Offline signature fraud detection using deep learning

Submitted in the Partial Fulfilment of the Requirements of the Award of the Degree

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

V.V. Sri Sravya (20W61A0581)

Under the Esteemed Guidance of

Mrs T. Nagamani Assistant professor

Department of Computer Science & Engineering

SRI SIVANI COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Permanently Affiliated to JNTUGV, Kakinada, CC: W6,'A' Grade by AP Knowledge Mission, ISO 9001:2015 Certified campus)

Chilakapalem Jn,. Etcherla (M) Srikakulam (Dist),
A.P, India.

(2020-2024)

SRI SIVANI COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Permanently Affiliated to JNTU-GV ,

'A' Grade by AP Knowledge Mission, ISO 9001:2015 Certified campus)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that this project work entitled with "OFFLINE SIGNATURE FRAUD DETECTION USING DEEP LEARNING"is the bonfide work carried by V.V. Sri Sravya (20W61A0581) submitted in partial fulfilment of the requirements for the Award of the degree of BACHELOR OF TECHNOLOGY in computer science and engineering During the year 2020-2024.

Project Guide	**Head of department**
(T.Nagamani)	**(Dr.M.Murali krishna)**
Asst.prof	

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

It is indeed with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We feel elated in manifesting our sense of gratitude to our guide, **Mrs.T.Nagamani**, Assistant Professor in the Department of Computer Science and Engineering for his valuable guidance. He has been a constant source of inspiration for us and we are very deeply thankful to him for his support and invaluable advice. We would like to thank our Head of the Department **Dr. M.Murali krishna**, for his constructive criticism throughout our project. We are highly indebted to Principal **Dr. Y Srinivasa Rao**, for the facilities provided to accomplish this project. We are extremely grateful to our Departmental staff members, lab technicians and non-teaching staff members for their extreme help throughout our project. Finally we express our heartfelt thanks to all of our friends who helped us in successful completion of this project.

Presented by

V.V. Sri Sravya (20W61A0581)

DECLARATION

I do hereby declare that the work embodied in this project report entitled " **OFFLINE SIGNATURE FRAUD DETECTION USING DEEPLEARING** " is the outcome of genuine research work carried out by me under the direct supervision of **Mrs.T.Nagamani** Assistant Professor, Department of Computer Science Engineering and is submitted by us to Sri Sivani College of Engineering. The work is original and has not been submitted elsewhere for the award of any other degree or diploma.

Presented by

V.V. Sri Sravya (20W61A0581)

INDEX

S.NO	CONTENTS	Page no
	ABSTRTACT	5
	CHAPTER-1	
1.1	INTRODUCTION	6-7
1.2	PROPOSED SYSTEM	8
1.3	OBJECTIVE OF THE PROJECT	8-10
1.4	MOTIVATION OF THE PROJECT	11
	CHAPTER-2	
2	LITERATURE REVIEW	12-15
	CHAPTER-3	
	SYSTEM ANALYSIS	
3.1	SOFTWARE DEVELOPMENT LIFE CYCLE	16-17
3.2	STUDY OF THE SYSTEM	
3.3	SYSTEM REQUIREMENTS HARDWARE REQUIREMENTS	18
3.4	INPUT DESIGN AND OUTPUT DESIGN	19-25
3.5	ACTIVITY DIAGRAM	26
	CHAPTER-4	
	FESABLITY STUDY	
4	TECHNICAL FEASIBLITY	27-28
	CHAPTER-5	
5	OPERATIONAL FEASIBLITY	28
	CHAPTER-6	

	METHODOLOGY	
6.1	PREPROCESSING	29
6.2	THEORTICAL APPROACH	29-30
6.3	PREPROCESSING INVOLVES THE FOLLOWING STAGES	31-39
6.3.1	FILTERING	31
6.3.2	BINARIZATION	32
6.3.0	CONVERSION FROM COLOR IMAGE (RGB) GRAY LEVEL IMAGE	32
6.3.2.1	CONVERSION OF GRAY LEVEL IMAGE INTO BINARY IMAGE	34
6.3.2.2	CROPPING	34
6.3.2.3	THINNING	35
6.3.2.4	IMAGE RESIZING	35
	BOUNDING BOX OF THE SIGUNATURE	
	FEATURE EXTRACTION	
	ASPECT RATIO	
	HORIZONTAL AND VERTICAL CENTER OF THE SIGNATURE	
	EVALUATION PROCESS OF SUBSET OF FEATURES	
	CHAPTER-7	
	SOFTWARE ENVIRONMENT	

