OFFLINE SIGNATURE VERIFICATION SYSTEM USING ARTIFICIAL NEURAL NETWORKS

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

G.Rakesh(317126512132) **N.Santosh Kumar(317126512152)**

B.Pavan khagesh(317126512122) **B.Mani Shankar(318126512L31)**

S.Surya Sai Kiran(316126512168)

Under the guidance of

Mr.V.Vijay kumar Raju

M.TECH(Ph.D)

Assistant Professor

PRAGNANAM BRAHMA

ANITS

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)

Sangivalasa, Bheemili mandal, Visakhapatnam dist.(A.P)

2020-2021

DEPARTMENTOF ELECTRONICS AND COMMUNICATIONS ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGYAND SCIENCES
(Permanently Afiliated toAU,ApprovedbyAICTEandAccreditedbyNBA& NAACwith'A'Grade)
Sangivalasa, Bheemili Mandal, Visakhapatnam dist.(A.P)

PRAGNANAM BRAHMA
ANITS

CERTIFICATE

This is to certify that the project report entitled **"Offline Signature Verification System using Artificial Neural Networks(ANN)"** submitted by **G.Rakesh(317126512132), N.Santosh Kumar (317126512152), B.Pavan Khagesh(317126512122), B.Mani shankar (318126512L31), S.Surya Sai Kiran(316126512168)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** of **Andhra University**, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

Project Guide	**Head of the Department**
Mr.V.Vijay Kumar Raju	**Dr. V. Rajyalakshmi**
Assistant Professor	Professor & H.O.D
Department of E.C.E	Department of E.C.E
ANITS	ANITS

Assistant Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Visakhapatnam

Head of the Department
Department of E.C.E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mr.V.Vijay Kumar Raju,**Assistant Professor, Department of Electronics and Communication Engineering, ANITS, for her guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

G.Rakesh(317126512132)

N.Santosh Kumar(317126512152)

B.Pavan Khagesh(31712651212122)

B.Mani Shankar(318126512L31)

S.Surya Sai Kiran(316126512168)

LIST OF CONTENTS

ABSTRACT

In this paper we present offline signature verification system using Artificial neural networks (ANN).This scheme is based on the technique that applies preprocessing on the signatureacquired to get a binary image and then extracted feature points from it and maintain a feature vector. On the basis of these feature points, all computations are made.The feature vector obtained from theextracted features is converted tonormalised feature vector which is used to compare with the noramalised feature vector of incoming testing signature. Basedon the values obtained, the trained networkwill decide the appropriateness of the signature. Here the network is trained usingFeed Forward -Back Propagation alogorithm .The suggested scheme discriminates between original and forged signatures using artificial neural network (ANN) for training and verification of signatures.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

CHAPTER 1

1.1 Introduction:

Users are increasingly interacting with theirbanks through web sites and mobile applications for varying variety of tasks. Such tasks involve money transfer, balancecheck, loan applications, bill payments, andmany more. The transaction that is occurring through cheques has seen a steepdownfall over the decade by savings account holders but current account andother business venture accounts still make use of cheques as premiere source of transaction Such a transaction involving cheques is time consuming as it takes efforts form the receiver to visit banks physically, fill out redundant slips and bearthe queues at deposit corners. The wait doesnot end here the cheque is then sorted and scanned to be sent to the data centres of thebanks where it is being processed which takes about two or more days to avail the money in the receivers account. Recently, there are various banks that are issuing mobile applications in which you can submit snapshots of your cheque which youwant to be cleared which saves us from theeffort of physically visiting the banks but itlacks in transparency. These setbacks in theexisting system motivated us to work out the solution that we are proposing in this paper .

Signature has been a distinguishing featurefor person identification through ages. Even today, a rising number of transactions, particularly financial transactions, are approved by signatures, necessitating the development of methods for automatic signature verification if authenticity is now to be confirmed on a frequent basischeck, loan applications, bill payments, andmany more. The transaction that is occurring through cheques has seen a steep downfall over the decade by savings account holders but current account andother business venture accounts still make use of cheques as premiere source of transaction Such a transaction involving cheques is time consuming as it takes efforts form the receiver to visit banks physically, fill out redundant slips and bearthe queues at deposit corners. The wait doesnot end here the cheque is then sorted and scanned to be sent to the data centres of thebanks where it is being processed which takes about two or more days to avail the money in the receivers account. Recently, there are various banks that are issuing mobile applications in which you can submit snapshots of your cheque which youwant to be cleared which saves us from theeffort of physically visiting the banks but itlacks in transparency. These setbacks in theexisting system motivated us to work out the solution that we are proposing in this paper .Signature has been a distinguishing featurefor person identification through ages. Even today, a rising number of transactions, particularly financial transactions, are approved by signatures, necessitating the development of methods for automatic signature verification if authenticity is now to be confirmed on a frequent basis Approaches to signature verificationis of two categories based on the acquisitionof the data:1) Online and

2)Offline. Onlinedata records the motion of the stylus while the signature is produced, and includes location, and possibly velocity, accelerationrepetitive. Offline data is a 2D image of thesignature. Since there are no steady dynamic characteristics, processing offline is complicated.. Difficulty also lies in the fact that it is hard to segment signaturestrokes due to highly stylish and unconventional writing styles. The challenge is exacerbated by the non-repetitive character of signature variation caused by age, disease, geographic location, and maybe to some extent the person's emotional condition. When all of these factors are combined, they result in a lot of intrapersonal diversity. A robust system has been designed which should not only be able to consider these factors but also detect various types of forgeries .

1.2 Proposed System

The overall design of our signature recognition system follows the following steps:Signatureacquisition, Preprocessing, Feature extraction and Classification.Initially Offline signatures ofdifferent persons with multiple samples hastaken and considered as database.nearly 350 signatures samples has considered. Offline signatures are the signatures made on papers. This necessitates specifying the resolution, image type, and format for each image to be scanned. As a result, the initial step in any offline signature verification system is to scan signatures from papers. The signature sheet is given to the scanner, which produces a scanned image of the signature. During the pre-processing stage.The RGB picture of the signature is transformed to grayscale and subsequently to binary and pen pressure, as functions of time. Online systems use the informationcaptured during acquisition. These dynamiccharacteristics are specific to each individual and sufficiently stable as well as image. Later,Thinning is applied to make the signature lines as single stroke lines andany noise present in scanned images are removed by cropping ,thus making the signature image ready to extract features. Features available to extract in offline signatures can be either global features or texture featuresIn this system, the features extracted are Aspect ratio,Center of mass, Maximum Black pixels,Normalised Area,Tri Surface,Six Fold and Transition Feature. These extracted 2 features combined to form a normalised vector which is used to compare and there byclassify Signatures either genuine or forge.The normalised vector is given as input to the Neural Network which is trained using Feed Forward-Back Propagation algorithmand signatures are tested to classify whether signature is genuine or forged.

1.3 Ojective of the Project

The aim of off-line signature verification is to decide, whether a signature originates from a given signer based on the recorded image of the signature and a few images of the original signs of the signer. Signature is a special case of handwriting which includes special characters and flourishes. As many signs can be unreadable. They are a kind of artistic handwriting objects. However, a signature can be handled as an image, and hence, it can be recognized using computer vision. Signature recognition and verification involves two separate but strongly related tasks: I) Identification of the owner of signature, II) Whether the signature is original or forged.

