.2 0.2, the best test result is low loss and high accuracy, we end the analysis at step 180 , because we perceive the target result. So the accuracy is 84. There are two types of pool: maximum pool and average pool. Max Pooling returns the maximum value from the image split reported by the kernel. But average returns the average of all values from the part of the image that the kernel covers. To improve the accuracy of signature verification, some studies [8,9,10] use machine learning techniques, which are one of the most notable technologies. Buriro et al. [3] used a multilayer perceptron (MLP), a two-class classifier, to verify a finger-drawn signature with dynamic features involving finger and phone movements. Their method showed a verification accuracy of 94.8% for subject and other object classification. However, this technique does not represent the ability to distinguish forged signatures. Two-class classifiers such as MLP are at greater risk of misclassifying a forged signature as a subject because forged signatures closely resemble the subject [10,14]. Although the signature marked on the smartphone screen disappears immediately after verification, it is easy for an adversary to imitate the signature through shoulder surfing and smearing, which involves tracking the smudge left on the screen [11]. Therefore, it is necessary to improve the authentication for distinguishing forged signatures to provide secure services. An important issue is also the difference in time between the registration of a signature and the verification of a new signature. Unlike biometrics such as fingerprints and irises, behavioral patterns can change over time. Although [15] focused on the problem of this time difference, they worked on PIN verification and not dynamic finger-drawn signatures. In the first paper, we propose a new dynamic approach to fingerprint signature verification that provides better accuracy against forged signatures and time-delayed signatures. The proposed method uses two deep learning algorithms: a convolutional neural network (CNN) [16,17] for feature extraction and an autoencoder (AE) [18] as a classifier. CNNs are trained to distinguish forged signatures from genuine signatures, and the trained CNNs are used for feature extraction, not as a classifier. Specifically, the output of the CNN intermediate layer, which we call the S-vector, is used as input to the AE to create the subject model. Since CNN is known to be able to extract features for classification by itself [17], we hypothesize that a CNN trained for a specific purpose could extract features effective for that purpose. For example, a CNN trained on forged signatures can extract features that are common in forgery, such as hesitation and delay before drawing a complicated part of the signature. The proposed method uses AE as a classifier due to its high accuracy in solving single-class discrimination problems such as user authentication. Our previous work [19] and M. Fayyza et al. [10] showed that the one-class AE model is better at distinguishing simulated signatures than the two-class model. However, AE is also highly dependent on the accuracy of the input data [14]. The S-vector is valuable as an AE input and could subsequently lead to an increase in the accuracy of dynamic signature verification. Although the main problem is forgery signature discrimination, the proposed method should consider other problems, i.e.

time difference problem and classification of subject/others. They are also important issues for a signature verification service, and a single-purpose model could reduce the performance of other issues. The proposed method achieves better performance for time-varying signatures and subject/other classification using an experimental approach.

The main contributions of this post are as follows.

- To the best of our knowledge, the proposed scheme is the first CNN-AE model for hand-drawn signature verification in mobile environments.
- Experiments using real user signatures show that S-vector achieves better accuracy for dynamic signature verification. The proposed scheme reduces the same error rate (EER) by 1.1% (subject/other), 3.2% (time difference) and 13.7% (qualified forgery) compared to previous work.

This document is organized as follows. Section 2 introduced the proposed method. In Section 3, the proposed model is evaluated based on experimental results, where the CNN was trained with four different classification datasets to determine the most efficient S-vector. Section 4 concludes the paper. this includes methods for acquiring, processing, analyzing and understanding images and, in general, high- dimensional data from the real world to produce numerical or symbolic information, e.g. in the form of decisions [5]. Computer vision covers the basic technology of automated image analysis, which is used in many fields [6]. Computer vision as a scientific discipline deals with the theory behind artificial systems that extract information from images. Image data can take many forms, such as video sequences, views from multiple cameras, or multidimensional data from a medical scanner. As a technological discipline, computer vision tries to apply its theories and models to the construction of computer vision systems [5].

signatures can be used to identify and authenticate the subscriber. An automated verification process would allow banks and other financial institutions to significantly reduce check and money order forgeries, which represent large monetary losses every year. Reliable signature verification can be of great help in many other application areas such as law enforcement, industry, security control, and so on. Handwritten signatures appear on many types of documents, such as bank checks and credit slips, etc. [7][12]. A large number of such documents require automatic signature verification. A signature verification system requires high reliability.