	WHAT IS PYTHON	
	HISTORY OF PYTHON	
	HOW TO INSTALL PYTHON ON WINDOWS AND MAC	
	PACKAGES AND LIBRARIES USED	

	CHAPTER-8	
	SYSTEM TESTING AND IMPLEMENTATION	
8.1	STRATEGIC APPROACH TO SOFTWARE TESTING	55
8.2	UNIT TESTING	56
8.2.1	WHITE BOX TESTING	56
8.2.2	BASIC PATH TESTING	56
8.2.3	CONDITIONAL TESTING	57
8.2.4	DATA FLOW TESTING	57
8.2.5	LOOP TESTING	57
8.2.6	TEST CASES	58
	CHAPTER-9	
9	RESULTS & SCREEN SHOTS	59-61
	CHAPTER-10	
10	CONCLUSION	62
	CHAPTER-11	
11	REFERENCES	63-64

LIST OF FIGURES

S.NO	CONTENT	PAGE NO
1.1	DATABASE USED	9

1.2	SIGNATURES OF DIFFERENT PERSONS	10
1.3	DATASET USED FOR TRAINING AND TESTING	10
3.1	USE CASE DIAGRAM	25
3.5	ACTIVITY DIAGRAM	26
6.1	3*3 WINDOW	31
6.2	WINDOW AFTER NOISE REMOVED	31
6.3	SIGNATURE IMAGE WITH NOISE FILTERING	32
6.4	SIGNATURE IMAGE AFTER FILTERING	32
6.5	A GRAY LEVEL SIGNATURE IMAGE	34
6.6	A BINARY SIGNATURE IMAGE	34
6.7	AN UNCROPPED IMAGE	34
6.8	A CROPPED IMAGE	34
6.9	THINNING IMAGE	35
6.10	RESIZED IMAGE	36
6.11	BOUNDING BOX OF AN IMAGE	36
6.12	A SIGNATURE IMAGE SHOWING HEIGHT AND WIDTH	40

6.17	FLOW CHART OF FEATURE SELECTION PROCESS	42
6.18	OPERATIONS IN FILTER METHOD	44
6.19	THE PROPOSED SYSTEM	46
9	SAMPLES IMAGES FOR SIGNATURE IN DATASET	59
9.1	CNN MODEL WITH PARAMETERS	60
9.2	RESULT OF THE INPUT IMAGE	60

LIST OF TABELS

S.NO	CONTENT	PAGE NO
8.2.6	TEST CASES	58
9.3	MODEL ARCHITECTURE	61

Signature fraud detection using deep learning

ABSTRACT

signature are used many important documents such bank cheque, passport, driving license, etc. and can be faked in multiple ways. This creating many problems such as fake identifications, identify theft, hacking etc. To reduce this issue, our project is focused on developing a system for detecting whether a signature is real or fake from dataset of signatures using CNN and Deep learning. The reason we are using CNN and deep learning is because signature change over a period of time based on multiple behavioral changes such age, state of mind, physical health etc. We require a system that can learn from multiple training datasets and increase its accuracy of detection. There are two types of signatures authentication methods, which are online signature and offline signature verification methods. Our project is based on offline signatures forgery detection method .This type of signatures are handwritten on the documents and require an image of the signature. This is why we also should consider image processing for this project. We are referencing a few papers which implement the project using a few methods for both online and offline signature forgery detection methods based on deep learning models, we plan on implementing the offline methods and try to achieve a better accuracy..

Keywords Deep CNN,offline signature, forgery detection, deep learning

CHAPTER 1

Introduction

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. Online data 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:Signature acquisition, Preprocessing, Feature extraction and Classification. Initially Offline signatures of different persons with multiple samples has taken 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 information captured during acquisition. These dynamic characteristics 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 features In 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 by classify Signatures either genuine or forge. The normalised vector is given as input to the Neural Network which is trained using Feed Forward-Back Propagation algorithm and signatures are tested to classify whether signature is genuine or forged.