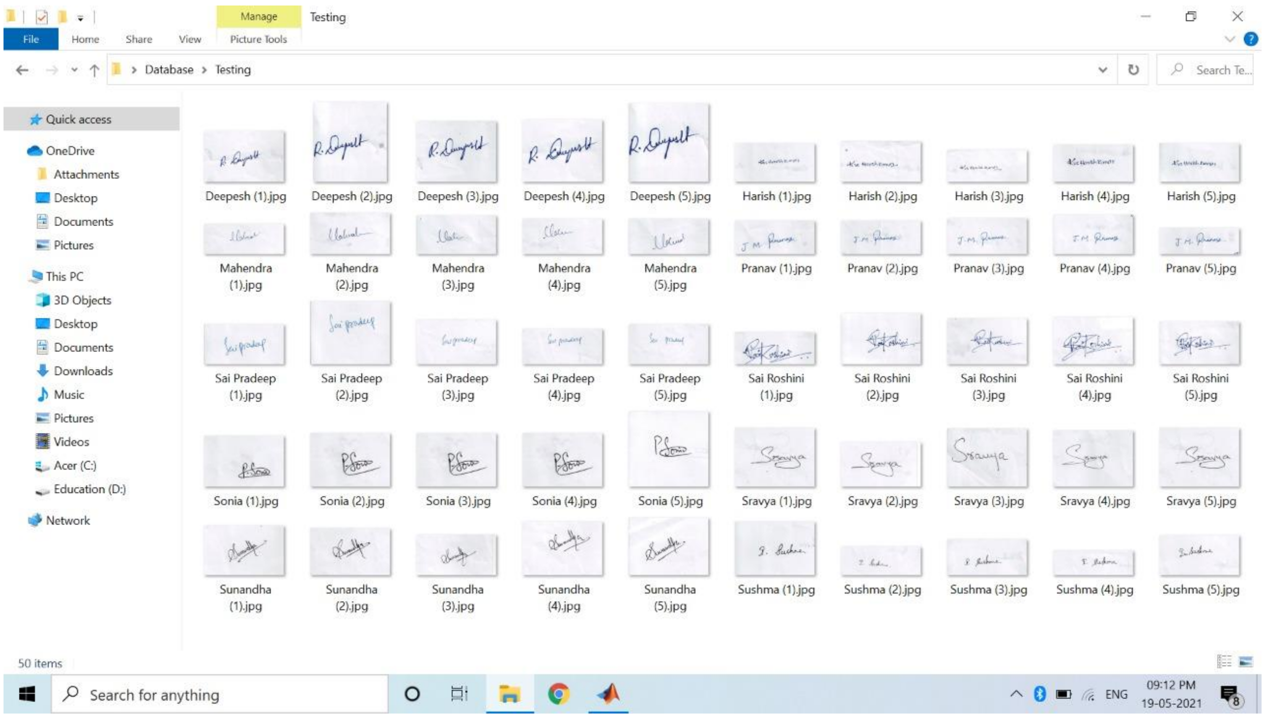


Fig 1.1 . Database used

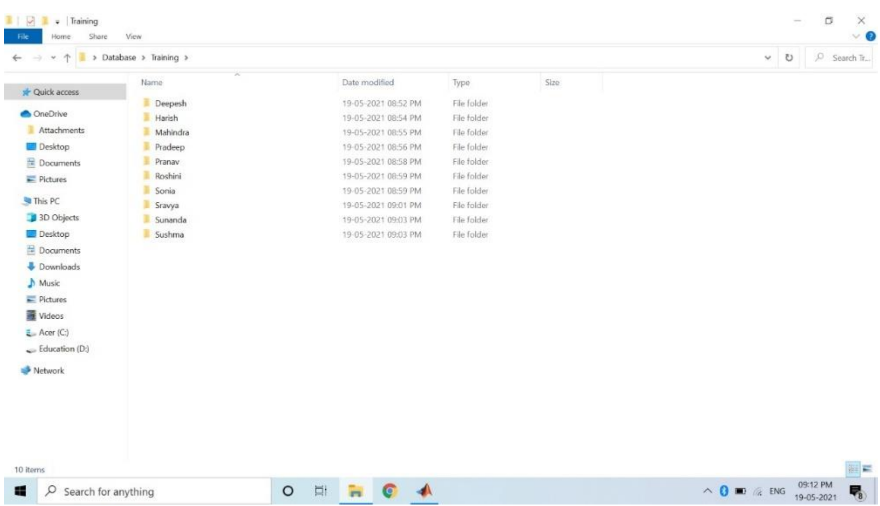


Fig 1.2 Signatures of different persons

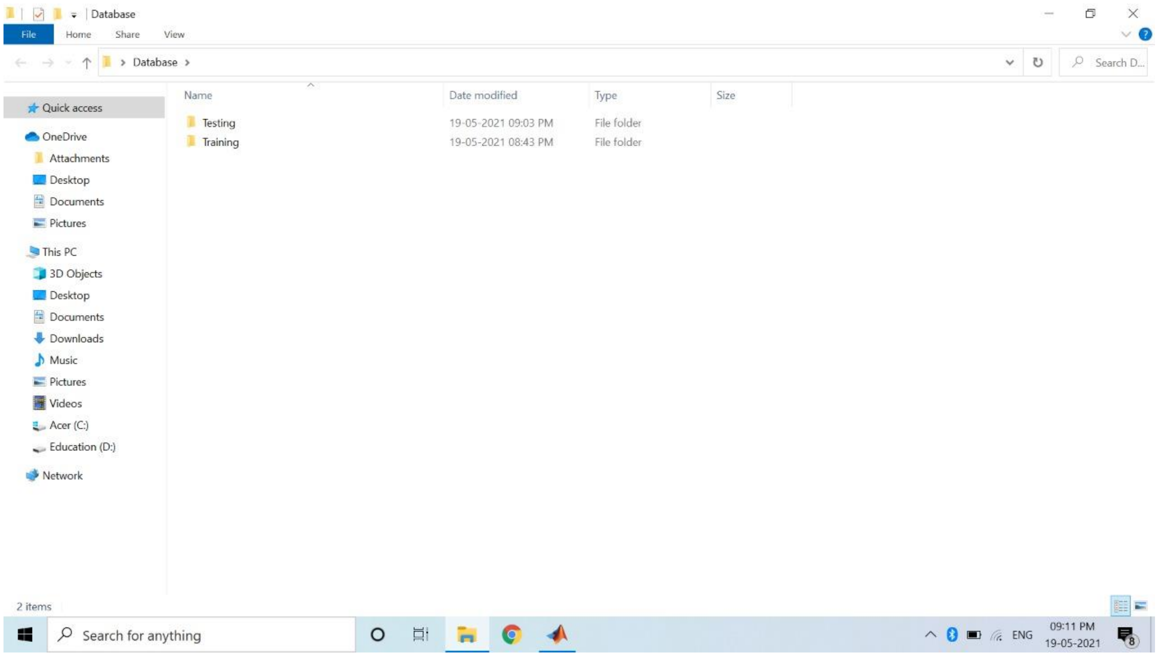


Fig 1.3 Data set used for Training and Testing

1.4 Motivation of the Project:

Signatures are composed of special characters and flourishes and therefore most of the time they can be unreadable.Wide number of applications - financial areas like banks, checks and legal documents are verified on the basis of account number and signature of the account holder.Various methods have already been introduced in this field but by far the texture method for feature extraction we have used has not been used for signatures

CHAPTER-2

DATABASE AND IMAGE ACQISATION

CHAPTER-2

Database and Image Acquistion

2.1 Introduction to Database:

Signature database plays a significant role in the process of signature verification. But there is no standard offline signature database for the researchers [1]. The major setback faced by the researchers in offline signature verification is the non-availability of a sufficiently large signature database with required quality. Because of privacy issues, not many people agree to make their signatures available to others and that too for practicing forgeries. Therefore, it is still not easy to develop a sufficiently large database with quality signatures. Some limited numbers of offline signature databases are available. But their sizes are smaller and they lack in quality. In pattern classification, there are two phases - (i) training and (ii) testing. In training phase, the signature classifier is modeled. Sufficient numbers of signature samples are required to model the classifier. The larger database represents a population properly and it helps in producing more reliable results. In most of the pattern recognition techniques, large datasets help in achieving better results. A small training set may result unsatisfactory verification performance. Lesser number of signature samples against high number of extracted features is a snag in signature verification [2]. Widely accepted and standard benchmark signature database is not yet available for offline signature verification. Therefore, we tried to develop our own signature databases for our experimentation. Database preparation includes the first two steps of offline signature verification - (1) Data Acquisition and (2) Preprocessing .

In this paper a framework for the modelling of neural networks and their integration into database systems is presented. This framework is based on the object-oriented approach, which is provided by many modern database systems. It allows for the easy and handy definition and administration of neural networks of predefined database types or the creation of new and specialized network types. To model neural network objects the generator approach of Smith [22] is used, which is extended by communication operators for the database environment. Different operator models are described and their properties discussed. Further we present a common framework for embedding neural networks into the based component of a knowledge-based database system.According to this, a neural net is a complex data value and can be stored as a normal object. The object-oriented approach seams (and in our opinion has proven) the most comfortable and natural design model for neural networks [10]. In the terms of object oriented database systems neural networks are treated generally as complex objects. These systems showed very valuable at handling and administrating such objects in different areas, as computer aided design, geographic databases, administration of component structures, and so on.The proposed paradigm allows the definition of a single network up to a family of similar networks. These are networks with similar properties, like network paradigm and structure. The term similar is not restricted to network properties only, but covers also networks, which accomplish special tasks The characteristics of these networks can be very different, but the database input (the task description) is the same. The data manipulation facilities of the system are exploited to handle the networks and their tasks within the same network.

2.2 Theme of Database:

Database systems have proven very valuable at handling and administrating complex objects. In the following guidelines for embedding neural networks into such systems are presented. It is our goal to treat networks as normal data in the database system. From the logical point of view, a neural network is a complex data value and can be stored as a normal data object.It is generally accepted that rule-based reasoning will play an important role in future database applications. The knowledge base consists of facts and rules, which are both stored and handled by the underlying database system.Neural networks can be seen as representation of intensional knowledge of intelligent database systems. So they are part of a rule based knowledge pool and can be used like conventional rules. The user has a unified view about his knowledge base regardless of the origin of the unique rules.

2.3 Adavantages

The usage of a database system as an environment for neural networks provides both quantitative and qualitative advantages.

2.3.1 Quantitative Advantages: Modern database systems allow an efficient administration of objects. This is provided by a smart internal level of the system, which exploits well-studied and known data structures, access paths, and more. A whole bunch of further concepts i inherent in these systems, like models for transaction handling, recovery, multi-user capability, concurrent access, to name only a few. This gives an unchallenged platform in speed and security for the definition and manipulation of large data sets

2.3.2.Qualitative Advantages: The user has powerful tools and models at hand, like data definition and manipulation languages, report generators or transaction processing. These tools provide a unified framework for handling neural networks and the input or output data streams of these nets. The user is confronted with one general user interface only. This spares him awkward tricks to analyze the data of his database with a separate network simulator system.A second very important aspect is the usage of neural networks as part of rule component of a knowledge-base database system . Neural networks represent inherently knowledge by the processing in the nodes

. Trained neural networks are similar to rules in the conventional symbolic sense. A very promising approach is therefore the embedment of neural networks directly into the generalized knowledge framework of a knowledge-based database system .

Fig 2.1 : Sample Database

2.4 The Neural Network-Database Integration Framework:

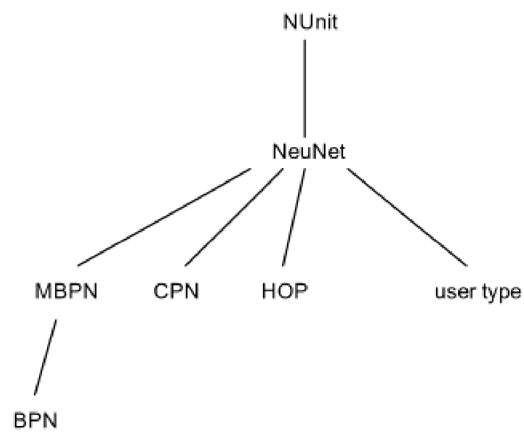
One valuable property of neural nets is their beneath unrestricted flexibility in providing solution strategies to all types of problems. This leads from different versions of neural nets of one net paradigm for different problems to nets of different paradigms for a common problem. The data model of the underlying object-oriented system has to be based on three elements: types, object, and functions.Objects represent unique entities and are classified by types and described by functions (correspond to attributes in the relational model). Types are organized in type hierarchies with function inheritance. Functions access and manipulate objects and can be applied to other functions. This indirection allows a high degree of flexibility and expressional power.

2.4.1 The Neural Networks Types and Ojects:

One describing property of the object-oriented design is the hierarchy of types. A type comprises a set of objects, which share common functions. Generalization and specialization define a hierarchical type structure, which organizes the unique types. All of its subtypes along the type hierarchy also inherit functions defined on a supertypeA paradigm for neural networks in database systems has to provide flexibility in two directions, embedment level and expressional

2.4.1.1 Power Embedment level*:* The system has to support the user on all levels of embedment . The term embedment was introduced to define the user control of the static and dynamic properties of neural network objects. Control means the possibility to change or adapt the properties in question. Existing neural network systems can be classified according to the embedment level

2.4.1.2 Expressional power: The proposed framework has to cover all different types of networks. This reaches from simple networks consisting of a few processing elements to large net systems, which comprise several networks (a network of networks) possibly with different net paradigms. In the last few years these systems yield specific importance, eg, in the area of data analysis, image processing, etc., where different networks are closely connected and responsible for different tasks, like noise reduction, pattern recognition.The object-oriented system has to provide a generic type for a neural network system. This basic generic neural network type comprises all possible network paradigms and embedment level. We call this type NUnit (for Neural Unit). The purpose for this rather generic type is the possibility to express complex neural systems consisting of different networks. Further, it allows for the definition of the data communication with the environment, in our case the database system. This results in data input and output objects. We use the term object to avoid the anticipation with a specialized, implementation dependent, data format, like functions and streams.The NUnit type is a subtype of the general object type. Subtype of this NUnit type is a generic neural network type, NeuNet. It represents a unique network, which defines the net paradigm, connected processing elements. Specializations of this type are predefined system types or customized, user defined types.



BPN	...	backpropagation net
MBPN	....	multilayer backpropagation net
CPN	...	counter propagation net
HOP	...	Hopfield net

Fig 2.2: Type of hierarchy of NUnit

.The general NUnit type provides just the functions for name. A finalized subtype of NUnit, which defines the net paradigm, defines all other describing and necessary characteristics. At a very early moment the user can create an instantiation of a neural network system, but has not to decide about the number of networks, net paradigm, attributes, and functions. This is very useful for database systems where families of networks have to be administrated for similar problems. These neural network objects can be referenced by names as all other objects in the database. All data administration facilities of the system are applicable on NUnit objects as well, as insertion, deletion,and update.Further the user has the possibility to define special properties for an instantiation of these data types, like processing elements, connections, local memory, transfer functions, and so on .In the following discussion we use the general terminology introduced by [22]. A neural network system consists of 2 element sets, nodes N and arcs A. The nodes describe the characteristic properties, like node names, ports, state-space and functions. A node n can be seen as a tuple

n= NODENAME x set(PORTS) x STATE-SPACE x FUNCTIONS

The function component defines all net functions algebraically, like activation, input or output function and all net dynamics, like evaluation and training method.The arcs represent the connections between processing elements. These connections are directed and can be described by an outport and an inport. An arc is a tuple The term image processing generally refers to processing of a two dimensional picture by a digital computer, in the broader context, it implies digital processing of a two dimensional data. A digital image is an array of real numbers represented by a finite number of bits. The principle disadvantage of digital image processing is its versatility, repeatability and the preservation of original data precision.

a = OUTPORT x INPORT Formally, a net is therefore described by

Net = (N, A)

where N is the set of nodes and A is the set of arcs

2.5 Data Acquistion:

Signature images can be acquired with a digital scanner or digital camera. Digital scanner is preferable over a camera for acquiring a document image because of the following reasons:

(i) Lighting: A scanner has its own lighting system for illuminating the object (document). This always maintains a constant light in the image. But in case of a camera, ambient light condition varies and hence it affects the image and results inconsistent images.

