OFFLINE SIGNATURE VERIFICATION

A report submitted in partial
Fulfilment of the requirements for the degree of
Bachelor of Technology
in
Computer Science & Engineering

By

KUMAR SHUBHAM

(ROLL NO: 10400115090, REGISTRATION NO: 151040110090)
&

KUNAL KUMAR

(ROLL NO: 10400115091, REGISTRATION NO: 151040110091)

Under the supervision of

PROF. SOURAV SAHA
Assistant Professor
Department of Computer Science & Engineering
Institute of Engineering & Management

Department of Computer Science & Engineering
Institute of Engineering & Management
Gurkul, Y-12, Block –EP, Sector-V, Salt Lake Electronics Complex
Kolkata-700091, West Bengal, India

MAY, 2019

Affiliated to:

Maulana Abul Kalam Azad University of Technology, West Bengal
formerly known as West Bengal University of Technology
BF 142, Sector 1, Salt Lake City, Kolkata-700064

INSTITUTE
OF ENGINEERING & MANAGEMENT
श्रद्धावान लभते ज्ञानम्
Salt Lake Electronics Complex, Kolkata - 700091, WB, INDIA

Phone : (033) 2357 2969/2059/2995
(033) 2357 8189/8908/5389
Fax : 91 33 2357 8302
E-mail : director@iemcal.com
Website : www.iemcal.com

CERTIFICATE

TO WHOM IT MAY CONCERN

This is to certify that the project report titled **"OFFLINE SIGNATURE VERIFICATION"**, submitted by **Kumar Shubham, Roll No: 10400115090, Registration Number: 15104010090**, **Kunal Kumar, Roll No : 10400115091, Registration Number : 15104010091**, students of **Institute of Engineering & Management** in partial fulfilment of requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering**, is a bonafide work carried out under the supervision of **Prof. Sourav Saha** during the final year of the academic session of 2015 - 2019. The content of this report has not been submitted to any other university or institute for the award of any other degree.

It is further certified that the work is entirely original and the performance has been found to be satisfactory.

Prof. Sourav Saha

Assistant Professor

Department of Computer Science and Engineering Institute of Engineering & Management

----

Prof. (Dr.) Himadri Nath Saha

H.O.D.

Department of Computer Science and Engineering Institute of Engineering & Management

----

Prof. (Dr.) Amlan Kusum Nayak

Principal

Institute of Engineering & Management

INSTITUTE OF ENGINEERING & MANAGEMENT

DECLARATION

FOR NON-COMMITMENT OF PLAGIARISM

We, **Kumar Shubham**, **Kunal Kumar**, students of B.Tech. in the Department of Computer Science and Engineering, Institute of Engineering & Management have submitted the project report in partial fulfillment of the requirements to obtain the above noted degree. We declare that we have not committed plagiarism in any form or violated copy right while writing the report and have acknowledged the sources and/or the credit of other authors wherever applicable. If subsequently it is found that we have committed plagiarism or violated copyright, then the authority has full right to cancel/reject/revoke our degree.

Name of the Student: **KUMAR SHUBHAM**

Full Signature: ______________________________

Name of the Student: **KUNAL KUMAR**

Full Signature: ______________________________

Date: ______________________________

ABSTRACT

Human being authentication by offline handwritten signature biometric research has been increasing, especially in the last decade. The fact that the signature is widely used as a means of personal verification emphasizes the need for an automatic verification system because of the unfortunate side-effect of being easily abused by those who would feign the identification or intent of an individual. A great deal of work has been done in the area of signature verification over the past few decades.

Verification can be performed either Offline or Online based on the application. Online systems use dynamic information of a signature captured at the time the signature is made. Offline systems work on the scanned image of a signature. Verification process of an offline handwritten signature is not trivial task, because an individual rarely signs exactly the same signature whenever he/she signs, which is referred to as intra-user variability.

In this project we have implemented offline system of signature detection. Before implementing the algorithm, pre-processing of a scanned image is necessary to isolate the signature part and to remove any noise present. The objective of this project is proposing a feature vector of an offline handwritten signature by using an efficient algorithm as a strong feature extraction namely Histogram Orientation Gradient (HOG), in order to be passed into Artificial Neural Network for the recognition operation. An experiment has been conducted to estimate the accuracy and performance of the proposed algorithm by using local database, which has more than 48 genuine and 48 forged signature samples taken from 4 individuals. The result has given accuracy as 96.8% as successful rate coming from the error type as: False Accept Rate (FAR) is 3% and False Reject Rate (FRR) is 3.35%.