1.3 Objective 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.

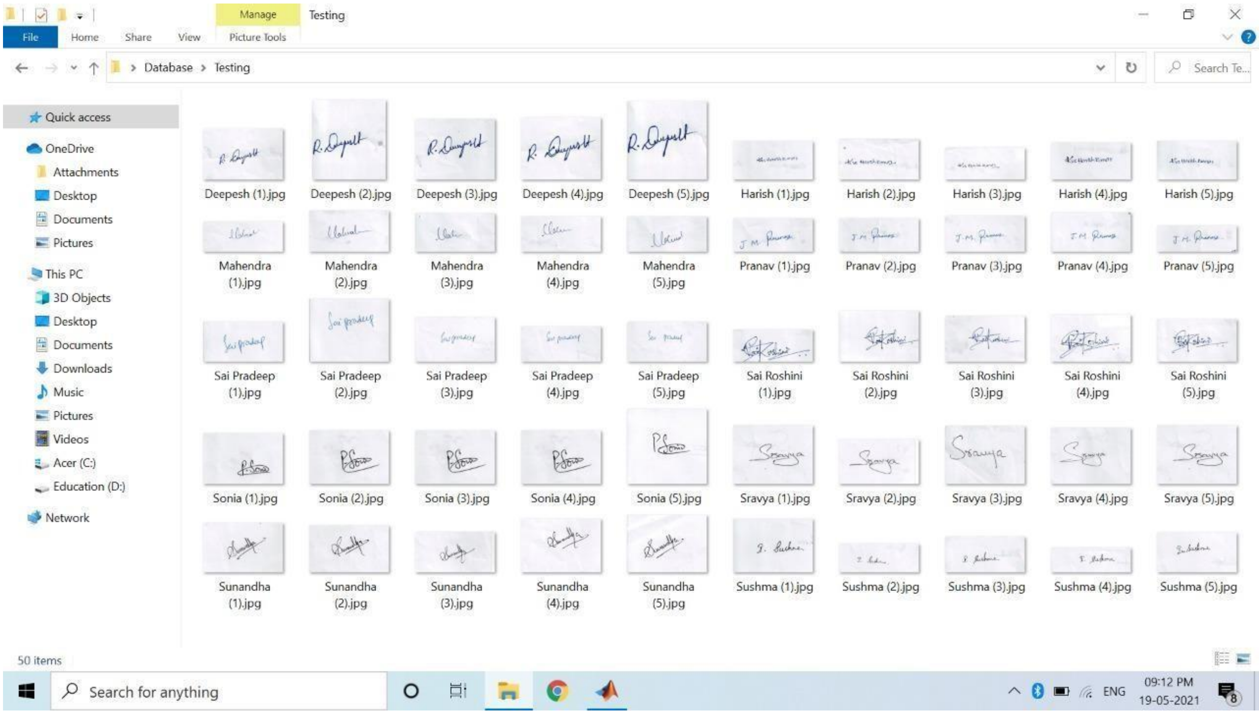
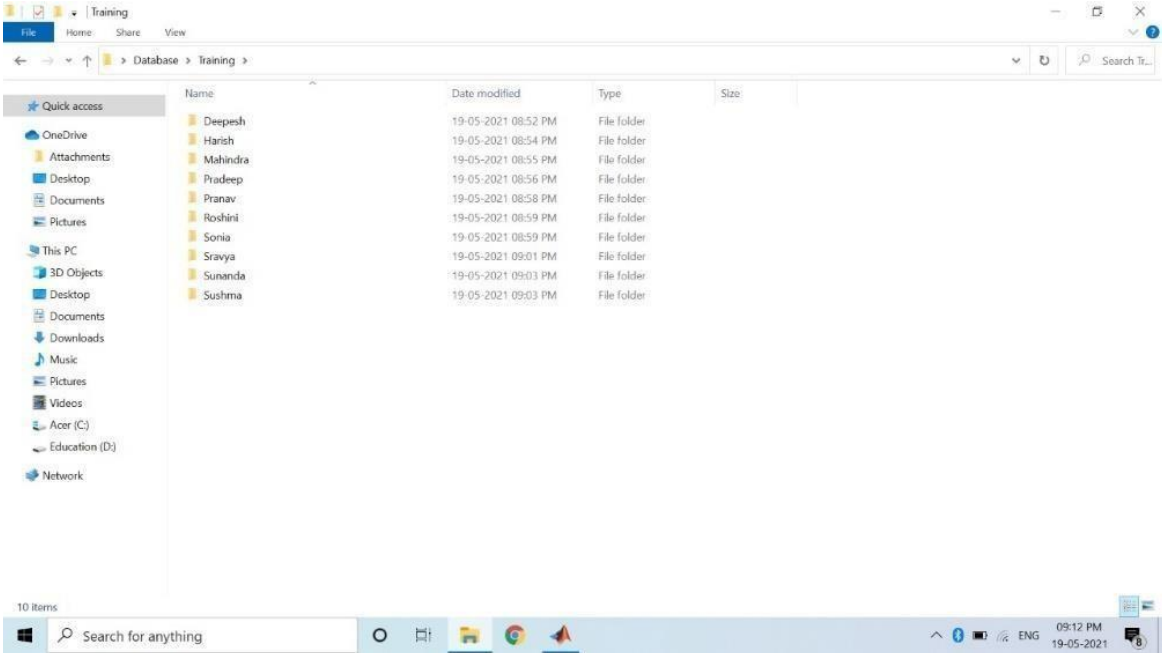
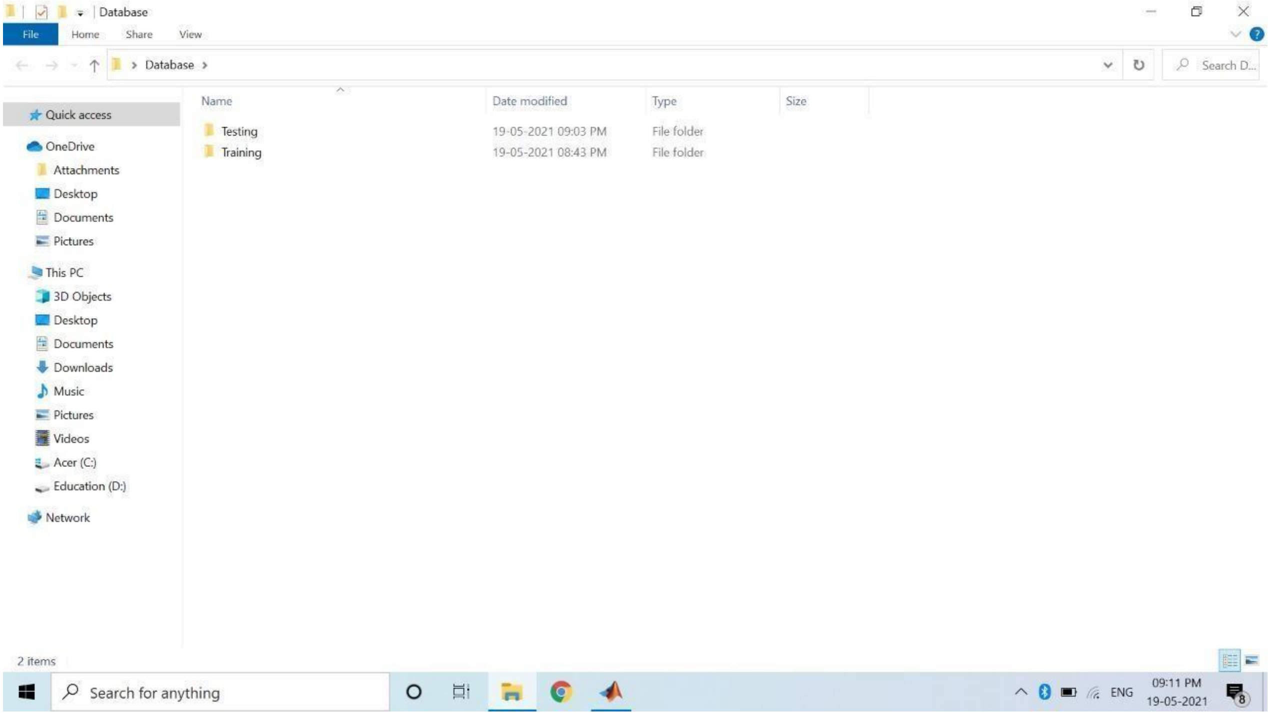