(ii) Focus: A scanner always focuses sharply on its object because the flat bed is at fixed position. But a camera may not focus on the document as good as a scanner and there is always a chance of tilt. But a scanner positions the document perfectly flat.

(iii) Quality: In case of a document, the image quality of a scanner is always better than that of a camera

2.5.1 Scanner Specification:

A scanner is specified by many factors such as - Scanner Type, Scanner Resolution, Maximum Resolution, Maximum Scan Area, Scanning Speed, Light Source, Colour Bit Depth etc. Our concern was only the Scanner Resolution. Because other parameters do not affect the essential quality of the signature images required for our purpose. Scanner Resolution Scanner Resolution is a measurement of the resolving power of a scanner. Resolution of a scanner is the measurement of number of pixels that it can sample in the scanned image. It is expressed in dots per inch (dpi). 200 dpi means 200×200 or 40,000 dots per square inch. If resolution of a scanner is more, it implies that more numbers of dots or pixels are captured per inch of the image by the scanner. Image scanned with a higher resolution can be enlarged more. Thus scanner resolution indicates the enlargement capacity of the scanner. Resolution of scanned images is also an important factor that influences the process of signature verification. High resolution results more detailed images. But, they need more storage space and may contain noise. Thus computational cost is also higher. On the other hand, computational cost is lower with low resolutions.

But required information may be lost in such images as their quality is affected by lower resolution. Therefore, resolution of the signature images should be suitably decided.

If a signature is scanned with a higher resolution, the image is sampled at a higher sampling rate. Thus, size of the scanned image becomes larger. It occupies more storage space. On the other hand, an image scanned with low resolution takes lesser space but suffers from information loss due to low sampling rate. Vargas et al. analyzed the effect of image resolution on the accuracy of offline signature verification system. They investigated the performance of an offline signature verification system using Hidden Markov Model with signature image resolutions ranging from 45dpi to 600dpi. They found that 150dpi was the appropriate resolution for acquisition of signature image in their experiment [3]. For our experimentation, we have developed three signature databases. Every database consisted of both genuine and forged (skilled) signatures.

2.5.2 Signature Collection Procedure:

There must be a proper way of collecting the signature samples. Quality of the signature database largely relies on the signature collection protocol. To develop a high quality signature database for research purpose, their collection must be carefully controlled. It is seen that the genuine signatures possess high stability and less variation. But the forged signatures are highly inconsistent. Forged signatures are collected from lay forgers. It is not possible to convince (or even find) a professional forger to develop a database of forged signatures. It is difficult for a non-professional forger to replicate a signature as close as the genuine signature. Such forged signatures are not only deviated from their genuine counterpart, they contain large variations among themselves. These affect the performance of the signature verification system [4].

To overcome the inconsistency in the forged signatures, following approaches are suggested by the researches [5]-[9]:

(i) The forger should be motivated by an award for his good consistent work and they must be encouraged to practice sufficiently.

(ii) Forgers with inconsistency should be denied while developing the database.

(iii) Stability of the signatures should be checked using statistical test. Signature samples passing a minimum predefined threshold limit should only be included in the database.

(iv) Some simulated signatures may be generated from the genuine signatures. These simulated signatures should be used as the forged signatures.

(v) Some researchers have introduced the concept of ‘Disguised Signature’ [10]. Disguised signatures are produced by the genuine authors against their willingness and these are produced with an intension to get rejected in the verification procedure. Thus disguised signatures are like forged signatures which are produced by the genuine authors.

Database	Resolution (in dpi)	No. of sets	Genuine Samples	Forged Samples	Reference
GPDS-39	75	40	24	30	[11]
GPDS-100	600	100	24	24	[12]
GPDS-160	300	160	24	30	[13]
GPDS-960	300	960	24	30	[14]
MCYT-75	600	75	15	15	[15]
CEDAR	600	55	24	24	[16]
4NSigComp2010	600	6	2	2	[17]

TABLE 2.1 SOME AVAILABLE OFFLINE SIGNATURE DATABASES

Table 2.1 shows some available offline signature databases. But all of them contain limited number of signature samples, which according the researchers are not sufficient to develop an efficient signature verification system. A quality database of offline signatures is still a need in the field of offline signature verification

2.5.3 Signature Collection Protocols used in developing our signature databases:

While collecting the genuine signature samples, following protocols were followed:

(i) All the signatures were collected on white A4 size paper.

(ii) Different pens were used by the signer.

(iii) Signatures were collected in seven (7) different sessions in a span of nine (9) days. Some signatures were collected in the morning, some in the noon, some in the evening and some were collected in the late night.

(iv) Signature images were scanned by a digital scanner with 200dpi resolution. 50 | P a g e For producing the forged signatures, the forgers were provided the genuine signatures. To reproduce the signatures, they were allowed to practice the signature as long as they wished. Like the genuine signature samples, the forged signature samples were also collected in an A4 size white paper and scanned with 200dpi. All scanned signature images were stored in JPEG format.

CHAPTER-3

METHODOLOGY

CHAPTER – 3

METHODOLOGY

3.1 Preprocessing:

After capturing the signature samples, the next step is to enhance the images and make them ready for the subsequent processing. That is the scanned images need to be preprocessed before giving them to the next process. Preprocessing is done using signal processing algorithms. Preprocessing greatly helps to improve the performance of feature extraction and classification. It reduces computational cost in classification [18], [19].

Depending on the type of signature pattern, signature image quality and classification techniques to be used, preprocessing operations are determined. It must be kept in mind that during preprocessing, information from the images should not be discarded. Loss of information in preprocessing will affect the overall accuracy of the signature verification system. Various preprocessing techniques are used in offline signature verification as found in these literatures [19]-[29].

Once the signature image is scanned, the next step is to pre-process the image to improve the quality of the image Various methods were used to achieve this, including noise reduction, separating of signature from background, binarization through the identification of an optimal threshold, size normalization, data area , cropping, contrast and line improvement, edge detection by means of Sabor filter, skeletonization (also known as thinning) and segmentation. Well known techniques such as convolution masks, histogram analysis and equalization, gradient evaluation and morphological operators are used.

The pre processing step is applied both in training and testing phases. Signatures are scanned in gray. The purpose in this phase is to make signature standard and ready for feature extraction. The pre-processing stage improves quality of the image and makes it suitable for feature extraction

3.2 Theortical Approach:

The scanned signature image may contain spurious noise and has to be removed to avoid errors in the further processing steps. The gray image Io of size M*N is inverted to obtain an image Ii in which the signature part consisting of higher gray levels forms the foreground.

$$\mathbf{Ii(i,j) = Io,max - Io(i,j)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{(1)}$$

Where Io,max is the maximum gray-level. The background, which should be ideally dark, may consist of pixels or group of pixels with gray values between that of background and foreground. These are removed by performing a row averaging process to generate the row averaged image Ira, which is given by,

$$\mathbf{Ir(i,j) = Ii(i,j) - I{=}1 \sum M\, [Ii(l,j)]/M}$$

$$\mathbf{Ira(i,j) = Ir(i,j) \text{ if } Ir(i,j) > 0}$$

$$\mathbf{= 0 \text{ otherwise } \ldots\ldots(2)}$$

Further noise removal and smoothening is achieved using an n*n averaging filter to generate the cleaned image

$$\mathbf{Ia(i,j) = 1/9\ (\ l{=}i{-}1 \sum i{+}1\ k{=}j{-}1 \sum j{+}1\ Ira(l,k)\ ) \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(3)}$$

The gray image is converted into binary image by using automatic global thresholding. Following algorithm [5] was used to automatically calculate the global threshold:

An initial value, midway between the maximum and minimum gray level value, was selected for the threshold T.

1. Image was segmented using T.
2. Average gray level values μ1 and μ2 for the two groups of pixels was computed.
3. Based on step 3, new threshold value was computed. T = 0.5 * (μ1 + μ2)…..(4)
4. Steps 2 through 4 were repeated until the difference in T in successive iterations was smaller than 0.5.

3.3 Preprocessing involoves the following stages:

3.3.1 Filtering: A scanned signature image may contain noise. Noise in the image deteriorates the feature extraction and its successive processes. Hence, filtering of noise is an unavoidable preprocessing step in pattern recognition. It has been observed that the scanned images are usually affected by salt-peeper noise. A median filter effectively removes such type of noise preserving the edges of the images [18]. We applied a median filter of 3×3 window on our signature images.

The median filter is a non-linear spatial filter that uses a sub-image area or window. This window is usually of square shape and is of fixed size. This window slides over complete image pixel by pixel and replaces the center value in the window with the median of all the pixel values in the window.

The pixel value of the window in Fig. 2.1 (a) in ascending order is 3, 3, 4, 4, 5, 6, 6, 7, 87. So, the median is (the middle value of the string) 5. When the center value in the window (87) which is possibly a noise, is replaced with the median value (5), the following new window in Fig. 2.1 (b) is found, where the noise is removed.

3	5	3
6	87	4
7	6	4

Fig 3.1 A 3*3 window

3	5	3
6	5	4
7	6	4

Fig 3.2: window after noise removed

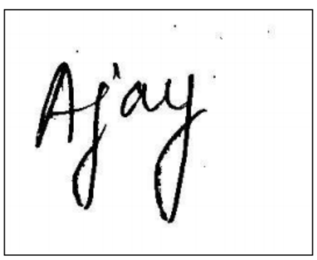


Fig. 3.3. Signature image with noise

Fig. 3.4. Signature image after filtering

[Note: The boundary boxes in the images are for indicating the image boundaries; these are not there in the original signature images]

3.3.2 Binarization:

A colour image comprises of three colour plans Red (R), Green (G) and Blue (B). In a colour image, every pixel value is defined by the combination of the values of these three plans. In a gray level image, there is no colour information. The image is defined by the pixel values of a single plan (the intensity plan). However, in a gray level image, the pixel values will have a range which is specified by the number of bits of the image. Eg. for an 8 bit image the pixel values will range from 0 to 255. When the pixel values of a gray level image is assigned with only two values, a binary image is resulted. Thus, image size is greatly reduced in a binary image as compared to its original colour image or the gray level image. There are two steps to convert a colour (RGB) image into a binary image, (1) Conversion of Colour image into Gray level image and (2) Conversion of Gray level image into binary Iamge

3.3.2.1 Conversion from Colour image (RGB) into Gray level image

There are several algorithms for converting a colour image into a gray level image. The following four algorithms are found to be more common in used [22]:

(i) Average method

In this method, the average value of R, G, B plan is considered as the gray level

$$I = \frac{R+G+B}{3}$$

(ii) **Lightness method**

Here, the most prominent and least prominent colours are averaged.

$$I = \frac{\left(\max\left(R,G,B\right)+\min\left(R,G,B\right)\right)}{2}$$

I = Calculated gray level

R= Value of Red colour plan

G= Value of Green colour plan

B= Value of Blue colour plan

(iii) Luminosity method

Human perception is taken into account in this method. It is weighted average method. Human brain is more sensitive to green colour than red and it is least sensitive to blue colour. Accordingly in this method, different weights are given to these colours.