Images are collected for training and stored in a database. Images are collected by scanning from a physical paper source. The database used is a self-created database that contains the signatures of three different people. The database consists of fifteen signatures belonging to each person, for a total of fortyfive signatures. More signatures can be easily added to the database and the number of signatures per person can be increased or decreased.

After the image has been pre-processed, various features are extracted from the image. The extracted features from each image are then stored in a MATLAB file. The following unique features are extracted from each image: [1] Height to Width Ratio: After cropping the image, the height to width ratio of the signature is calculated. [2] Centroid of Signature: The center of gravity or barycenter of the image is calculated. The centroid indicates the central point of the signature, which is a unique characteristic of the signature. The signature is divided vertically into two halves and the centroid of each half is calculated. [3] First derivative: The first derivative of the image matrix is calculated rowwise and columnwise. [4] Second derivatives: After the calculation of the first derivatives, the second derivatives of the image matrix are calculated by rows and columns. [5] Quadrant areas: The image is divided into four quadrants and then the area of the signature pixels in each quadrant is calculated. This area is the area of the signature strokes in that quadrant and does not include the background area. [6] COM Matrix: COM Matrix or Co-Occurrence Matrix refers to the distribution of co- occurring values at a given offset. It is used to measure texture in an image. Since our image is black and white after preprocessing, this means that the image matrix has values of either 0 or 1. It looks for a pattern distribution of these values and looks for where the patterns 00, 01, 11, and 10 occur. The co-occurrence matrix is also calculated for the signature. [7] Calculation of edge points: The number of edge points in the signature is calculated, which gives a distinctive characteristic of the signature. [8] Horizontal and vertical histogram: Each row and each column of the signature is traversed and the number of black pixels is calculated. The row and column with the maximum number of black pixels are recorded and used as a function. All of these properties provide unique signature characteristics and are used for signature classification.

A convolutional neural network (CNN) is a multi-layer neural network with a deep supervised learning architecture known to extract features for classification by itself [17]. A CNN consists of two parts: an automatic feature extractor and a trainable classifier. The feature extractor extracts features from the input data using two operations: convolutional filtering and downsampling. Based on these features, a trainable classifier is trained using a fully connected layer backpropagation algorithm to produce classification results.

The proposed method uses CNN only as a feature extractor, not as a classifier.

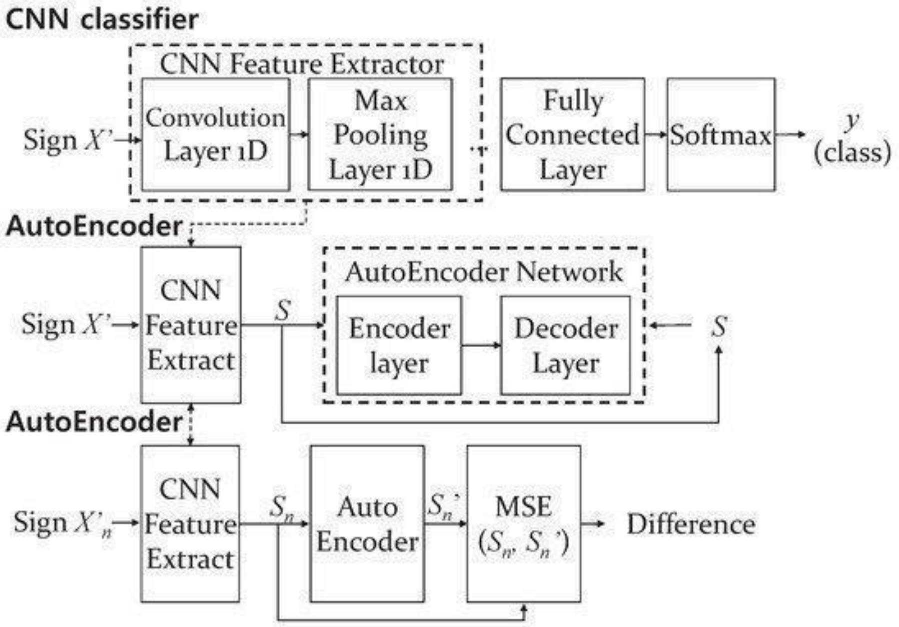


Fig 1.2: CNN Classifier

The *S-vector* extracted by the CNN feature extractor is used as the input of an AE. An AE is a type of deep neural network that has the same dimensions for input and output [14]. In the training phase, the AE is trained by using the same data (sample signatures) as input and output. In the test phase, the trained AE generates an output corresponding to an input (test signature). An AE can generate highly similar output for trained data patterns, whereas it does not for unfamiliar data. Therefore, a test signature is verified by a similarity comparison between the test signature and the output of the AE. Consequently, an AE is used in modeling a subject for authentication.