Fig 1.1 . Database used



Name	Date modified	Type	Size
Deepesh	19-05-2021 08:52 PM	File folder	
Harish	19-05-2021 08:54 PM	File folder	
Mahindra	19-05-2021 08:55 PM	File folder	
Pradeep	19-05-2021 08:56 PM	File folder	
Pranav	19-05-2021 08:58 PM	File folder	
Roshini	19-05-2021 08:59 PM	File folder	
Sonia	19-05-2021 08:59 PM	File folder	
Sravya	19-05-2021 09:01 PM	File folder	
Sunanda	19-05-2021 09:03 PM	File folder	
Sushma	19-05-2021 09:03 PM	File folder	

Fig 1.2 Signatures of different persons



Name	Date modified	Type	Size
Testing	19-05-2021 09:03 PM	File folder	
Training	19-05-2021 08:43 PM	File folder	

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

System Requirement:

REQUIREMENT ANALYSIS

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

CHAPTER 2

Literature Review

This paper presents previous work in the field of signature and writer identification to show the historical development of the idea and defines a new promising approach in handwritten signature identification based on some basic concepts of graph theory. This principle can be implemented on both on-line handwritten signature recognition systems and off-line handwritten signature recognition systems.

In this paper, we propose a signature verification system based on Dynamic Time Warping (DTW). The method works by extracting the vertical projection feature from signature images and by comparing reference and probe feature templates using elastic matching. Modifications are made to the basic DTW algorithm to account for the stability of the various components of a signature.