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide and mentor **Prof. Sourav saha** of the Department of Computer Science & Engineering, IEM Kolkata, for his wisdom, guidance and inspiration that helped us go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to **Prof. Dr. Satyajit Chakraborty**, Director, **Prof. Dr. Amlan Kusum Nayak**, Principal and **Prof. Dr. Himadri Nath Saha**, HOD of Computer Science & Engineering and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Kumar Shubham
Reg. No: 1510400110090 of 2015-2016
Dept. of Computer Science & Engineering
Institute of Engineering & Management

Kunal Kumar
Reg .No: 151040110091 of 2015-2016
Dept. of Computer Science & Engineering
Institute of Engineering & Management

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Traditional bank checks, bank credits, credit cards and various legal documents are an integral part of the modern economy. They are one of the primary mediums by which individuals and organizations transfer money and pay bills. Even today all these transactions especially financial require our signatures to be authenticated. The inevitable side-effect of signatures is that they can be exploited for the purpose of feigning a documents authenticity. Hence the need for research in efficient automated solutions for signature recognition and verification has increased in recent years to avoid being vulnerable to fraud.

Approaches to signature verification fall into two categories according to the acquisition of the data: On-line and Off-line. On-line data records the motion of the stylus while the signature is produced, and includes location, and possibly velocity, acceleration and pen pressure, as functions of time. Online systems use these data captured during acquisition. Online systems could be used in real time applications like credit cards transaction or resource access. While off-Line signature verification systems take as input the 2-D image of a signature. Offline systems are useful in automatic verification of signatures found on bank checks and documents. A robust system has to be designed which should not only be able to consider these factors but also detect various types of forgeries.

We approach the problem in two steps. Initially the scanned signature image is preprocessed to be suitable for extracting features. Then the preprocessed image is used to create model using the ANN.

CHAPTER 2
METHODOLOGY

2.1. Image Pre-processing and Features Extraction

We approach the problem in two steps. Initially, the scanned signature image is preprocessed to be suitable for creating model. Then, the preprocessed image is used to create model that can distinguish forged signatures from exact ones using the ANN approach.

2.2. Pre-processing

The signature is first captured and transformed into a format that can be processed by a computer. Now it's ready for pre-processing. In pre-processing stage, the RGB image of the signature is converted into grayscale and then to binary image. The pre-processing stage includes: Denoising, Color inversion, Filtering and Binarization.

2.2.1. Denoising

One of the fundamental challenges in the field of image processing and computer vision is image denoising, where the underlying goal is to estimate the original image by suppressing noise from a noise-contaminated version of the image. Image noise may be caused by different intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions which are often not possible to avoid in practical situations. Therefore, image denoising plays an important role in a wide range of applications such as image restoration, visual tracking, image registration, image segmentation, and image classification, where obtaining the original image content is crucial for strong performance. While many algorithms have been proposed for the purpose of image denoising, the problem of image noise suppression remains an open challenge, especially in situations where the images are acquired under poor conditions where the noise level is very high.

An alternative approach to the problem of image denoising based on data-adaptive stochastic optimization via Markov-Chain Monte Carlo sampling. By formulating the problem as a Bayesian optimization problem and taking a nonparametric stochastic strategy to solving this problem, such a Markov-Chain Monte Carlo denoising (MCMCD)[8] strategy dynamically adapts to the underlying image and noise statistics in a flexible manner to provide high denoising performance while maintaining relatively low computational complexity.

Examples of the results produced using the MCMCD strategy are shown below:

Fig.3. Signature Before Denoising

Fig.4. Signature After Denoising

2.2.2. Color Inversion

The true color image RGB is converted to the grayscale intensity image by eliminating the hue and saturation information while retaining the luminance.

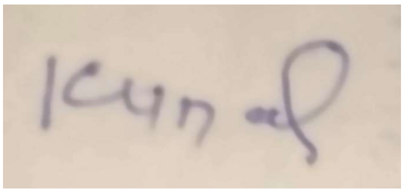

Fig.5. A sample signature to be processed

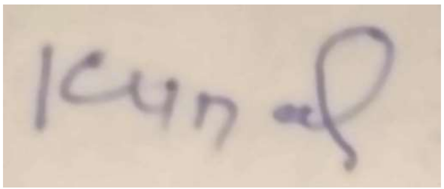

Fig.6. A Grayscale Intensity Image

2.2.3. Gray-scale Images

A gray-scale image is a data matrix whose values represent intensities within some range where each element of the matrix corresponds to one image pixel. It is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In fact a `gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image.

Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive gray-levels is significantly better than the gray-level resolving power of the human eye.

Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

2.2.4. Image Filtering and Binarization

Any image when resample is filtered by a low pass FIR filter. This is done to avoid aliasing. This aliasing occurs because of sampling the data at a rate lower than twice the largest frequency component of the data. So a low pass filter will remove the image high frequency components. And for this purpose the filter used. Now the grayscale image is segmented to get a binary image of objects. In a binary image, each pixel assumes one of only two discrete values: 1 or 0. A binary image is stored as a logical array.

Fig. 7. Binary Image interpreting the bit value of 0 as black and 1 as white

2.3. Feature extraction

We have experimented with two separate features: histogram of oriented gradients (HOG, [1]) relative to the dominant orientation and local binary patterns .