Paper-4 Abstract:

Hand signatures are very important in our social and legal life for verification and authentication. A signature can only be accepted if it is from the intended person. The probability that two signatures of the same person are the same is very less. Many characteristics of a signature can differ even if two signatures are created by the same person. Detecting a forgery thus becomes a challenging task.[4] In this paper, a convolutional neural network (CNN)-based solution is presented where a model is trained using a dataset of signatures and predictions are made whether a given signature is genuine or forged.

Method Proposed:

Keras library with Tensorflow backend is used to implement CNN. The directory of processed images is loaded and then we train the model with different training and testing ratios to evaluate the performance.[4]

Evaluation Methods:

The first raw RGB images are converted to grayscale and binarized. File management operations are then performed to split the images into batches based on the split ratio. Training and validation accuracies are plotted for all split ratios, and the split that gives the best results is considered.[4]

Paper-5 Abstract:

The signature capture and recognition system takes a signature image as input and trains the image by extracting various features and stores it in a database, then it will be compared with the original source signature using convolutional neural networks and recognize whether it is the original signature. Algorithms such as grayscale and binarization are used for feature extraction. Once the image is captured, it will be converted to a black and white image and then processed. This system needs to be trained very well in order to have better results. Sample signatures will be fed into the system for identification tests to maintain high accuracy in the system.[5]

Method Proposed:

The signature image is passed to image processing, where image enhancement, geometric transformation, etc. are applied. During feature extraction, local and global features are extracted. After extracting the feature, the CNN is applied to the image for comparison and then to the image with the image in the database. The image passes through a convolutional layer, a pooling layer and a fully connected layer. In this system, given a set of genuine signatures, the goal is to learn a model that can distinguish between genuine signatures and forgeries. The most common classification of forgeries in the literature is accidental forgery, where a person uses his signature to impersonate another person. Data acquisition Preprocessing Feature extraction Comparison using the CNN algorithm File management[5] **Evaluation Methods:**

First, a test signature is recognized with a given input training set using both CNN and Crest-Trough methods. Then, forgery detection algorithms (Harris Algorithmic followed by Surf Algorithm) are enforced on this classified image. The results from each algorithm are then compared. The signature capture and recognition system will take the signature image as input and train the image by extracting various features and store it in the database, then it will be compared with the original source signature using convolutional neural networks and recognize whether it is the original signature. Algorithms such as grayscale and binarization are used for feature extraction. Once the image is captured, it will be converted to a black and white image and then processed. This system needs to be trained very well in order to have better results. Sample signatures will be fed into the system for identification tests to maintain high accuracy in the system. Feature extraction is an important stage where the features of each signature are captured using a CNN algorithm. The idea behind this step is to identify every little detail of the signature. Subsequent identification of features and their proper extraction will lead to better or more accurate verification. A centralized database of correct customer signatures will be available. This particular database may be used by many systems that require customer information and signatures. This proposed system is focused on bank check signature verification system using artificial neural network. Signatures are verified based on parameters extracted from the signature using various image processing techniques. In detecting the exact person and providing more accuracy of signature verification for the implementation above, this paper uses convolutional neural networks to recognize and verify signatures of individuals.[5]

Paper-6 Abstract:

This paper presents an innovative approach for signature verification and forgery detection based on fuzzy modeling. The signature images are binarized and resized to a fixed size window and are then thinned. The thinned image is then partitioned into a fixed number of eight sub-images called boxes. Signature verification and forgery detection relate to the process of verifying signatures automatically and instantly to determine whether the signature is genuine or forged. There are two main types of signature verification: static and dynamic. Static, or off-line verification is the process of verifying an electronic or paper signature after it has been made, while dynamic or on-line verification takes place as a subject creates his signature on a digital tablet or a similar device.[2]