In this article, a new approach to offline signature verification, based on a general-purpose wide baseline matching methodology, is proposed. Instead of detecting and matching geometric, signaturedependent features, as it is usually done, in the proposed approach local interest points are detected in the signature images, then local descriptors are computed in the neighborhood of these points, and afterwards these descriptors are compared using local and global matching procedures

Elias N. Zois , Dimitrios Tsourounis , Ilias Theodorakopoulos , Anastasios L. Kesidis, and George Economou, during this paper, a feature extraction method for offline signature verification is presented that harnesses the facility of sparse representation (SR) so as to deliver state-of- the-art certify attainment in number of signature datasets like CEDAR, MCYT-75, GPDS, and UTSIG. Beyond the accuracy improvements, number of major parameters allied to SR; like selected configuration, dictionary size, sparsity level, and positivity priors are analysis. Besides, it's evinced that second order statistics of the sparse codes may be a powerful pooling function for the formation of the worldwide signature descriptor. Also, a radical evaluation of the consequences of preprocessing is introduced by an automatic algorithm so as to pick the optimum thinning level. Finally, a segmentation strategy which employs a special kind of spatial pyramid tailored to the matter of SR is presented in conjunction with the enhancing of the produced descriptor on relevant areas of the signature as arrive from the binary robust invariant scalable key-point identify mechanism.

Victor L. F. Souza, Adriano L. I. Oliveira, Robert Sabourin, in this work it is investigated whether the use of these CNN features provide good results in a writer- independent (WI) HSV context, based on the dichotomy transformation combined with the utilization of an SVM writer-independent classifier. The praised attainment within the Brazilian and GPDS datasets indicate that (i) the proposed approach outperformed other WI-HSV steps from the literature, (ii) within the worldwide threshold scenario, the proffer approach was ripe to outperform the writer-dependent method with CNN features within the Brazilian dataset, (iii) in an user threshold scenario, the results are almost like those obtained by the writer-dependent method with CNN features.

Muhammed Mutlu Yapıcı, Adem Tekerek, Nurettin Topaloglu, during this study, we proposed a Deep Learning (DL) based offline signature verification method to stop signature fraud by malicious people. The DL method used in the study is the Convolution Neural Network (CNN). CNN was designed and trained separately for 2 different models such one Writer Dependent (WD) and therefore the other Writer Independent (WI). The experimental outcome showed that WI has 62.5% of success and WD has 75% of success. It is predicted that the achievement of the obtained outcome will increase if the CNN step is supported by adding another feature extraction steps.

Elias N. Zois, Ilias Theodorakopoulos, Dimitrios Tsourounis, in this work, sparse dictionary learning and coding are for the first time employed as a means to provide a feature space for offline signature verification, which intuitively adapts to alittle set of randomly selected genuine reference samples, thus making it attractable for forensic cases. In this context, the K-SVD dictionary learning algorithm is used so as to make a writer oriented lexicon. For any signature sample, sparse representation with the utilization of the writer's lexicon and therefore the Orthogonal Matching Pursuit algorithm generates a weight matrix; features are then extracted by applying easy average pooling to the new sparse codes. The act of the proposed scheme is demonstrated using the favored CEDAR, MCYT75 and GPDS300 signature datasets, delivering state of the art results.

Wang Kai, Liu Jingzhi, Xu Shun, Wang kai, Gan Zhichun, In this paper, we study a fast and computationally efficient sparse representation classification scheme for battlefield textual information in which the block sparsity of sparse coefficients is exploited. A novel sparse approximation algorithm tailored for this low complexity classification method is proposed. Experiment output show that our classification algorithm that leverages the canny structure of the textual information outperforms plain canny representation classification procedures in all classification accuracy an computationally efficiency.

A. Hamadene and Y. Chibani, In this paper, we propose a one-class writer-independent system using feature dissimilarity measures (FDM) thresholding for classification and a reduced number of references. The proposed system involves the use of Contourlet Transform (CT) based directional code co-occurrence matrix (DCCM) feature generation method. The verification is achieved through a WI threshold which is automatically selected employing a new signature stability criterion. The proposed writer independent concept is besides addressed through the mixture of different writers' datasets in both design and verification stages. Experimental results show the effectiveness of the proposed system in spite of the strict verification protocol using the one class concept, a singular threshold for accepting or rejecting a questioned signature, the reduced number of writers and therefore the limited number of reference signatures.

Shih-Chung, Hsu, Chung-Lin, Huang, This paper proposes an object verification method in two different views by using sparse representation. The proposed method contains three major modules. First, we train the sparse matrix by using K-Singular Valued Decomposition (K-SVD) and therefore the maximum correlation training sample selection. Second, we project the training samples onto the sparse matrix to get the parse vector training set. Third, we combine two training sets of the same/different objects from two different views to generate positive/negative hybrid sparse vector sets for SVM classifier training. Our contributions in this paper are

(1) proposing a better dictionary representation learning than original K-SVD learning, and (2) developing an optimal sparse representation for object verification with very good accuracy. In the experiment, we show that our method has the better accuracy than the other methods.