2.3.1 Histogram of Oriented Gradients (HOG)

Histogram Orientation Gradient (HOG) is used for feature shape representation, which was introduced by Dalal and Triggs at the CVPR conference in 2005 [10].

HOG is basically used for person detector, which stands for Histograms of Oriented Gradients. In this research, HOG has been adopted to be as a feature extraction technique to recognize and authenticate the signature image.

Theoretically, the HOG descriptor technique counts occurrences of gradient orientation in localized portions of an image or region of interest (ROI). The basic implementation of the HOG descriptor, which is illustrated in Figure 3, is as follows: First, dividing the image into small connected regions (cells), and for each region compute a histogram of gradient directions or edge orientations for the pixels within the cell, then, using the gradient orientation obtained. After that, discretizing each cell into angular bins, then, each cell's pixel contributes weighted gradient to its corresponding angular bin, then, adjacenting cells are grouped into blocks in the spatial region. This forms the basis for grouping and normalization of histograms, finally, normalized group of histograms represents the block histogram and the set of these block histograms represents the descriptor.

In other words, computation of the HOG descriptor requires the following basic configuration parameters, masks to compute derivatives and gradients, geometry of splitting an image into cells and grouping cells into a block, block overlapping and normalization parameters.

In this paper, the HOG is characterized as a block size is [22x22] pixels, the cell size is 128 with 9 bin histogram per cell. Accordingly, the overall feature vector length is 216 used to represent each signature image sample. Figure 4 depicts two offline handwritten signatures with a variety of cell size that has been implemented in this research and viewed to elaborate the HOG implementation on offline signature. It is clear that, when the cell size is low, the number of plotted gradient and directions extensively exist clearer than the high cell size. Gradually, by increasing the cell size number of HOG parameter, the directions and gradient will be decreased. In Figure 4, cell size has been plotted ranging from 16 until 128 depicting the effects of HOG on the offline signature images.

2.3.2 Gradient computation

The first step of calculation in many feature detectors in image pre-processing is to ensure normalized color and gamma values. As Dalal and Triggs point out, however, this step can be omitted in HOG descriptor computation, as the ensuing descriptor normalization essentially achieves the same result. Image pre-processing thus provides little impact on performance. Instead, the first step of calculation is the computation of the gradient values. The most common method is to apply the 1-D centered, point discrete derivative mask in one or both of the horizontal and vertical directions. Specifically, this method requires filtering the color or intensity data of the image with the following filter kernels:

Dalal and Triggs tested other, more complex masks, such as the 3x3 Sobel mask or diagonal masks, but these masks generally performed more poorly in detecting humans in images. They also experimented with Gaussian smoothing before applying the derivative mask, but similarly found that omission of any smoothing performed better in practice.[9]

2.3.3 Orientation binning

The second step of calculation is creating the cell histograms. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is “unsigned” or “signed”. Dalal and Triggs found that unsigned gradients used in conjunction with 9 histogram channels performed best in their human detection experiments. As for the vote weight, pixel contribution can either be the gradient magnitude itself, or some function of the magnitude. In tests, the gradient magnitude itself generally produces the best results. Other options for the vote weight could include the square root or square of the gradient magnitude, or some clipped version of the magnitude.[9]

2.3.4 Descriptor blocks

To account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then the concatenated vector of the components of the normalized cell histograms from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks. R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram. In the Dalal and Triggs human detection experiment, the optimal parameters were found to be four 8x8 pixels cells per block (16x16 pixels per block) with 9 histogram channels. Moreover, they found that some minor improvement in performance could be gained by applying a Gaussian spatial window within each block before tabulating histogram votes in order to weight pixels around the edge of the blocks less. The R-HOG blocks appear quite similar to the scale-invariant feature transform (SIFT) descriptors; however, despite their similar formation, R-HOG blocks are computed in dense grids at some single scale without orientation alignment, whereas SIFT descriptors are usually computed at sparse, scale-invariant key image points and are rotated to align orientation. In addition, the R-HOG blocks are used in conjunction to encode spatial form information, while SIFT descriptors are used singly.

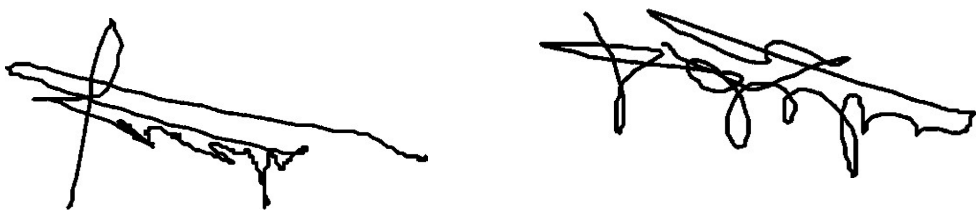

Figure 8: Offline signature samples as biometric.