I R G B = + + 0.21 0.72 0.07

(iv) Standard NTSC (National Television System Committee):

conversion Method Perception of this method is also similar to Luminosity method, but here the weights are different. This method is a standard method accepted by NTSC and is widely used. MATLAB® Image Processing Toolbox uses this method for converting a colour image into a gray level image.

I R G B = + 0.2989 +0.587 0.114 B

3.3.2.2 Conversion of Gray level image into binary image:

A suitable threshold value (pixel value) is considered to convert a Gray level image into a binary image. If a pixel value in the gray level image is greater than the threshold value then the new pixel value assigned is 1 (one) else 0 (zero). Thus, the new image will have only two pixel values ‘1’ (which corresponds to white) and ‘0’ (which corresponds to black).

Fig. 3.5. A graylevel signature image

Fig. 3.6 .A binary signature image

3.3.3 Cropping:

When scanned, signature image contains the signature and some white coloured non-signature regions. Those superfluous non-image portions are removed by cropping the image to the bounding rectangle of the signaturepart. Cropping is an essential preprocessing step for all types of classification techniques.

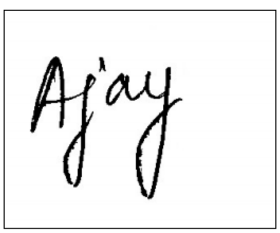


Fig.3.7. An uncropped image

Fig.3.8. A cropped image

[Note: The boudary boxes in the images are for indicating the image boundaries; these are not there in the original signature images]

3.3.4 Thinning:

In thinning, the signature image strokes are made one pixel thick. Thinning is mainly done to reduce the amount of data in the image. This helps to decrease the storage space requirement and also to reduce the computational complexities in successive stages. But during thinning, some information of the signature images such as stroke width may be lost. So, depending on the features to be extracted, thinning may or may not be required.

There are various thinning algorithms found in literatures [30]-[35].

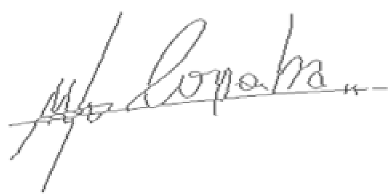


Fig 3.9 Thinning Image

3.3.5 Image Resizing:

Signature lengths are different for different signers. Even the lengths of the signatures of a single person are also not equal. But when a grid based signature verification approach is used, the signatures are projected on the grid of same size. Hence, all the signatures must be of same size. Therefore in that case, resizing of signature becomes important [44]. But, resizing is not a compulsory preprocessing step for all signature verification approaches.

The most basic method of image resizing is a kind of geometric transformation. In this method, there are two basic operations: (i) spatial transformation and (ii) gray level interpolation.

In spatial transformation, some pixels or points ('tie-points') are selected whose positions in the original image and the resized image are precisely known. From their locations in the two images, a spatial transformation equation is formulated. This equation is used as a mapping equation to find out the positions of all the pixels in the new resized image.

Gray level interpolation is used to assign gray levels to the new pixels in the resized image. It uses a nearest neighbour approach. In this method, gray level is assigned according to the pixel which is the nearest to the mapped pixel [18]. Some other algorithms used for resizing are discussed in [45].

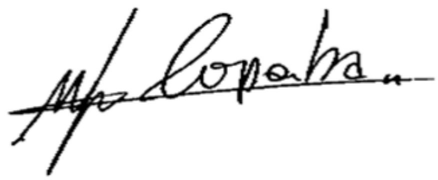


Fig 3.10.Resized image

3.3.6 Bounding box of the Signature:

In the signature image, construct a rectangle encompassing the signature. This reduces the area of the signature to be used for further processing and saves time. shows signature enclosed in a bounding box.

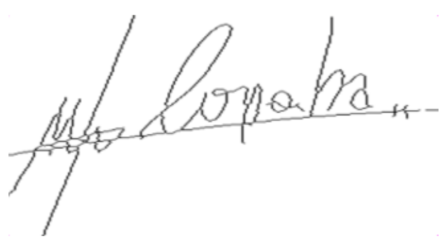


Fig 3.11.Bounding box of an Image

3.4 Feature extraction:

feature extraction is a process of deriving some characteristic parameters or functions from the patterns (signature images). The extracted characteristic parameters or functions are called ‘features’. Function features are functions of time and these can only be derived from online signatures. Characteristic parameters are extracted from offline signatures [49]. The features should affectively represent their parent patterns with reduced amount of data. Feature extraction helps in decreasing the computation complexities in the subsequent stages of signature verification.

The success of any pattern recognition system significantly depends on feature extraction. Extracted features must minimize the dissimilarity between same class patterns and must maximize the dissimilarity between two patterns from different classes. An ideal feature extraction method in offline signature verification system, should extract a minimum number of features that maximize the distance between the signature examples of other persons (interpersonal distance) but should minimize intrapersonal distance for those belonging to the same person [2].

Extracting features from an offline signature is challenging as compared to an online signature. Because in offline signature, information of dynamic features like pressure, acceleration, stroke order in the signature are lost [50].

3.4.1 Types of features in offline signatures:

Features extracted from an offline signature are basically classified into two categories [20], [51], [52]:

(i) Local Features and (ii) Global Features

3.4.1.1 Local Features:

Local features are extracted from a small part or a small region of the signature. The critical, distinct parts carrying distinguishing features are selected for this. Local features are very much noise sensitive. Extraction of local features is computationally expensive.

3.4.1.2 Global Features:

Global features are extracted considering the complete signature image s a whole. Global features are easy to extract and these features are least sensitive to noise. But global features are affected by position alignment and they are highly susceptible to signature variations [1].

Two additional types of signature features are also found in literatures. They are

(i) Geometrical features and (ii) Statistical features in [1], [53].

3.4.1.3 Geometrical Features:

Geometric features are derived from the geometrical parameters of the signature such as the height, width, aspect ratio, signature area etc. These features depict the characteristic geometry and topology of a signature. Both global as well as local properties are preserved in geometric features. This type of features has the ability to withstand distortion and rotation variation [54].

3.4.1.4 Statistical Features:

In many approaches of offline signature verification, researchers have used statistical features of the signature. They are derived from distribution of pixels in the signature image. Some statistical features extracted from offline signatures are mean, centre of gravity of the signature image, global maxima, local maxima, moments etc. Statistical features can tolerate slight variations in signature style and distortion

3.4.2. Features in offline signatures:

In literatures various features are found to be extracted from offline signatures [27], [55]- [60]. It has been observed that global features produced better recognition results in offline signature verification (please refer to page no. 21 of Chapter 1). We had tried to extract more number of global features with already proposed promising global features and some new global features proposed by us. We extracted the following features from the signature samples present in our datasets:

We extracted the following features from the signature samples present in our datasets:

3.4.2.1 Normalized Signature area (with respect to bounding box):

It is the total number of signature pixels or foreground pixels in the signature image. Signature area gives information about the signature density. If the signature image is skeletonized, signature area represents a measure of the density of the signature traces.

If in a signature image, total number of signature pixels (or black pixels if foreground image is black) = B,

and total number of pixels in the whole image = P, then

$$\text{Normalized Signature Area} = B/P$$

Normalized signature area is also called Signature Occupancy Ratio

$$\text{Signature Occupancy Ratio} = \frac{\text{number of pixels which belong to the signature}}{\text{total number of pixels in the signature image}}$$

Steps involved:

1. Read the preprocessed signature image.

2.Scan the image row wise and total number of black pixel is to be counted.

3.4.2.2 Aspect Ratio (Signature width to height ratio):

This is ratio of signature width to signature height of a cropped signature. It is seen that aspect ratio of the signatures of a person fairly remains constant.

If signature height is H and signature width is W, then Aspect Ratio is given by

Aspect Ratio= W/H

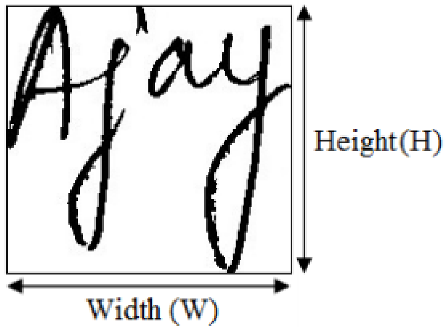


Fig.3.12. A signature image showing height and width

Steps involved:

1.Read the pre-processed signature image.
2. For width calculation, first by going column wise calculate the column which has the left most pixel as black and similarly the column which has the right most black pixel is calculated. Widthis the difference of those two column values.
3. For height calculation by scanning the image row wise check for the rows that encounters first black pixel and similarly the row which has the last black pixel.
4. Height to width ration is to be calculated.

3.4.2.3 Horizontal and vertical center of the signature:

These two measurements indicate about the Horizontal and Vertical location of the signature image.

The horizontal center (Cx) is given by

$$C_x = \frac{\sum_{y=1}^{x_{\max}} x \sum_{x=1}^{y_{\max}} b[x,y]}{\sum_{x=1}^{x_{\max}} \sum_{y=1}^{y_{\max}} b[x,y]}$$

The vertical center (Cy) is given by

$$C_y = \frac{\sum_{y=1}^{y_{\max}} y \sum_{x=1}^{x_{\max}} b[x,y]}{\sum_{x=1}^{x_{\max}} \sum_{y=1}^{y_{\max}} b[x,y]}$$

b x y [,] indicates signature pixel (black pixel)

Steps involved:

1. Read the preprocessed signature image.

2. Scan column wise. For each column, those row index values, which are having black pixels, are added in the row_index_sum. Also a counter is incremented each time a black pixel in any row is found for that particular column.

3. The same step is performed for all the columns.

4. Cx=row_index_sum / total black pixels encountered.

5. Scan row wise. For each row those column index values, having black pixels are added in column_index_sum. Also the counter is incremented each time a black pixel is encountered.

6. The same step is performed for all the rows.

7. Cy= column_index_sum/total black pixels encountered.

8. Centre is calculated by formula-= (Cx+1) * total column in signature + Cy.

9.This centre as cell value is stored as centre feature

3.4.2.4 Centre of Gravity or Centroid:

In a binary signature image with black signature pixels, Centre of Gravity (CG) or Centroid is the average coordinate point of all black pixels. The CG of a signature image is calculated by the following equations:

$$X = \frac{1}{N}(\sum_{i=1}^{n} x_i) \qquad Y = \frac{1}{N}(\sum_{i=1}^{n} y_i)$$

Where, i x is the column number of ON pixels and i y is the row number of ON pixels. The pixels that constitute the stroke of the signature image are called the ON pixels. In a binary image, ON pixels are the black pixels.