Method Proposed:

After the binarization and thinning of images, the thinned image is then partitioned into a fixed number of eight sub-images called boxes. This partition is done using the horizontal density approximation approach. Each sub- image is then further resized and again partitioned into twelve further sub-images using the uniform partitioning approach. The features of consideration are normalized vector angle (alpha) and distance (gamma) from each box. Each feature extracted from sample signatures gives rise to fuzzy sets. Since the choice of a proper fuzzification function is crucial for verification, the authors have devised a new fuzzification function with structural parameters, which is able to adapt to the variations in fuzzy sets. This function is employed to develop a complete forgery detection and verification system.

Since the main thrust here is to establish the genuineness of the signature thereby detecting the forgeries, we go in for fuzzy modeling of angle features. For the purpose of signature verification and detection of forgeries, we have employed the Takagi-Sugeno model. In this, we are following the same concept as outlined in for considering each feature as forming a fuzzy set over large samples. This is because the same feature exhibits variation in different samples giving rise to a fuzzy set. So, our attempt is to model the uncertainty through a fuzzy model such as the TS model.[2]

Evaluation Methods:

Inherent variation is used to judge the test signatures. For a particular feature, if the membership value lies within the range of variation, which is given by the difference of minimum and maximum thresholds, it is counted as 'true'. The total number of 'true' cases for a particular signature is divided by the total number of features to get the percentage. The skill-forged and unskilled forged signatures have corresponding figures of 88.5% and 82.3% respectively. The minimum limit or acceptable percentage for genuine signature is set at 91% referring to the output result of signature of signer. Signatures that have percentage less than 91% are treated as forged signatures. modified SPTA thinning algorithm- a proposed **algorithm** for **thinning** binary patterns, Pseudo- Bacterial Genetic Algorithm (PBGA) - The PBGA was proposed by the authors as a new approach combining a genetic algorithm (GA) with a local improvement mechanism inspired by a process in bacterial genetics. The proposed fuzzy modeling based on TS model discussed above has been applied on a signature database, developed in the Graphics Visualization & Games Development (GVGD) lab at the Multimedia University, Cybejaya, Malaysia. The efficacy of this system has been tested on a large database of signatures. The verification system is able to detect all types of forgeries: random, unskilled and skilled with utmost precision.[5]

From our work, we have an operating convolutional network that can successfully recognize signatures from 40 different individuals in 54% of the cases. Although the accuracy is not satisfactory for a real application, we attained a result that is significantly better than a random draw (which would have an accuracy of 2.5%). Based on these results, we consider our model to have learned a fair amount from the data provided. The low accuracy obtained in the training set relative to that of the test set is an evidence of overfitting. This result motivated us to try a few alternatives to reduce the gap between accuracies of training and test sets. These measures include: Optimizing the regularization term in the loss function, which did help to slightly improve the accuracy of the algorithm. We reached a point, however, where any increase of the hyperparameter lambda led to lower training and test set accuracies, and a decrease of it led to an increase in the gap between both accuracies[18]. Modifying the size of the network. Increasing the size of the network by expanding the number of neurons in the hidden layers only increased overfitting. On the other hand, decreasing the number of neurons in hidden layers did reduce overfitting, but at the cost of reducing the accuracy of the whole algorithm. We then chose to maintain the network structure. Stopping early. We also thought of stopping the training earlier (by epoch ~25, for example), to try to prevent the model from overfitting.[19] However, this did not have the desired effect and the accuracy of both the training and test sets decreased when attempting thi9s algorithm. Although the testing accuracy is not satisfactory for real signature recognition application, we consider our model to have learned a fair amount from the data provided and performs significantly better than a random draw. As a future work, a further exploration of the overfitting challenge identified would be key to improving the training accuracy of the CNN network. Additionally, additional modifications to the algorithm such as adding Batch Normalization or additional pre-processing steps could be explored, to investigate the impact on the model performance. Finally, memory limitations should be addressed to incorporate more training data to the model.[12][19]

Inherent variation is used to judge the test signatures. For a particular feature, if the membership value lies within the range of variation, which is given by the difference of minimum and maximum thresholds, it is counted as ‘true’. [4][5]The total number of ‘true’ cases for a particular signature is divided by the total number of features to get the percentage. The skill-forged and unskilled forged signatures have corresponding figures of 88.5% and 82.3% respectively. The minimum limit or acceptable percentage for genuine signature is set at 91% referring to the output result of signature of signer. Signatures that have percentage less than 91% are treated as forged signatures. [2][4]

Each signature will have a rule so we have as many rules as the number of features. The fuzzy set Ak is represented by the above exponential membership function where x, is the mean sigma is the variance of kth fuzzy set. The inclusion of parameters will help track the variations in the handwriting.