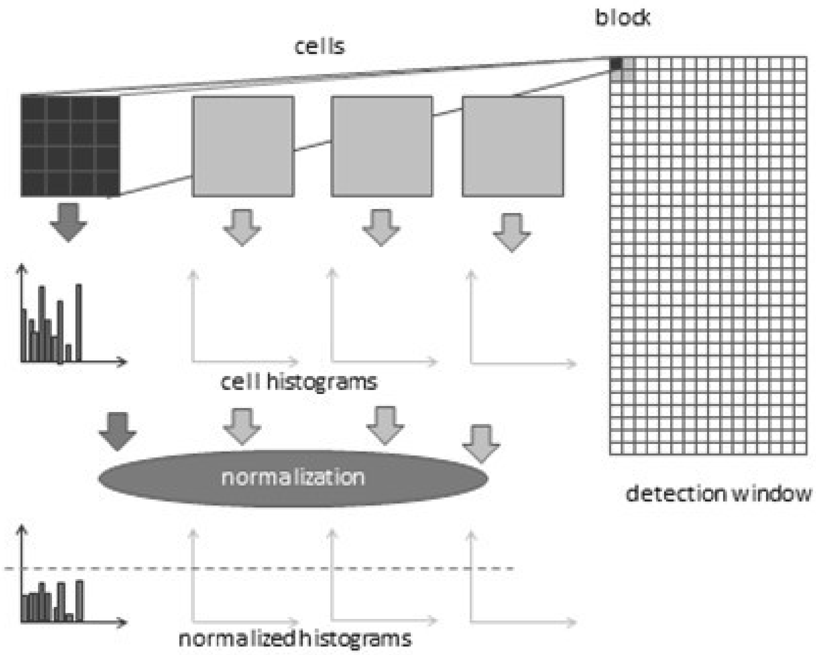

Figure 9: Demonstrates the HOG algorithm implementation.

Circular HOG blocks (C-HOG) can be found in two variants: those with a single, central cell and those with an angularly divided central cell. In addition, these C-HOG blocks can be described with four parameters: the number of angular and radial bins, the radius of the center bin, and the expansion factor for the radius of additional radial bins. Dalal and Triggs found that the two main variants provided equal performance, and that two radial bins with four angular bins, a center radius of 4 pixels, and an expansion factor of 2 provided the best performance in their experimentation (to achieve a good performance, at last use this configure). Also, Gaussian weighting provided no benefit when used in conjunction with the C-HOG blocks. C-HOG blocks appear similar to shape context descriptors, but differ strongly in that C-HOG blocks contain cells with several orientation channels, while shape contexts only make use of a single edge presence count in their formulation.[9]

2.3.5 Object recognition

HOG descriptors may be used for object recognition by providing them as features to a machine learning algorithm. Dalal and Triggs used HOG descriptors as features in a support vector machine (SVM)[9]; however, HOG descriptors are not tied to a specific machine learning algorithm.

Note that while complex features give more information, simpler features such as gradient orientation are more robust to normal variations found in a signature.

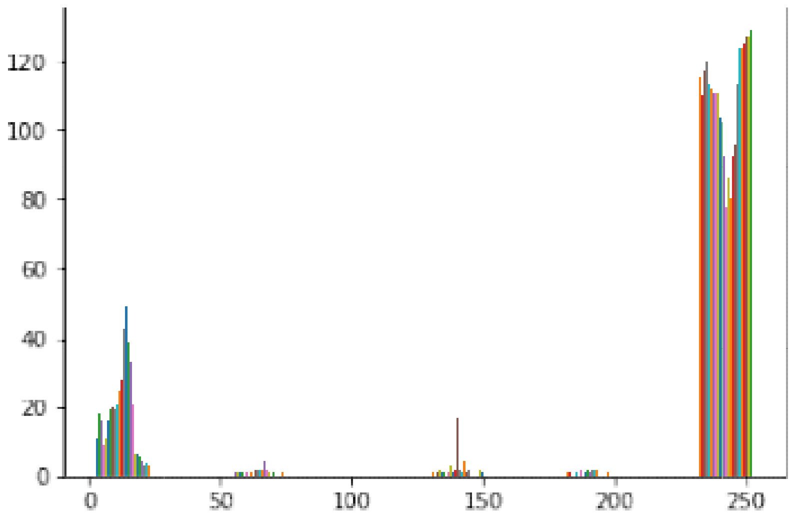

Fig. 10. Histogram of image pixel

2.4. Artificial Neural Network

Artificial neural networks (**ANN**) or **connectionist systems** are computing systems vaguely inspired by the biological neural networks that constitute animal brains. The neural network itself is not an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the results to identify cats in other images. They do this without any prior knowledge about cats, for example, that they have fur, tails, whiskers and cat-like faces. Instead, they automatically generate identifying characteristics from the learning material that they process.

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it.

In common ANN implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called 'edges'. Artificial neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Artificial neurons may have a threshold such that the signal is only sent if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

The original goal of the ANN approach was to solve problems in the same way that a human brain would. However, over time, attention moved to performing specific tasks, leading to deviations from biology. Artificial neural networks have been used on a variety of tasks, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games and medical diagnosis.