[Note: The boudary boxes in the images are for indicating the image boundaries; these are not there in the original signature images])

3.4.2.5 Tri surface feature:

Two different signatures may have same area .so to increase the accuracy of the features three surface feature has been used. A signature is split into three equal pieces, with the area of each section determined separately

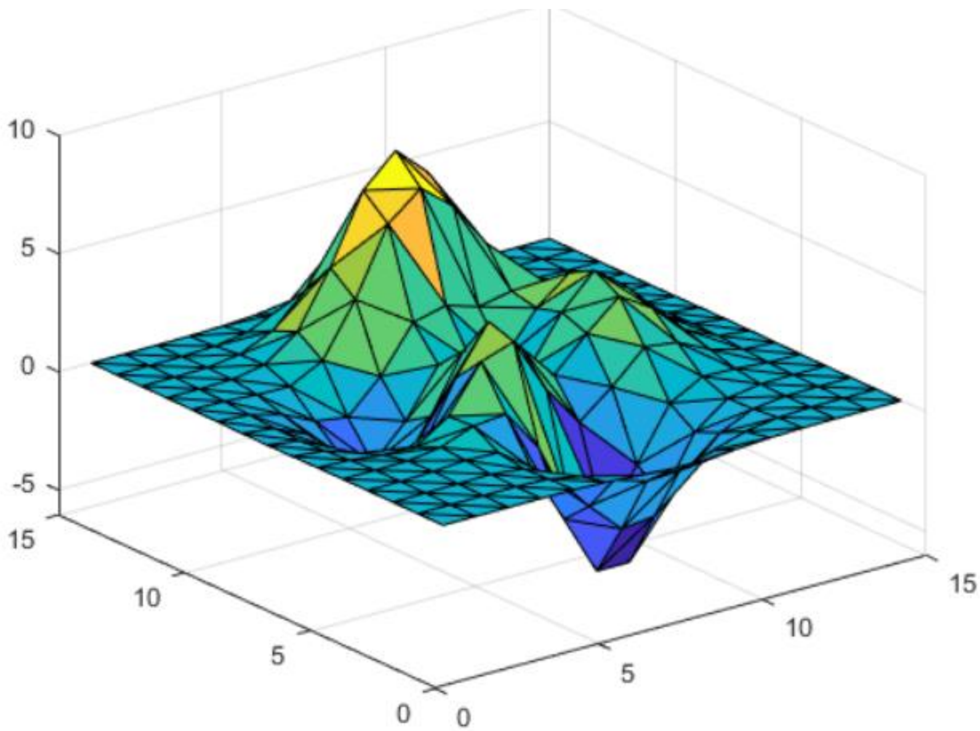


Fig 3.13 Trisurface plot of an image

3.4.2.6 The six fold surface feature :

Find a bounding box for each portion of a signature by dividing it into three equal sections. Then figure out where each part's centre of mass is. Calculate the area of signature above and below the centre of mass inside a bounding box by drawing a horizontal line through the centre of mass of each part.

3.4.2.7 Maximum black pixels:

The image is scanned horizontally and vertically and therow and column with maximum black pixels is recorded.

3.4.2.8 Transition feature :

traverse a signature picture from left to right, calculating a ratio between the position of the transition and the width of the picture traversed and recording it as a feature each time there is a transition from 1 to 0 or 0 to 1. Repeat the process from left to right, top to bottom, and bottom to top. Calculate the total number of transitions from 0 to 1 and 1 to 0 as well

3.4.3 Choice of Features:

While considering the choice of features in an offline signature verification system, following two aspects need to be considered:

1. Types of features
2. Total number of features

3.4.3.1 Types of features:

Two factors decide the choice of types of features (global or local) for a signature verification system. They are (i) the style of the signature and (ii) the type of forgeries to be detected. It has been suggested by some of the researchers that a combination of global and local features may improve the performance of an offline signature verification system (cited by [61])

Computational cost in extracting global features is low. Global features are less susceptible to noise and variations in the signatures. Recognition rate achieved in case of skilled forgeries using global features is not very high. But these features are found suitable for detecting random forgeries. Global features perform better when combined with other types of features [62].

Local features are extracted locally; so they are affected by zoning process. Some researchers have reported that local features perform better in detecting skilled forgeries [2].

Local features are related to a small region of the signature and they extract more exhaustive information from the image. Local features are very much noise sensitive. Computational cost of local features is high. But computation of one region is not influenced by other regions of the signature. Accuracy of local features is higher than global features [63].

Amount of information delivered by global features is limited [64]. Global features are immune to small distortions at the local regions of the signature. But a change in the overall position alignment of the signature has an effect on global features. Global features are vulnerable to style variations [2].

Local features have the ability to discriminate writers. They extract detailed representation of writing shapes. But finding and extracting consistent local features is very difficult. In one-dimensional sequence, calculation of local shape features is easier than that of two dimensional images. Therefore, local features are more popular in online signature verification[64].

In manual signature verification, ordinarily the verifier compares the global features in the original and the test signatures to decide its genuineness. In global features, intra personal variation is very low.

There are some global features which can be applied locally to a small region of the signature. Similarly, some local features also can be applied globally. As an example, contour based features can be extracted at both local and global level [20]. According to Hou Weiping et al. [63], a proper combination of both the types of features may increase the effectiveness of the signature verification system. If a global feature is applied locally (or vice versa), the system can take advantage of both types of features and their negative aspects may be avoided. This is how a signature verification system can be made more immune to intrapersonal variances (variations in the signatures of a single person) and hence ability of the classifier can be improved.

3.4.3.2 Total number of features:

Increasing the number of feature doesn’t necessarily produce a good result in classification. In fact in most of the situations, this will have a negative effect on classification. It is because all the extracted features may not carry the essential uniqueness of its parent pattern.

Some of the features carry some ambiguous information of the pattern, which confuse the classifier system and as a consequence classification accuracy is hampered. Thus, from all the extracted features, some useful features need to be selected for better classification efficiency.

In pattern recognition, feature selection is a process of selecting a subset of the most important and relevant features from the complete set of extracted features so that the total number of features is reduced but their class discriminatory information is still intact [64]- [66]. All the extracted features carry information of the pattern from which they were extracted. But all of them don’t carry information about the pattern equally. In a feature set, some of the features are irrelevant or redundant. If the classifier is fed with the irrelevant features, there three problems may be induced by the irrelevant features:

(i) Due to more number of features, computational cost increases.

(ii) Presence of the irrelevant features may cause misclassification and thus the classification efficiency is decreased.

(iii) (iii) The irrelevant features may cause overfitting. Therefore, pattern classification is not efficiently done by the complete set of features; instead it is affectively done by some of the selected features (a subset of the features) excluding the irrelevant features. In any pattern recognition system, feature selection is one important step that improves the classification efficiency.

Overfitting:Overfitting is a curve fitting problem associated with classifier design. When a limited number of data points are available and a function is fitted too tightly to them to achieve perfect classification in the training samples, overfitting occurs. Practically in every set of data, some amount of random errors is always present. No data set can be guaranteed as error free or accurate. Therefore, when any data set is tried to model too closely, the errors present there exaggerate the model and the model becomes excessively complex, its predictive power is also reduced.

Underfitting: When a learning model is designed excessively simple by decreasing the variance, it results underfitting. In undefitting condition, learning model can't model the fundamental trend of the data accurately and shows poor prediction.

Goodfitting: This is the optimal fitting curve. Both overfitting and underfitting lead to poor predictions on new data sets.

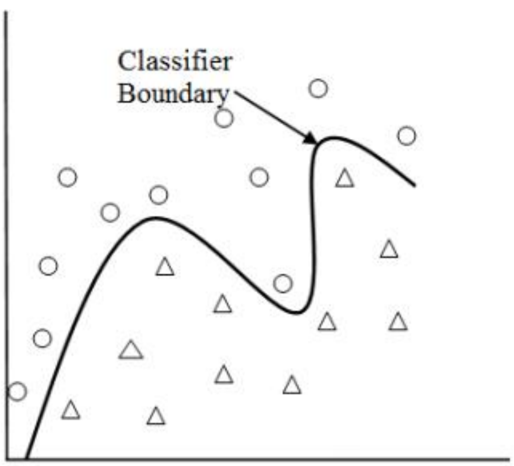


Fig 3.14 Overfitting

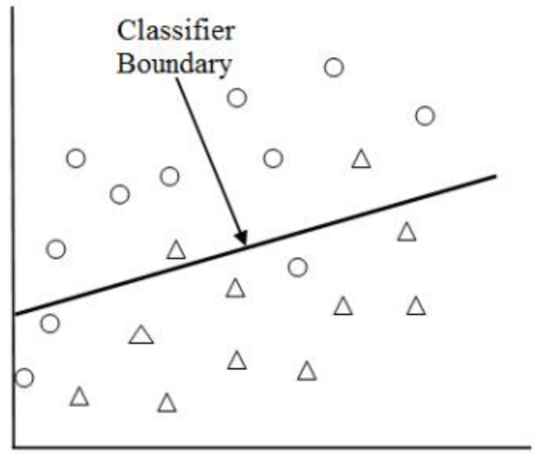


Fig 3.15 Underfitting

○← Data from class I

△← Data from class II

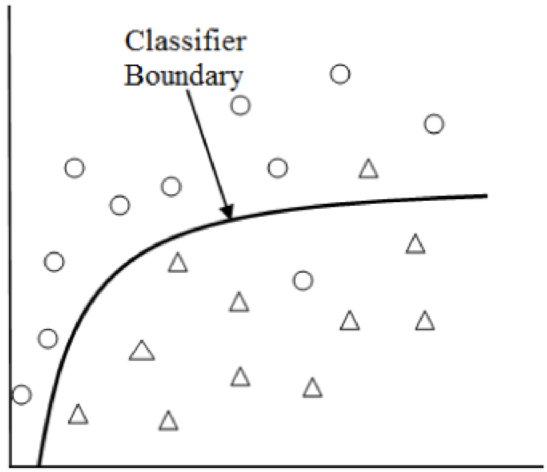


Fig 3.16 Good fitting

3.4.4 Feature Selection Methods:

The basic purpose of feature selection is to find out the smallest possible feature set that sufficiently represents the pattern (signature). Sometimes the features which do not seem to be relevant alone may be highly relevant when taken with other features. But at the same time, relevant features are redundant; they increase computational complexity. This makes the selection of the best feature subset a difficult task

An exhaustive search through the space of feature subsets is required to find out the 'best feature subset' that contains the least number of features and contribute most towards classification accuracy. There are four aspectsthat are decisive in the search process

(i) Starting point of search in the feature space

(ii) Evaluation process of subset of features

(iii) The search procedure

(iv) Stop point of search

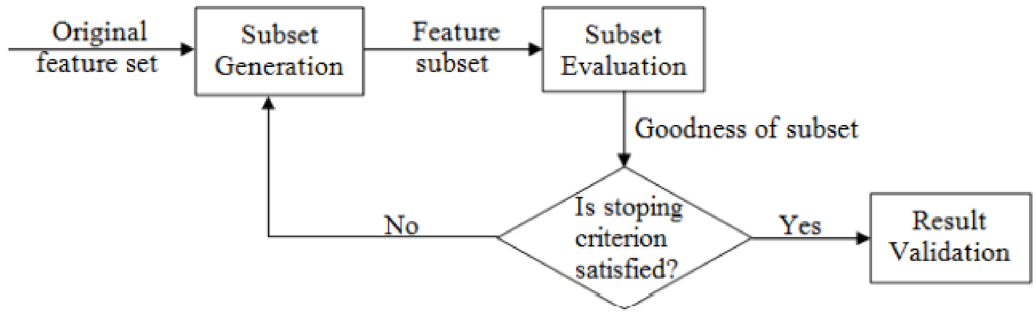


Fig 3.17 Flow chart of Feature selection process

3.4.4.1 Starting point of search in the feature space:

Depending upon the starting point of search, there can be two approaches of feature selection - (a) Forward selection and (b) Backward selection

(a) **Forward selection:**

Initially, there are no features selected. Then the features that give the minimum error are added one by one. This process is repeated until any further addition of features does not contribute to significant decrease in error.

(b) **Backward selection**:

This approach starts with all features considered initially. Then the features with the highest error are eliminated one by one. This process is repeated until any further elimination of features does not contribute to significant increase in error

3.4.4.2 Evaluation process of subset of features :

There are three main methods for the evaluation of subset of features [68]-[74]:

(i) Filter methods

(ii) Wrapper methods and

(iii) Embedded methods.

Filter Methods : This method was first proposed by John, G. H., Kohavi, R, & Pfleger [69]. Filter method of feature selection uses different statistical tests (e.g. T-test, F-test, i-test, Euclidean distance, (2 χ) Chi-squared test, ANOVA, Information gain, Correlation coefficient scores etc) to find out the features that have the highest predictive power. For a chosen statistical measure, this method calculates a score for each feature and based on the scores features are given a rank. Filter method doesn't take the classifier (Learning algorithm or Induction algorithm) to be used into account.