Mrs. Madhuri R.Deore, Mrs. Shubhangi M. Handore, The signature identification can be offline or online. We apply the image processing technique for offline signature recognition here no dynamic feature are available in offline identification. A brief survey on various off-line biometric identification & verification schemes is represented this paper.

Amit Kishore Shukla, Pulkit Mohan, Gaurav Ojha, Manoj Wariya, the target of this paper is to process the hand written signature and verify it. For verifying the signatures of a particular person, we have taken n samples of Genuine Signature, signed by that person on a piece of paper. Further we scanned the paper containing the set of signatures. Now we've extracted each of the real signatures of the person and stored it in separate file of the format .bmp. The extraction of the signatures within the last step has been in minimum area to supply accurate area of the signing of signature. We could have matched the signature of each person with the other signature but usually it is almost impossible to produce exactly the same set of signatures. We would verify the signatures on the subsequent parameters allowing a percentage of error in it. Permissible boundary, Hand pressure, Euclidian distance, Center of cylinder generated from minimum spanning tree, Delaunay triangulation of the signature, Angle between base line and center of gravity.

Unnila A. Jain, Prof. Nitin N. Patil, This paper presents current approaches to off-line signature verification with the goal of surveying the most beneficial techniques that are available. This investigation also will introduce techniques which will significantly boost the achieved classification accuracy rate. This paper presents a comparative study of varied approaches of offline signature verification too

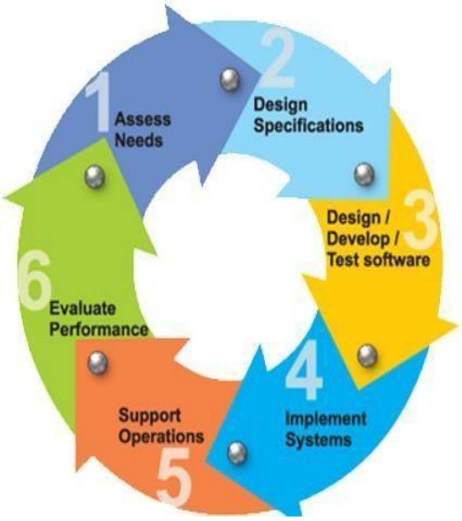
H. Firouzi, M. Babaie-Zadeh, A. Ghasemian Sahebi, C. Jutten, during this paper an extension of the sparse decomposition problem is taken into account and an algorithm for solving it's presented. In this extension, it is known that one of the shifted versions of a signal has a sparse representation on an over complete dictionary, and we are looking for the sparsest representation among the representations of all the shifted versions of. Then, the proposed algorithm finds simultaneously the quantity of the specified shift, and therefore the sparse representation. Experimental results emphasize on the performance of our algorithm.

CHAPTER 3

system analysis

3.1 Software Development Life Cycle

There is various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models". Each process model follows a particular life cycle in order to ensure success in process of software development.



Requirements

Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements. Who is going to use the system? How will they use the system? What data should be input into the system? What data should be output by the system? These are general questions that get answered during a requirements gathering phase. This produces a nice big list of functionality that the system should provide, which describes functions the system should perform, business logic that processes data, what data is stored and used by the system, and how the user interface should work. The overall result is the system as a whole and how it performs, not how it is actually going to do it.

Design

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here) are all part of the deliverables of a design phase.

Implementation

Code is produced from the deliverables of the design phase during implementation, and this is the longest phase of the software development life cycle. For a developer, this is the main focus of the life cycle because this is where the code is produced. Implementation my overlap with both the design and testing phases. Many tools exists (CASE tools) to actually automate the production of code using information gathered and produced during the design phase.

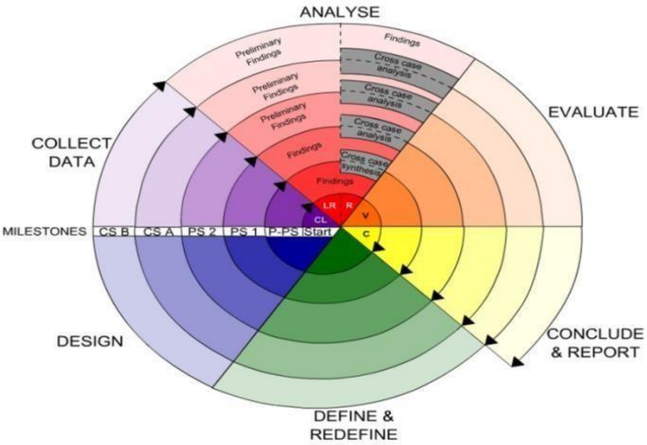
Testing

During testing, the implementation is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole.