2.4.1 Basic Structure of ANNs

The idea of ANNs is based on the belief that working of human brain by making the right connections, can be imitated using silicon and wires as living neurons and dendrites.

The human brain is composed of 86 billion nerve cells called neurons. They are connected to other thousand cells by Axons. Stimuli from external environment or inputs from sensory organs are accepted by dendrites. These inputs create electric impulses, which quickly travel through the neural network. A neuron can then send the message to other neuron to handle the issue or does not send it forward.

ANNs are composed of multiple nodes, which imitate biological neurons of human brain. The neurons are connected by links and they interact with each other. The nodes can take input data and perform simple operations on the data. The result of these operations is passed to other neurons. The output at each node is called its activation or node value.

Each link is associated with weight. ANNs are capable of learning, which takes place by altering weight values. The following illustration shows a simple ANN −

2.4.2 Types of Artificial Neural Networks

There are two Artificial Neural Network topologies − FeedForward and Feedback.

FeedForward ANN

In this ANN, the information flow is unidirectional. A unit sends information to other unit from which it does not receive any information. There are no feedback loops. They are used in pattern generation/recognition/classification. They have fixed inputs and outputs.

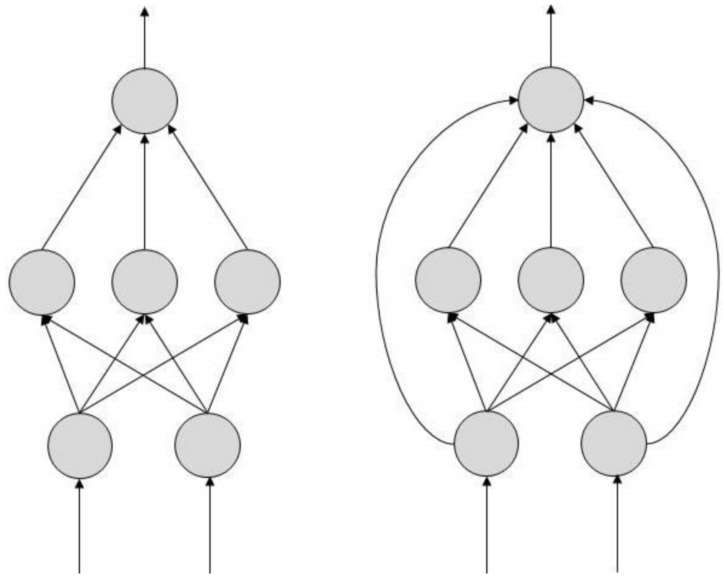

Fig. 11. FeedForward ANN

FeedBack ANN

Here, feedback loops are allowed. They are used in content addressable memories.

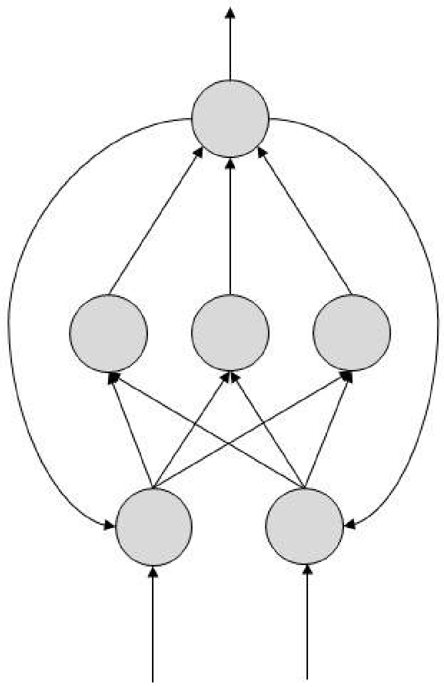

Fig. 12. FeedBack ANN

2.4.3 Working of ANNs

In the topology diagrams shown, each arrow represents a connection between two neurons and indicates the pathway for the flow of information. Each connection has a weight, an integer number that controls the signal between the two neurons.

If the network generates a “good or desired” output, there is no need to adjust the weights. However, if the network generates a “poor or undesired” output or an error, then the system alters the weights in order to improve subsequent results.

2.4.4 Machine Learning in ANNs

ANNs are capable of learning and they need to be trained. There are several learning strategies −

- **Supervised Learning** − It involves a teacher that is scholar than the ANN itself. For example, the teacher feeds some example data about which the teacher already knows the answers.

  For example, pattern recognizing. The ANN comes up with guesses while recognizing. Then the teacher provides the ANN with the answers. The network then compares it guesses with the teacher’s “correct” answers and makes adjustments according to errors.

- **Unsupervised Learning** − It is required when there is no example data set with known answers. For example, searching for a hidden pattern. In this case, clustering i.e. dividing a set of elements into groups according to some unknown pattern is carried out based on the existing data sets present.