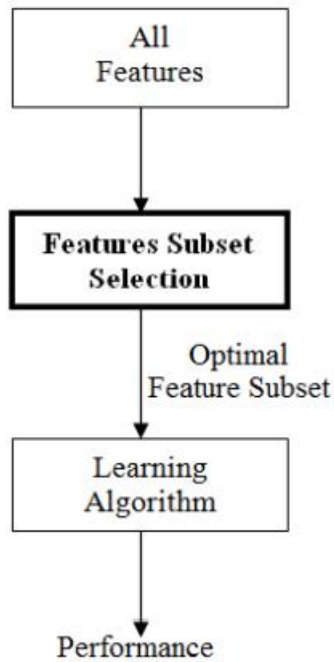


Figure 3.18. Operations in filter method

Filter Algorithm [69], [70]

input: $F(F_0 F_1 F_2 ...., F_{n-1})$ // a training data set with N no. of features

S_0 // a subset with which the search is started

T // a stopping criterion

output: S_{best} // the best selected subset

01 **begin**

02 **initialize:** $S_{best} = S_0$;

03 $\chi_{best} = eval(S_0, F, M)$; // evaluate S_0 by an independent measure M

04 **do begin**

05 $S = generate(F)$; // generate a subset for evaluation

$\chi = eval(S, F, M)$; // evaluate the current subset S by M

if (χ is better than χ_{best})

$\chi_{best} = \chi$;

$S_{best} = S$;

end until (T is arrive at);

return S_{best};

end;

Advantages of filter methods:Filter method is computationally easy and fast. There is no chance of overfitting when this method is used. Selection of features is done only once without using any classifier. So, the selected features can be used with different classifiers.

Drawbacks of filter methods: This method doesn't check the relationships between two different features, thus feature correlation is ignored. Therefore there is a chance that redundant features may get selected in filter method; this results poor classification performance

CHAPTER 4

TRAINING

AND

TESTING OF ANN

CHAPTER-4

TRAINING AND TESTING OF ANN

4.1Artificial Neural Network:

Artificial neural networks (ANN the biological neural networks that constitute animal brains. 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 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 about cats other about cats, for example, that they have fur, tails, whiskers and cat generate identifying characteristics from the learning material that they process

An ANN is based on a collection of connected units or nodes called the neurons in a biological brain. Each connection, like the signal from one artificial neuron to another. An ar then signal is a and the output of each artificial neuron is computed by some non 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 decreas Note that while complex features give more information, simpler features such as gradient orientation are more robust to normal variations found in a signature..

Histogram of image pixel Network 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 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.

based on a collection of connected units or nodes called artificial neurons, which loosely model 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 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 that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at Note that while complex features give more information, simpler features such as gradient orientation are are computing systems

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

4.2 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 −

4.3 Types of Artificial Neural Networks :

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

4.3.1 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.

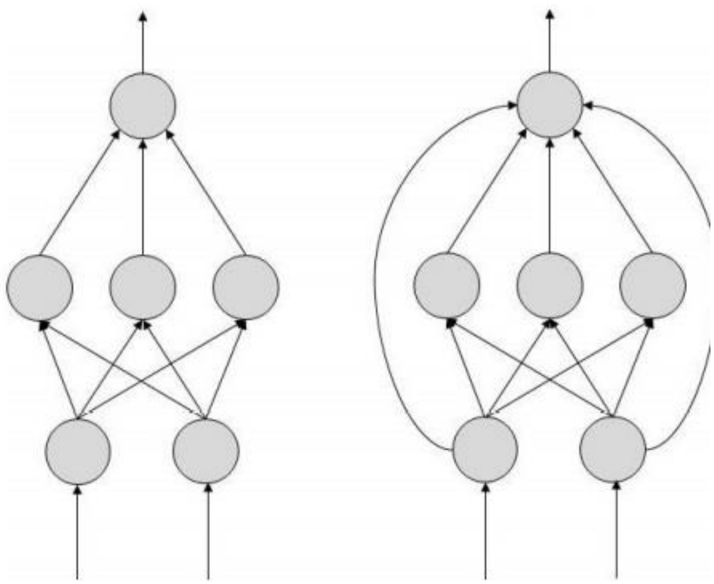


Fig 4.1 Feed Forward ANN

A feedforward neural network is a biologically inspired classification algorithm. It consist of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal: each connection may have a different strength or weight. The weights on these connections encode the knowledge of a network. Often the units in a neural network are also called nodes.Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. During normal operation, that is when it acts as a classifier, there is no feedback between layers. This is why they are called feedforward neural networks. the following figure we see an example of a 2-layered network with, from top to bottom: an output layer with 5 units, a hidden layer with 4 units, respectively.

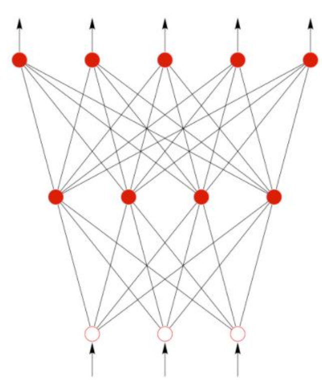
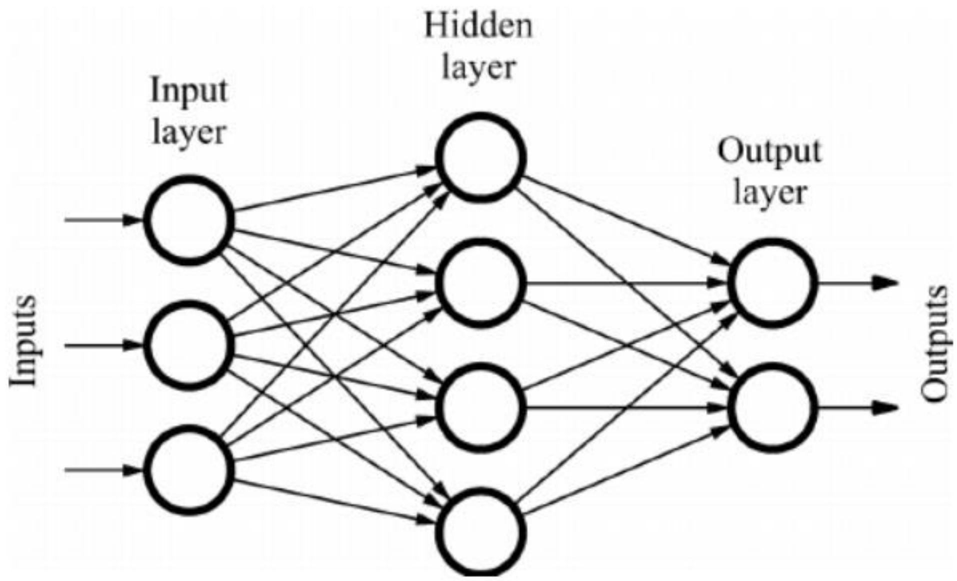


Fig 4.2 ANN With 3 input nodes

The 3 inputs are shown as circles and these do not belong to any layer of the network (although the inputs sometimes are considered as a virtual layer with layer number 0). Any layer that is not an output layer is a hidden layer. This network therefore has 1 hidden layer and 1 output layer. The figure also shows all the connections between the units in different layers. A layer only connects to the previous layer.



Source

Fig 4.3 Flow of information

4.3.1.1 Applications of Feed Forward Neural Networks:

While Feed Forward Neural Networks are fairly straightforward, their simplified architecture can be used as an advantage in particular machine learning applications. For example, one may set up a series of feed forward neural networks with the intention of running them independently from each other, but with a mild intermediary for moderation. Like the human brain, this process relies on many individual neurons in order to handle and process larger tasks. As the individual networks perform their tasks independently, the results can be combined at the end to produce a synthesized, and cohesive output.

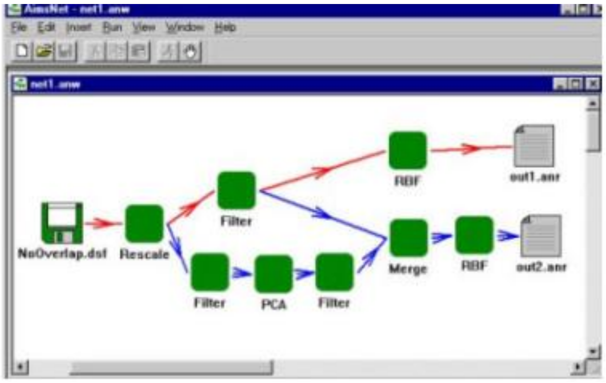


Fig 4.4 Neural Network software

4.3.2 FeedBack ANN :

Here, feedback loops are allowed. They are used in content addressable memories. Signals can travel in both the directions in Feedback neural networks. Feedback neural networks are very powerful and can get very complicated. Feedback neural networks are dynamic. The 'state' in such network keep changing until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback neural network architecture is also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations. Feedback loops are allowed in such networks. They are used in content addressable memories.

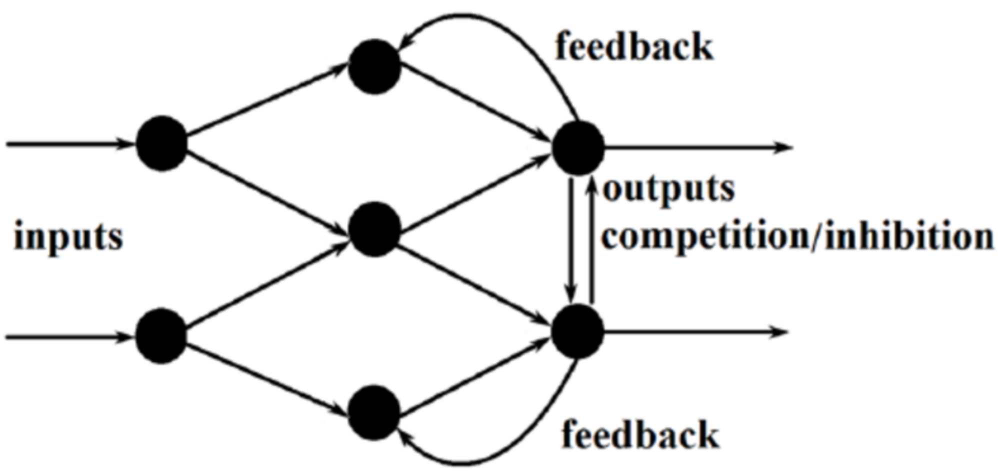


Fig 4.5 Feed back ANN

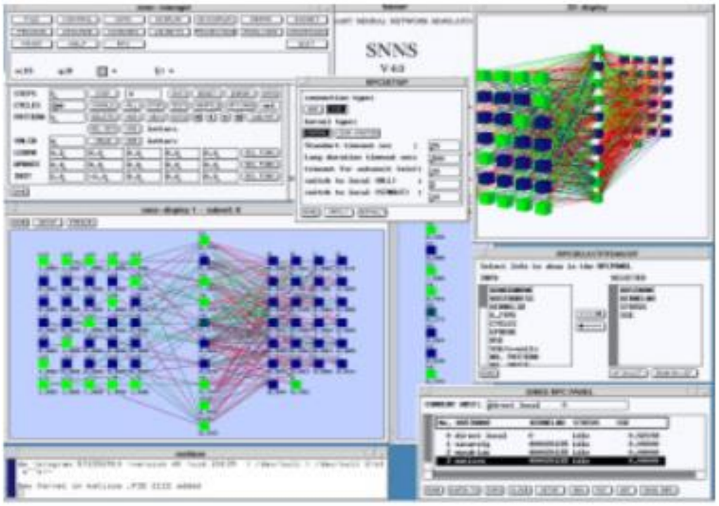


Fig 4.6. Simulation of ANN

4.4 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.