3.2 STUDY OF THE SYSTEM

In the flexibility of uses the interface has been developed a graphics concepts in mind, associated through a browser interface. The GUI's at the top level has been categorized as follows

1. Administrative User Interface Design
2. The Operational and Generic User Interface Design The administrative user interface concentrates on the information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The Interface helps the administration with all the transactional states like data insertion, data deletion, and data updating along with executive data search capabilities.



3.3 SYSTEM REQUIREMENTS Hardware Requirements

- RAM: 4GB and Higher
- Processor: Intel i3 and above
- Hard Disk: 500GB: Minimum

Software Requirements

- OS: Windows:
- Python IDE : python 3.7.x and above
- Pycharm IDE Required
- Setup tools and pip to be installed for 3.6 and above
- Language : Python Scripting

3.4 INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user.

Efficient and intelligent output design improves the system's relationship to help user decision-making.

Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output

1. element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives. ✞ Convey information about past activities, current status or projections of the ✞ Future.

✞ Signal important events, opportunities, problems, or warnings.

✞ Trigger an action.

✞ Confirm an action.

Input Design

1. In this system admin can register vm’s information in fog Provider.
2. After that cloud Provider will accept the resource request /reject.
3. User registration.

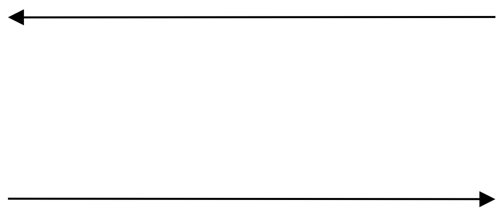
Output design

1. In this place user can login after that he will migrate vm’s from one (A) host to another host(B).
2. He will measure downtime and migration time.

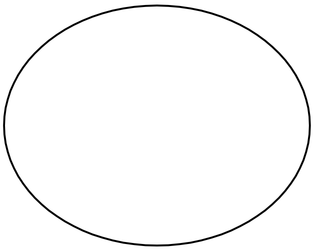
Data Flow Diagrams

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart. DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts. The Basic Notation used to create a DFD’s are as follows:

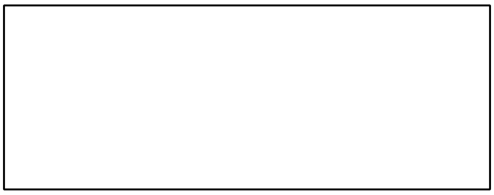
1. **Dataflow:** Data move in a specific direction from an origin to a destination.



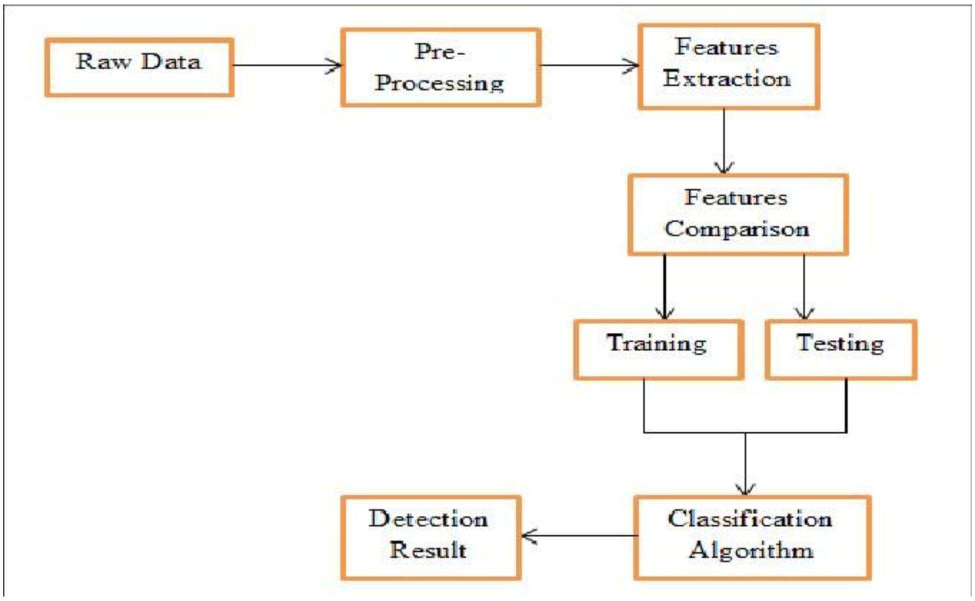
2. **Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.