- **Reinforcement Learning** − This strategy built on observation. The ANN makes a decision by observing its environment. If the observation is negative, the network adjusts its weights to be able to make a different required decision the next time.

2.4.5 Back Propagation Algorithm

It is the training or learning algorithm. It learns by example. If you submit to the algorithm the example of what you want the network to do, it changes the network’s weights so that it can produce desired output for a particular input on finishing the training.

Back Propagation networks are ideal for simple Pattern Recognition and Mapping Tasks.

2.4.6 Bayesian Networks (BN)

These are the graphical structures used to represent the probabilistic relationship among a set of random variables. Bayesian networks are also called Belief Networks or Bayes Nets. BNs reason about uncertain domain.

In these networks, each node represents a random variable with specific propositions. For example, in a medical diagnosis domain, the node Cancer represents the proposition that a patient has cancer.

The edges connecting the nodes represent probabilistic dependencies among those random variables. If out of two nodes, one is affecting the other then they must be directly connected in the directions of the effect. The strength of the relationship between variables is quantified by the probability associated with each node.

There is an only constraint on the arcs in a BN that you cannot return to a node simply by following directed arcs. Hence the BNs are called Directed Acyclic Graphs (DAGs).

BNs are capable of handling multivalued variables simultaneously. The BN variables are composed of two dimensions −

- Range of prepositions
- Probability assigned to each of the prepositions.

Consider a finite set $X = \{X_1, X_2, \ldots, X_n\}$ of discrete random variables, where each variable X_i may take values from a finite set, denoted by $Val(X_i)$. If there is a directed link from variable X_i to variable, X_j, then variable X_i will be a parent of variable X_j showing direct dependencies between the variables.

The structure of BN is ideal for combining prior knowledge and observed data. BN can be used to learn the causal relationships and understand various problem domains and to predict future events, even in case of missing data.

2.4.7 Applications of Neural Networks

They can perform tasks that are easy for a human but difficult for a machine −

- Aerospace − Autopilot aircrafts, aircraft fault detection.
- Automotive − Automobile guidance systems.
- Military − Weapon orientation and steering, target tracking, object discrimination, facial recognition, signal/image identification.
- Electronics − Code sequence prediction, IC chip layout, chip failure analysis, machine vision, voice synthesis.
- Financial − Real estate appraisal, loan advisor, mortgage screening, corporate bond rating, portfolio trading program, corporate financial analysis, currency value prediction, document readers, credit application evaluators.
- Industrial − Manufacturing process control, product design and analysis, quality inspection systems, welding quality analysis, paper quality prediction, chemical product design analysis, dynamic modeling of chemical process systems, machine maintenance analysis, project bidding, planning, and management.

- Medical − Cancer cell analysis, EEG and ECG analysis, prosthetic design, transplant time optimizer.
- Speech − Speech recognition, speech classification, text to speech conversion.
- Telecommunications − Image and data compression, automated information services, real-time spoken language translation.
- Transportation − Truck Brake system diagnosis, vehicle scheduling, routing systems.
- Software − Pattern Recognition in facial recognition, optical character recognition, etc.
- Time Series Prediction − ANNs are used to make predictions on stocks and natural calamities.
- Signal Processing − Neural networks can be trained to process an audio signal and filter it appropriately in the hearing aids.
- Control − ANNs are often used to make steering decisions of physical vehicles.
- Anomaly Detection − As ANNs are expert at recognizing patterns, they can also be trained to generate an output when something unusual occurs that misfits the pattern.

2.4.8. ANN Training

Artificial Neural Network or ANN[7] resembles the human brain in learning through training and data storage.

The ANN is created and trained through a given input/ target data training pattern. During the learning process, the neural network output is compared with the target value and a network weight correction via a learning algorithm is performed in such a way to minimize an error function between the two values..

The *mean-squared error* (MSE) is a commonly used error function which tries to minimize the average error between the network's output and the target value.

Twelve exact signatures and twelve forged signatures train the network and they were enough to give very good results in verification.

Table1 contains all the information related to the design of the neural network. Both original and forgery signatures are used for training the network. Testing signatures are also available.

Para meter	Value
Number of Layers	2
Number of neurons Output layer	1
Number of inputs	12
Learning rate (Constant)	Default
Transfer Function First Layer	Sigmoid
Transfer Function Second Layer	Sigmoid
Initial weights	Randomized
Initial biases	Randomized
Max number of epochs	1000
Error goal	0.0001
Number of patterns for original signature	12
Number of patterns for fake signature	12
Number of tested signatures	24
Number of tested original signatures	12
Number of tested fake signatures	12

Table.1. Neural Network Specifications

CHAPTER 3
EXPERIMENT & RESULTS

3.1 EXPERIMENT

The experiment rule is conducted on the offline signature template to measure the verification accuracy of the proposed offline signature verification method. The experiment is implemented according to the following steps:

1-Using signatures from the local database, the training matrix is built. The training matrix consists of signatures from 4 individuals, for each individual, 12 genuine samples and 12 forged samples are obtained (five of them are random forged samples and the rest are skilled forged samples).