4.5 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.

4.6 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

The backpropagation algorithm is based on generalizing the Widrow-Hoff learning rule. It uses supervised learning, which means that the algorithm is provided with examples of the inputs and outputs that the network should compute, and then the error is calculated. The backpropagation algorithm starts with random weights, and the goal is to adjust them to reduce this error until the ANN learns the training data. Standard backpropagation is a gradient descent algorithm in which the network weights are moved along the negative of the gradient of the performance function.

The combination of weights that minimizes the error function is considered a solution to the learning problem. The backpropagation algorithm requires a differentiable activation function, and the most commonly used are tan-sigmoid, log-sigmoid, and, occasionally, linear. Feed-forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. This structure allows the network to learn nonlinear and linear relationships between input and output vectors. The linear output layer lets the network get values outside the range − 1 to + 1.

.For the learning process, the data must be divided in two sets: the training data set, which is used to calculate the error gradients and to update the weights; and the validation data set, which allows selecting the optimum number of iterations to avoid overlearning. As the number of iterations increases, the training error drops, whereas the validation data set error begins to drop, then reaches a minimum and finally increases. Continuing the learning process after the validation error arrives at a minimum leads to overlearning. Once the learning process is finished, another data set (test set) is used to validate and confirm the prediction accuracy.

Properly trained backpropagation networks tend to give reasonable answers when presented with new inputs.Usually, in ANN approaches, data normalization is necessary before starting the training process to ensure that the influence of the input variable in the course of model building is not biased by the magnitude of its native values, or its range of variation. The normalization technique, usually consists of a linear transformation of the input/output variables to the range (0, 1).

4.6.1 Speeding up of back propagation algorithm:

The backpropagation algorithm gives approximations to the trajectories in the weight and bias space, which are computed by the method of gradient descent. The smaller the learning rate in Eqs. (3.4) and (3.5) we used, the smaller the changes to the weights and biases of the network will be in one iteration, as well as the smoother the trajectories in the weight and bias space will be. However, the update process will be achieved slowly using a slower learning rate. If we use a too large learning rate in order to speed up the rate of learning, the resulting changes in the weights and biases of the network may become unstable.

Therefore, a simple method of speeding up the training process and simultaneously avoiding the problem of instability is developed to modify the backpropagation algorithm. The main change is that a momentum term is introduced to the update rules in Eqs. (3.4) and (3.5). The modified rules of the updating the weights are expressed as

$$\Delta w_{i,j}(k) = \eta \Delta w_{i,j}(k-1) + \alpha \frac{\partial E(w,b)}{\partial b_i^l}$$

$$w_{i,j}^l(k) = w_{i,j}^l(k-1) - \Delta w_{i,j}(k)$$

where η is called the momentum constant and is usually set as a value between 0 and 1. The modified rules of updating the biases are as same as those of updating the weights. The addition of the momentum term not only smoothes the weight and bias updating but also tends to resist erratic weight changes because of the gradient noise or high spatial frequencies in the weight and bias space. This modified backpropagation algorithm is the mostly used algorithm for training MLP in intelligent fault diagnosis.

There are also some modified strategies but they are not commonly used, so we have not included them in this book.One reason of the popularity of the MLP lies on its nonlinear ability, which actually originates from the function f^l shown in Fig. 3.4. This function is usually a differentiable nonlinear activation function. There are two types of the most widely used activation functions in the MLP, that is, sigmoid function and tanh function.

4.7 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 = {X1, X2, …,Xn} of discrete random variables, where each variable Xi may take values from a finite set, denoted by Val(Xi). If there is a directed link from variable Xi to variable, Xj, then variable Xi will be a parent of variable Xj 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.

4.8 Applications of Neural Network:

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.

4.9 ANN Training and Testing:

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. .

4 major stages involved in a training phase:

1) Obtaining signature from a database

2) Image Pre-processing

3) Extraction of features

Instructions for neural network training:

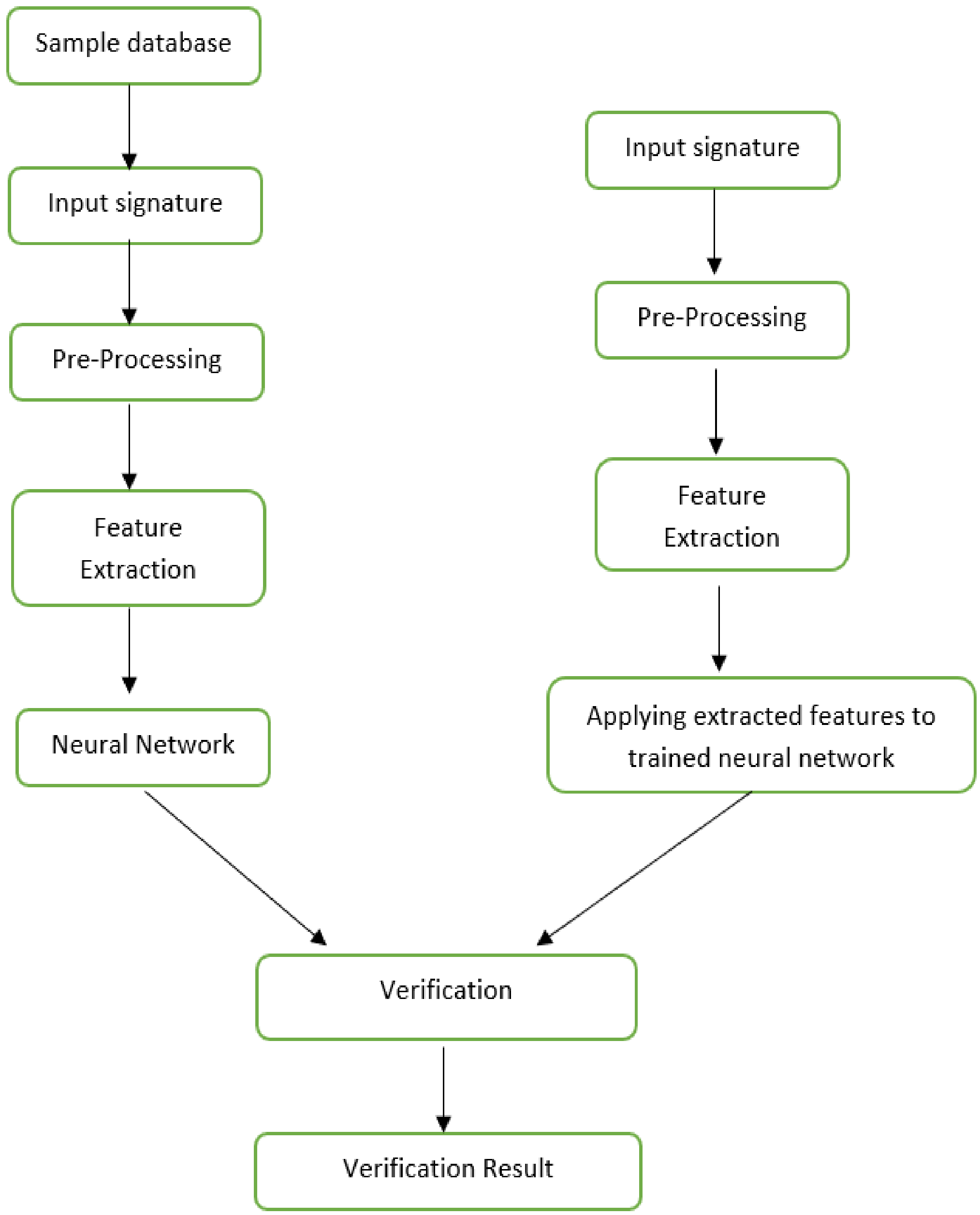


Fig 4.7 Flowchart

4) 4 main stages involved in a testing phase:

1) Improving a signature that can beevaluated against a database
2) Pre-processing of images
3) Extraction of features
4) Validation

4.10 Algorithm:

Input :signature from a database

Output: verified signature classified as a genuine or forged

1. Retrieval of signature image from a database
2. Preprocessing the signatures
3. Feature extraction
4. Creation of feature vector by combining extractcted features
5. Normalizing a feature vector
6. Training a neural network with a normalized feature vector
7. Steps 1 to 5 are repeated for testing signatures
8. Applying normalized feature vector of test signatures to trained neural network
9. Using a result generated by the output neuron of the neural network declaring a signature as a genuine or forged

4.11 Creation of Feature Vector:

1. A feature vector of size 27 is formed by combining allthe features extracted as discussed above.

2. Extracted 27 feature points are normalized to bring them in the range of 0 to 1.

3.These normalized features are applied as input to the neural network

4.12 Validation:

In the verification stage, thesystem compares the extracted features from tested signature with the features extracted from the corresponding signaturein the database in order to verify the authenticity of the signature and makes a final decision for verification as genuine orforged signature. As verifier, in this paper the ANN is used. Among the main advantageS that discriminate ANN is Fast training processand gives,accurate results.

CHAPTER 5

RESULTS AND CONCLUSIONS

CHAPTER-5

RESULTS AND CONCLUSIONS

The proposed system has been tested with 350 signatures from 10 different persons were used in this project. 300 signatures are used for training the network and remaining 50 signatures are used for testing the neural network. The input of the neural network will be set of feature vectors are extracted from the original images. Target to the neural network is Normalised vector which is specified manually in a excel sheet of dimension 10 X 300 where 10 is number of classes and 300 is number of images.The confusion matrix plotted for 300 images from 10 different persons are as follows. Figure 5.1 shows the 5.2 layer neural network. Figure 2 shows the training state of the neural network. Fig. 4 shows the performance plot in neural network.

5.1 Neural Network : Multi-layer networks use a variety of learning techniques, the most popular being back-propagation. Here, the output values are compared with the correct answer to compute the value of some predefined error-function. By various techniques, the error is then fed back through the network.

Using this information, the algorithm adjusts the weights of each connection in order to reduce the value of the error function by some small amount. After repeating this process for a sufficiently large number of training cycles, the network will usually converge to some state where the error of the calculations is small.