When sk = 1 and tk = -1, the membership function is devoid of structural parameters

$$\mu_k(x_k) = \exp-\left[\frac{(1-s_k)+s_k^2\left|x_k-\overline{x_k}\right|}{(1+t_k)+t_k^2\sigma_k^2}\right]$$

PROPOSED MODEL:-

Architecture:

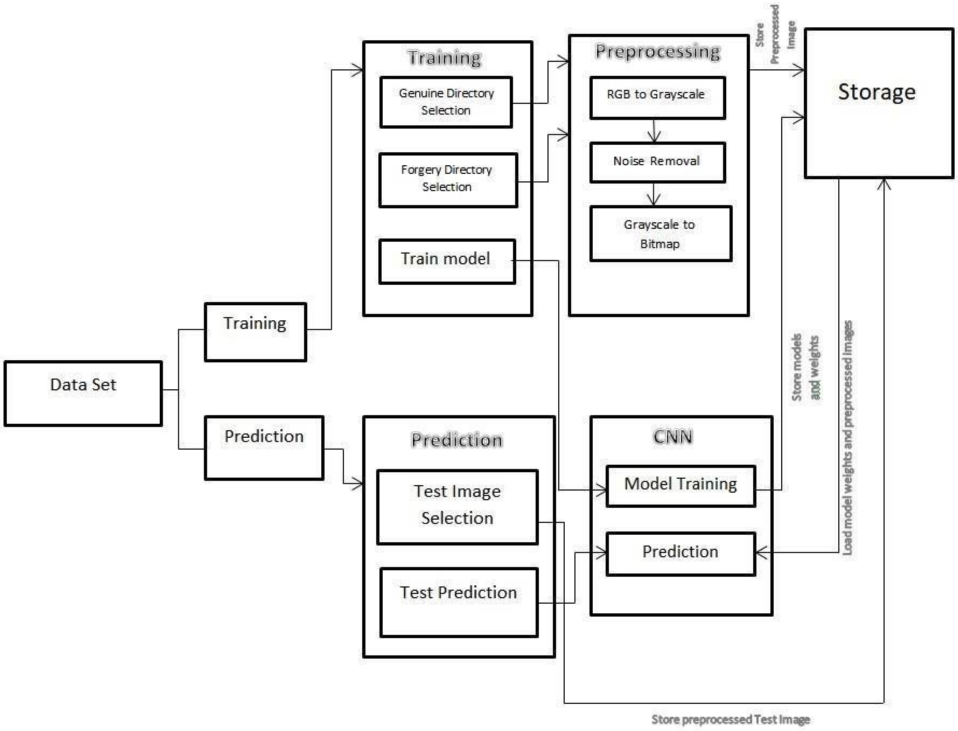


Fig 2: Architecture

In this work, signature images are preprocessed batch by batch and split into training and test sets based on the split ratio (which is chosen) as shown in Fig 2. This is done in Jupyter using functions from the image processing toolkit. After these signatures are preprocessed, they are stored in a file directory structure that the Keras Python library can work with. Then the CNN is implemented in Python using Keras with a TensorFlow backend to learn the patterns associated with the signatures. Then, the derived model was validated using accuracy and loss metrics to see how well the model fit the data. Finally, the model was tested using the signature from the challenge set to see if the predictions were correct. The figure shows a detailed architectural scheme of the implementation.

EVALUATION/EXECUTION:- DATASET:-

The dataset which has been used in this project work is a collection of 120 signatures, with 60 genuine and 60 forged signatures per subject. This dataset was carefully prepared by us, with one person making the originals and two others making the forgeries. All the images are in RGB format.