2-The evaluation of the result produced by ANN is done by extracting the False Accept Rate (FAR) and the False Reject Rate (FRR) for each individual separately. The testing matrix is built similar to the way the training matrix was built.

3-In the training target (destination) of ANN, a sign +1 is assigned to the first 10 signature samples of the trained matrix, conversely, -1 is assigned to the second 10 signature samples of the training matrix to mark and train the ANN that the first 10 are genuine samples and the second 10 are forged samples.

4-The threshold that has been used is (zero).

5-FRR is computed by evaluating the resulting scores of the first 10 samples. If any sign of the first 10 samples is less than the threshold, False Rejection (FR) counter will be increased by one (FR = FR + 1) , since they are supposed to be as accepted (signs are larger than threshold) but they are wrongly rejected by the verifying system. On the other hand, if the results of the second 10 samples have signs more than the threshold, they are considered as False Accept (FA) and the counter will be incremented by one (FA =FA + 1) .

The FAR and FRR are computed as in (3) and (4) respectively:

$$\mathrm{FAR} = \frac{\mathrm{FA}}{10} \times 100\% \qquad (3)$$

$$\mathrm{FRR} = \frac{\mathrm{FR}}{10} \times 100\% \qquad (4)$$

6-The accuracy of each user is computed by using (5):

3.2 Results

Concerning the result of the experiment, Table 1 lists FAR, FRR and their average of the proposed recognition algorithm:

TABLE 2: THE PERFORMANCE OF THE PROPOSED ALGORITHM

FAR %	FRR %	Accuracy %
3	3.35	96.8

In order to consolidate the result of this paper, a comparison study has been done with set of benchmarks works published recently as in Table 2 compares the results of proposed scheme with some of the existing signature recognition algorithm.

As it is clear of Table 2, that FAR and FRR are less than the existing work, which has been used as a benchmark that recently published. In terms of FAR the proposed work is 3%, which is less than that in paper [10] as 5.05% and paper [11] as 4.9%. Regarding to the FRR error, the proposed work is 3,35%, which is less than the aforementioned article papers as FAR is 4.25% and 5.2% for the paper [10], [18] respectively.

It is worth to mention that, the main contribution of this paper is using HOG, which has been discovered as it is quite useful for offline biometric feature extraction by getting promising recognition rate in the future, since it has been confirmed that this research's result outperforms the state-of-the- art of offline signature recognition as shown in Table 2 comparison.

$$\mathrm{User}_{\mathrm{Accuracy}}\% = 100 - \frac{\mathrm{FAR}+\mathrm{FAR}}{2}$$

7-Then, to take into consideration all individuals in the local database, an average of the 200 individuals' accuracy is computed by using

$$\mathrm{AVR}_{\mathrm{Accuracy}} = \frac{1}{200}\sum_{u=1}^{200} \mathrm{User}_{\mathrm{Accuracy}}[u]$$

TABLE 3: THE PERFORMANCE OF THE PROPOSED ALGORITHM

Existing Techniques	FAR (%)	FRR (%)
Normalized Static Features and ANN Classification[17] / 2016	5.05	4.25
Normalized Weighted Coefficients[18] / 2016	4.9	5.2
Proposed Scheme	3	3.35

However, setting up small HOG parameters (cell size, block size, bin) will increase the length of the feature vector of the signature represented sample and that will cause much time consuming for the classifier and results low processing speed of the recognition. For this reason a balance maintaining of the HOG parameters selecting (not so small and not so large) will lead to the optimized and promising results in terms of both recognition accuracy and processing speed.

CHAPTER 4
CONCLUSION

Neural networks have demonstrated their success in many applications due to their ability to solve some problems with relative ease of use and the model-free property they enjoy. One of the main features, which can be attributed to ANN, is its ability to learn nonlinear problem offline with selective training, which can lead to sufficiently accurate response.

Application of Artificial Neural Network (ANN) to the above mentioned problem has attained increasing importance mainly due to the efficiency of present day computers. In addition, the times of simulation and testing in the ANN application are minimal. And the verification system based on ANN is able to learn different kinds of signature datasets.

Moreover, the use of large data is not required to show the capability of learning for this sort of problem, we have chosen only twelve genuine signatures and twelve forged ones for training, and we get very good results. However for real practice use, larger training data can increase the robustness of the system.

After training, the best classification accuracies were achieved. The classification ratio exceeds 93%. The algorithm we supported uses simple geometric features to characterize signatures that effectively serve to classify signature as exact or forged. The system is robust and can detect random, simple and semi-skilled forgeries. We have no clear idea about its performance in case of very skilled forgeries because we are not skillful imitating signatures to the extent being considered as skilled forgeries.

REFERENCES

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[2] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. pages A:582–585, 1994.