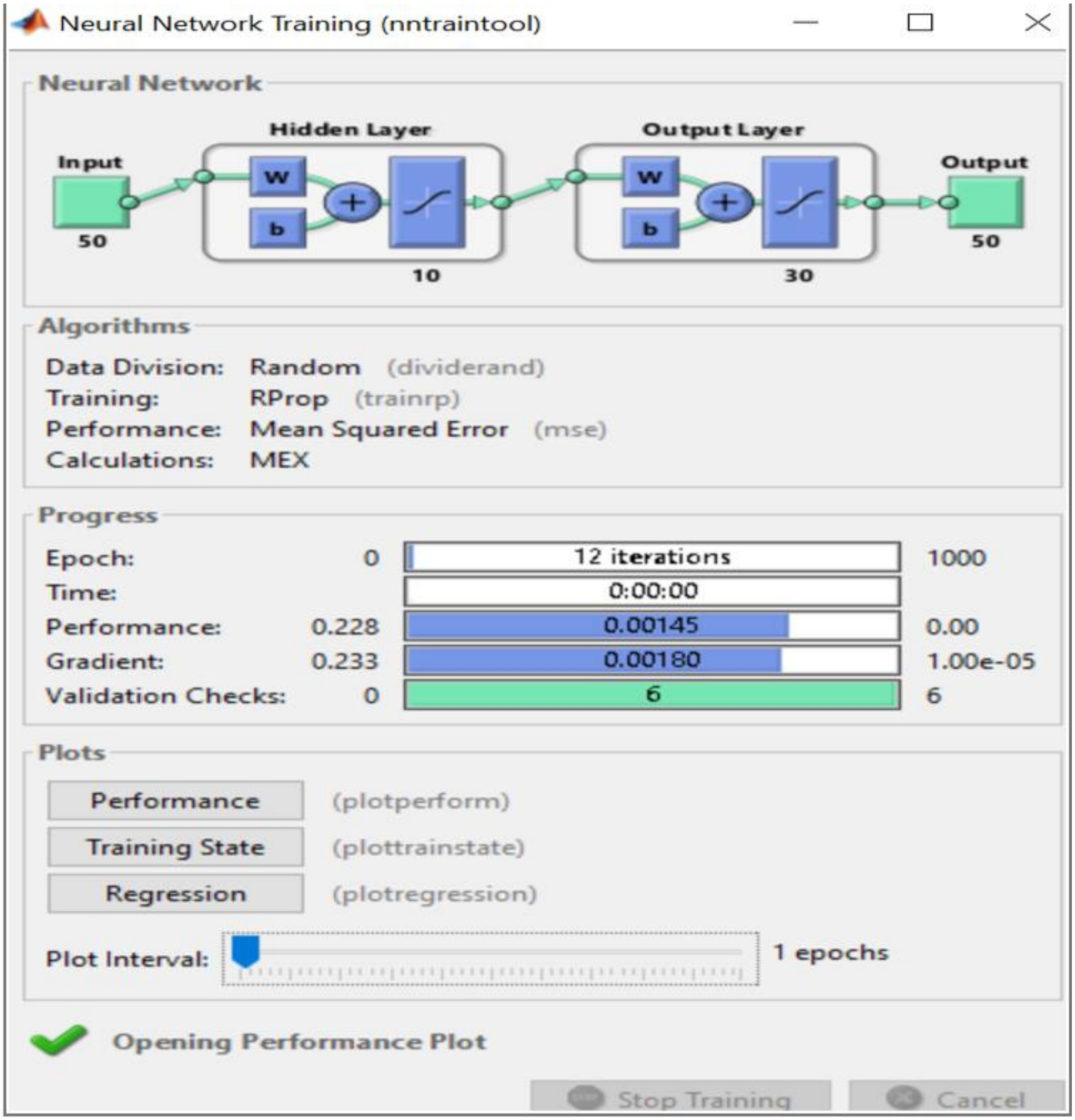


Fig 5.1 Creation of Neural Network

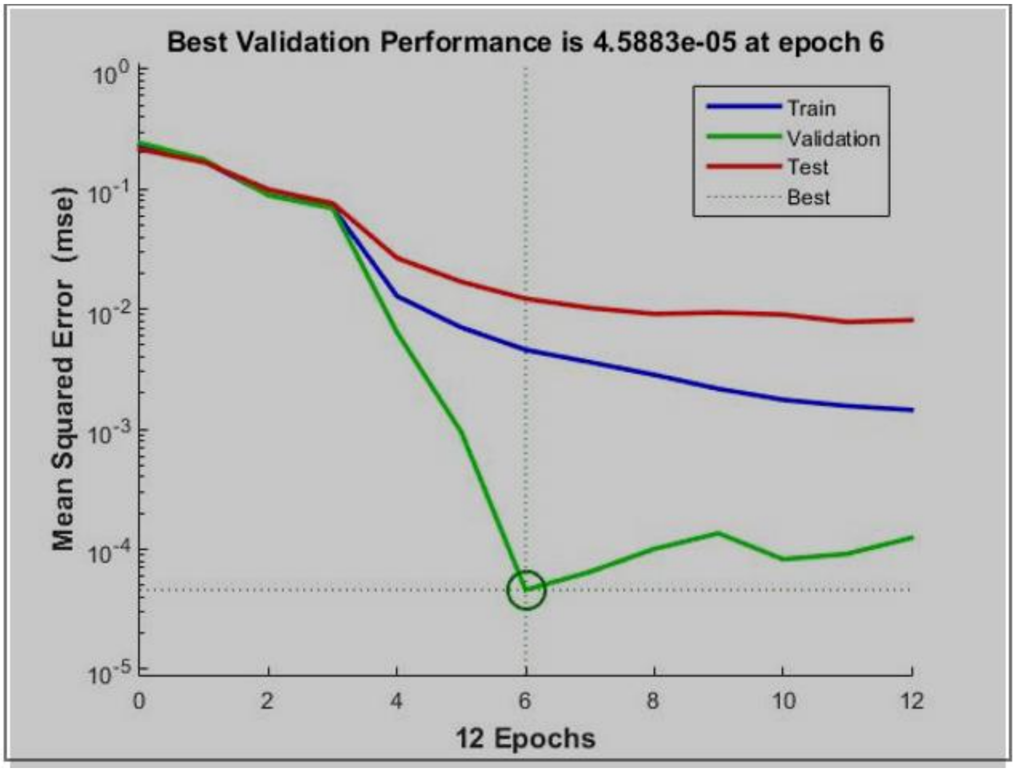


Fig 5.2 trained ANN Performance plot

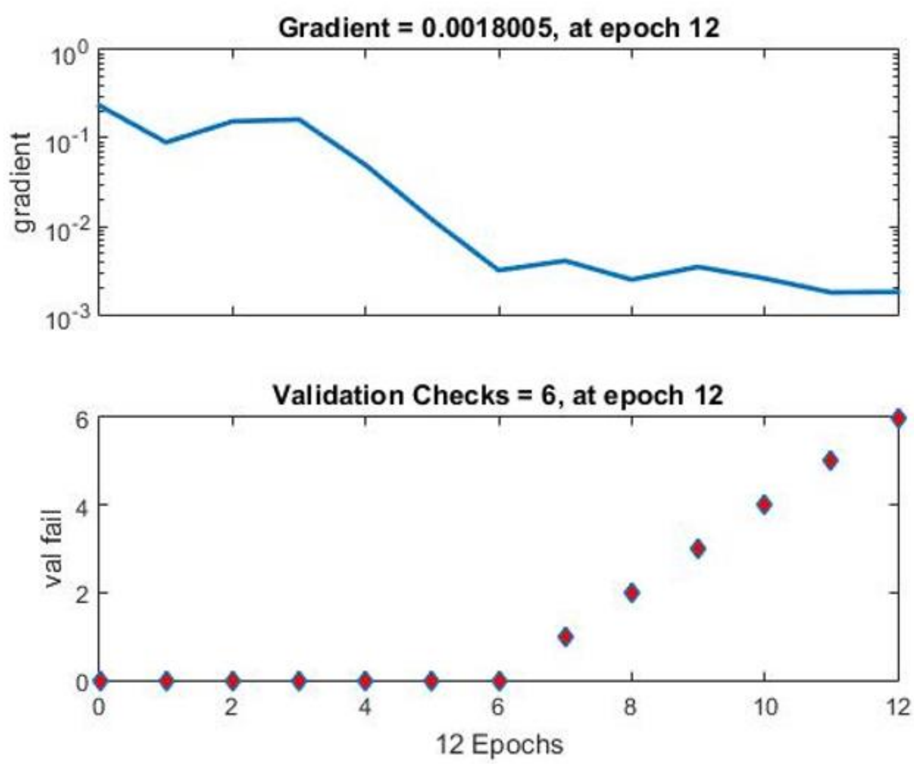


Fig 5.3 Regression plot

.

5.2 CONCLUSION:

This model presents the offline handwritten signature verification using Artificial neural network (ANN)approach. This method uses featuresextracted from preprocessed signature images. The extracted features are used to train a neural network using feed forwardback propagation training algorithm. The network could classify all genuine and forged signatures correctly, when the network was presented with signaturesamples from database different than the ones used in training phase.

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, by using only geometrical offline features.

Moreover, the use of large databases is not required to show the capability of learning for this sort of problem, we have chosen only five genuine signatures and three 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%, although the threshold, the parameter deciding the genuineness of an image, is 90%. 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

CHAPTER 6

REFERENCES

6.1 References:

1. Hafemann, L.G.; Sabourin, R.; Oliveira,L.S. Offline handwritten signatureverificationLiteraturereview.In Proceedings of the 2017 SeventhInternational Conference on Image Processing Theory, Tools andApplications (IPTA), Institute of Electrical and Electronics Engineers (IEEE), Montreal, QC, Canada,28 November–1 December 2017; pp. 1–8.

2. Alvarez, G.; Sheffer, B.; Bryant, M. Offline Signature Verification withConvolutional Neural Networks; TechnicalReport; Stanford University: Stanford, CA, USA, 23 March 2016.

3. Zhang, X.-Y.; Bengio, Y.; Liu, C.-L. Online and offline handwritten Chinese character recognition:A comprehensive study and new benchmark. Pattern Recognit. 2017, 61, 348–360. [CrossRef]

4. Diaz, M.; Ferrer, M.A.; Impedovo, D.; Malik, M.I.; Pirlo, G.; Plamondon, R. A perspective analysis ofhandwritten signature technology. ACM Comput. Surv.2019, 51, 1–39. [CrossRef]

5. Simonyan, K.; Vedaldi, A.; Zisserman,A. Deep inside convolutional networks: Visualising image classificationmodels andsaliency maps. arXiv 2013, arXiv:1312.6034.

6. Bouamra, W.; Djeddi, C.; Nini, B.; Diaz, M.; Siddiqi, I. Towards the design ofan offline signature verifierbased on a smallnumber of genuine samples for training. Expert Syst. Appl. 2018, 107, 182–195. [CrossRef]

7. Hafemann, L.G.; Sabourin, R.; Oliveira,L.S. Meta-Learning for fast classifier adaptation to new users of signature verification systems. IEEE Trans.Inf. Forensics Secur. 2020, 15, 1735–1745.[CrossRef]Appl. Sci. 2020, 10, 3716 14 of15

8. Shah, A.S.; Khan, M.N.A.; Shah, A. Anappraisal of off-line signature verification techniques. Int. J. Mod.Educ. Comput. Sci.2015, 7, 67–75. [CrossRef]

9. Leclerc, F.; Plamondon, R. Automatic signature verification: The state of the art— 1989–1993. In Proceedingsof the Progress in Automatic Signature Verification, Montreal, QC, Canada, 21–23 June 1994; pp. 3–20.

10. Impedovo, D.; Pirlo, G.; Plamondon,R. Handwritten signature verification: New advancements and openissues. In Proceedings of the 2012 International Conference on Frontiers in Handwriting Recognition,Institute of Electrical and Electronics Engineers (IEEE), Bari, Italy, 18–20 September 2012; pp. 367–372.

11. Plamondon, R.; Srihari, S. Online and off-line handwriting recognition: A comprehensive survey. IEEE Trans.PatternAnal. Mach. Intell. 2000, 22, 63–84. [CrossRef]

12. Deng, P.S.; Liao, H.-Y.M.; Ho, C.W.; Tyan, H.-R. Wavelet-based off-line handwritten signature verification.Comput.Vis. Image Underst. 1999, 76, 173–190.[CrossRef]

13. Pal, S.; Alaei, A.; Pal, U.; Blumenstein, M. Performance of an off-linesignature verification method based ontexture features on a large indic-script signature dataset. In Proceedings of the 2016 12th IAPR Workshop onDocument Analysis Systems (DAS), Santorini, Greece, 11–14 April 2016.

14. Khalajzadeh, H.; Mansouri, M.; Teshnehlab, M. Persian signature verification using convolutional neuralnetworks. Int. J. Eng. Res. Technol.