Forged:

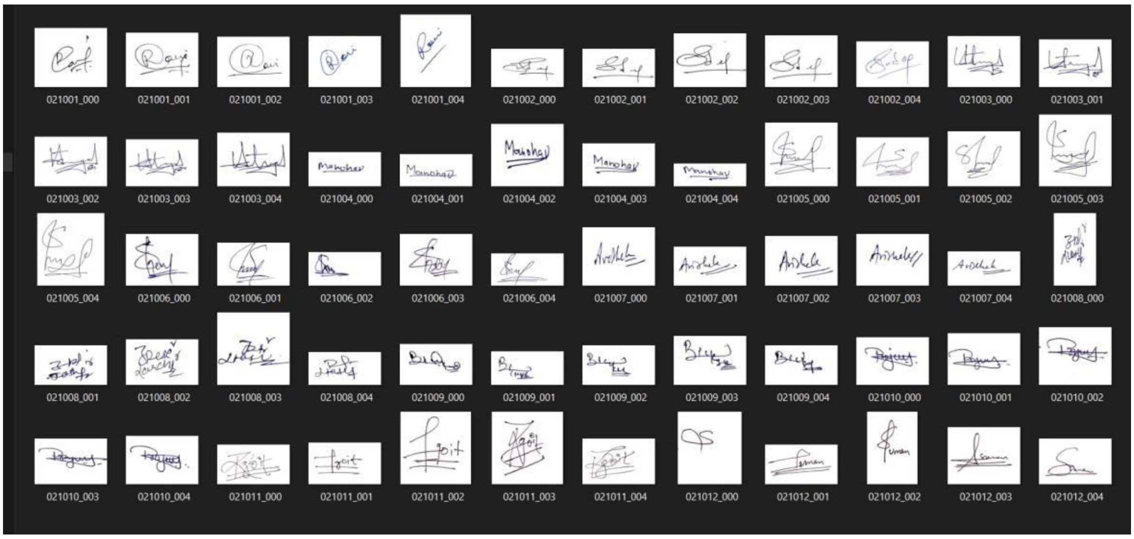


Fig 3: Forged Signatures

Real:

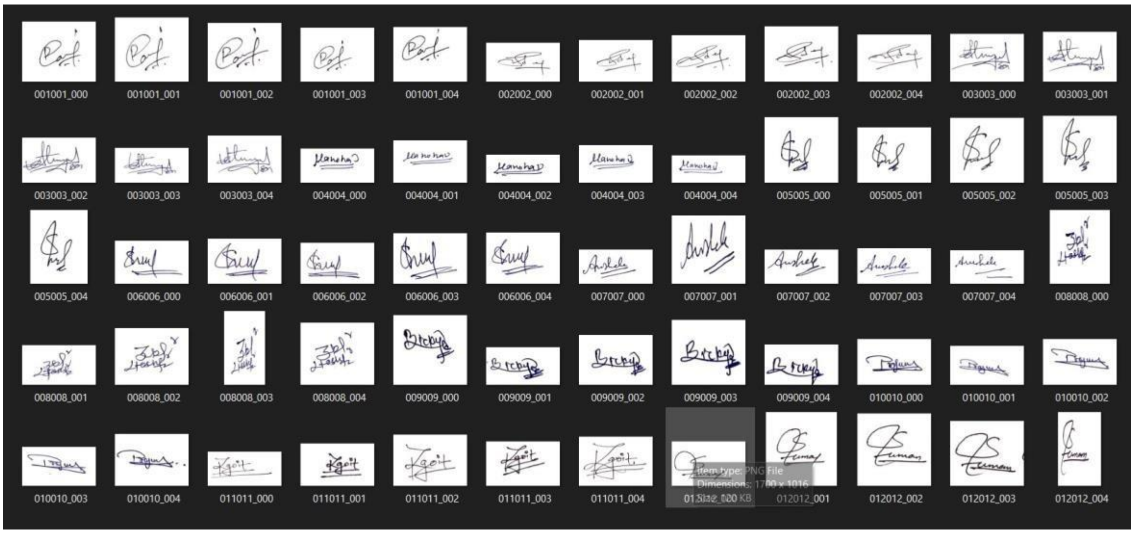


Fig 3.1: Real Signatures

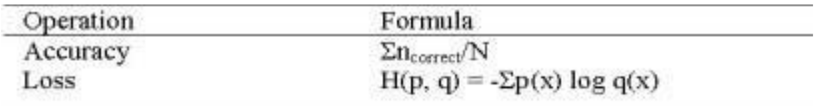
METHODOLOGY WITH THE DATASET:-

The handwritten signature is a behavioural biometric that is not based on any physiology characteristics of the individual signature but on the behaviour that changes over time. Since an individual's signature alters over time, the verification and authentication of the signature may take a long period of time, which includes the potential for the errors to be higher in some cases. An inconsistent signature leads to higher false rejection rates for an individual who did not sign in a consistent way.