[3] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso. Off-line signature verification based on grey level information using texture features. *Pattern Recogn.*, 44:375–385, February 2011.

[4] M. A. Ferrer, J. B. Alonso, and C. M. Travieso. Offline geometric parameters for automatic signature verification using fixed point arithmetic. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):993–997, 2005.

[5] C. M. T. J. B. Francisco Vargas, Miguel A. Ferrer. Off-line handwritten signature gpds-960 corpus. In *IAPR 9'th International Conference on Document Analysis and Recognition*, pages 764–768, September 2007.

[6] J. K. Guo. *Forgery detection by local correspondence*. PhD thesis, College Park, MD, USA, 2000. Director-Rosenfeld, Azriel.

[7] Wikipedia. https://en.wikipedia.org/wiki/Artificial_neural_network

[8] Alexander Wong, Stochastic image denoising based on Markov-chain Monte Carlo sampling

[9] *"Histograms of Oriented Gradients for Human Detection"* ,http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[10] M. Trikha, M. Singhal, and M. Dutta, "Signature Verification using Normalized Static Features and Neural Network Classification," International Journal of Electrical and Computer Engineering, vol. 6, no. 6, pp. 2665, 2016.

[11] M. Singhal, M. Trikha, and M. Dutta, "Time Independent Signature Verification using Normalized Weighted Coefficients," International Journal of Electrical and Computer Engineering, vol. 6, no. 6, pp. 2658, 2016.

APPENDIX

UI DESIGN

Through the following interface, the user can select signature image of interest from available database. Then train the network with the content of this database and simulate it. The interface shows directly the 'matched ' or 'not matched' status which indicates if the signature is exact or forged.

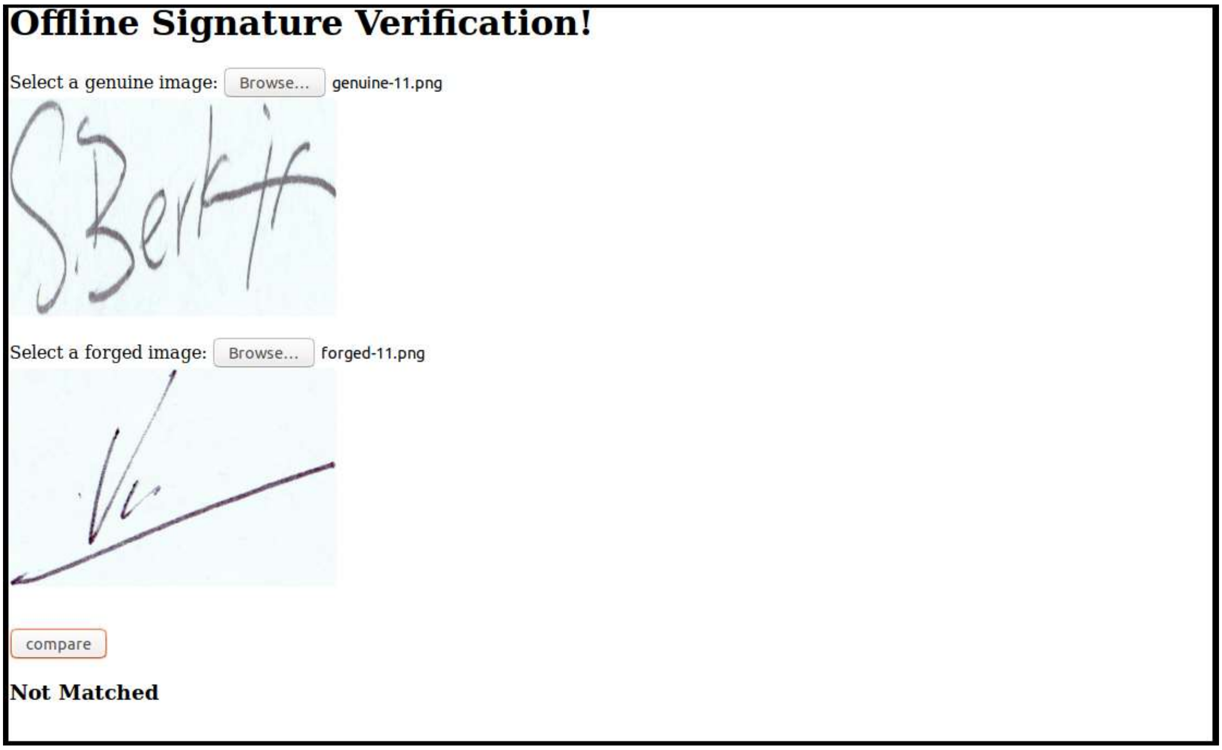

Fig 11 : Screenshot of UI

TESTING

The system has been tested for its accuracy and effectiveness on a data of about 24 signatures from 3 users which contains both their genuine and skilled forged signature sample counterparts.

All the samples of our data were pre-processed.
After pre-processing and training model, testing is done and the result is displayed.