Evaluation Measures:

First raw RGB images are converted to grayscale and binarized. Then file management operations are are carried out to split the images into batches based on the split ratio. Training and Validation accuracies are plotted for all the split ratios and the the split which gives the best results is considered.

The above accuracy and loss formulae are used to plot the accuracy and loss for the test splits. Accuracy is the total number of correct predictions divided by the total number of predictions. In loss, p is the true distribution and q is the coding distribution.



Operation	Formula
Accuracy	$\Sigma n_{correct}/N$
Loss	$H(p, q) = -\Sigma p(x) \log q(x)$

Evaluation methods section:

One of the most common evaluation technique used in signature forgery detection is plotting accuracy and validation graphs of various test split ratios and finding the best possible ratio. The dataset is divided into batches using several split ratio and accuracy and validation graphs are plotted and the split ratio with the best results is taken. The accuracy% and loss% are plotted against epoch. One unique way that we found was taking, a test signature is recognized with a given input training set using both CNN and Crest-Trough methods.

Custom variation is used to assess test signatures. For a particular function, if the membership value lies within the range of variation given by the difference of the minimum and maximum thresholds, it counts as "true". The total number of "real" occurrences for a particular signature is divided by the total number of features to get a percentage. Skilled forged and unskilled forged signatures have corresponding figures of 88.5% and 82.3%. The minimum limit or acceptable percentage for a genuine signature is set to 91% with respect to the output result of the signer's signature. Signatures that have a percentage of less than 91% are considered forged signatures[11].

Performance Measures:

In this work, the signature images are preprocessed in a batch manner and divided into training and test sets based on the split ratio (which is chosen). This is done in MatLab with functions from the Image Processing Toolkit. After these signatures are preprocessed, they are stored in a file directory structure that the keras python library can work with. Then the CNN was implemented in python using Keras with a TensorFlow backend to learn the patterns associated with the signatures. Then, the derived model was validated using accuracy and loss metrics to see how well the model fit the data. Finally, the model was tested using the signature from the challenge set to see if the ns predictions were correct.

In our implementation, the image goes through 3 convolutional and max pooling layers that alternate. When an image goes through the convolution process, a predefined number of feature maps are created, which are fed into a maximum pooling layer, which creates pooled feature maps from the feature maps received from the convolutional layer that precedes it. This pooled feature map is sent to the next convolutional layer and this process continues until we reach the third maximum pooling layer. The pooled feature map from the last maximum pooling layer is merged and sent to the fully connected layers. After several rounds of forward and back propagation, the model is trained and now a prediction can be made.

One of the most common evaluation technique used in signature forgery detection is plotting accuracy and validation graphs of various test split ratios and finding the best possible ratio. The dataset is divided into batches using several split ratio and accuracy and validation graphs are plotted and the split ratio with the best results is taken. The accuracy% and loss% are plotted against epoch. One unique way that we found was taking, a test signature is recognized with a given input training set using both CNN and Crest-Trough methods.

For a particular function, if the membership value lies within the range of variation given by the difference of the minimum and maximum thresholds, it counts as "true". The total number of "real" occurrences for a particular signature is divided by the total number of features to get a percentage. Skilled forged and unskilled forged signatures have corresponding figures of 88.5% and 82.3%. The minimum limit or acceptable percentage for a genuine signature is set to 91% with respect to the output result of the signer's signature. Signatures that have a percentage of less than 91% are considered forged signatures[11].

In our work, we will first convert raw RGB images to grayscale. We then add salt and pepper noise with a density of 0.01 and remove all the noise using the mean and median filters. We then binarize the images and store them appropriately. We then perform the required processing and file management operations to split the image batches based on the desired split ratio. After the models are built, accuracy and loss graphs are created.

We created models for vaious splits of data and plotted the training and validation accuracies to get an idea of the presence of any overfitting or underfitting