3. **Source:** External sources or destination of data, which may be People, programs, organizations or other entity



4. **Data Store:** Here data are stored or referenced by a process in the System.



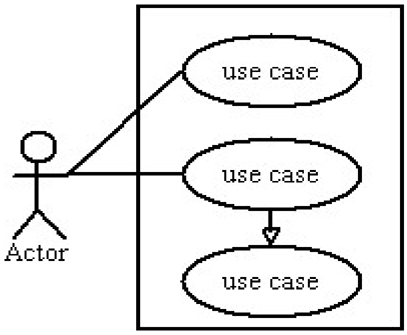
Model diagram

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services or functions provided by the system to its users.

Basic Use Case Diagram Symbols and Notations:

System

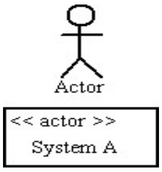
Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



Use Case

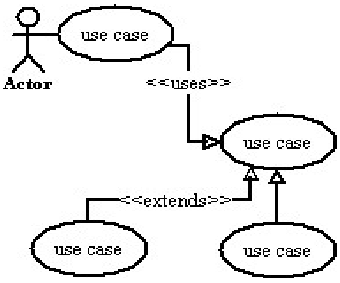
Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

Actors



Relationships

Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



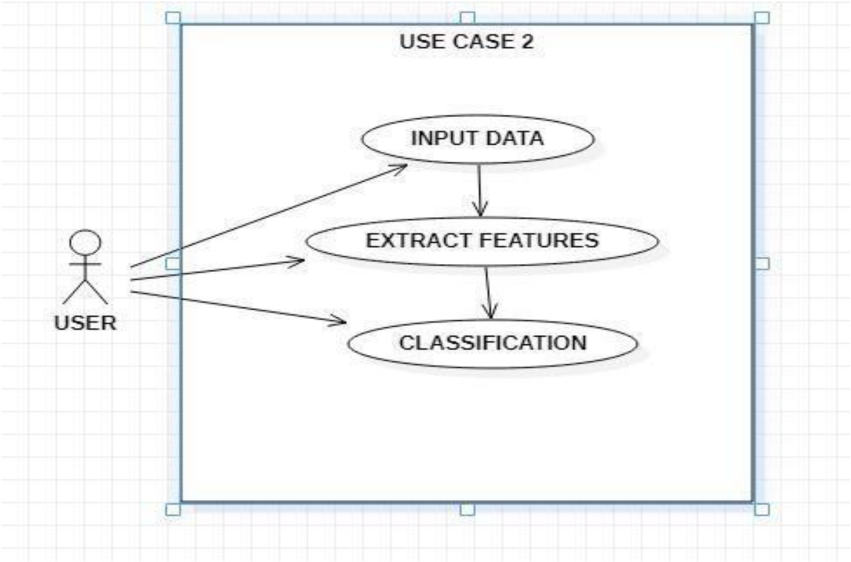
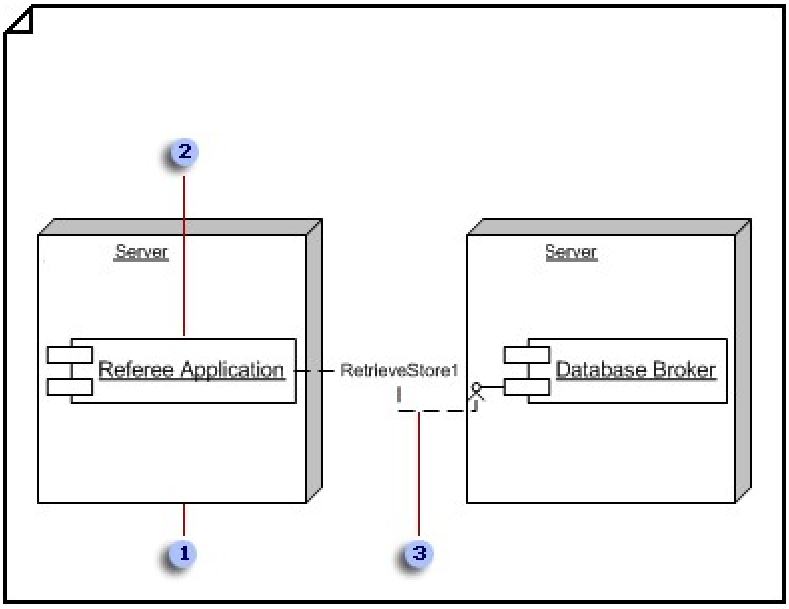


Fig use case diagram

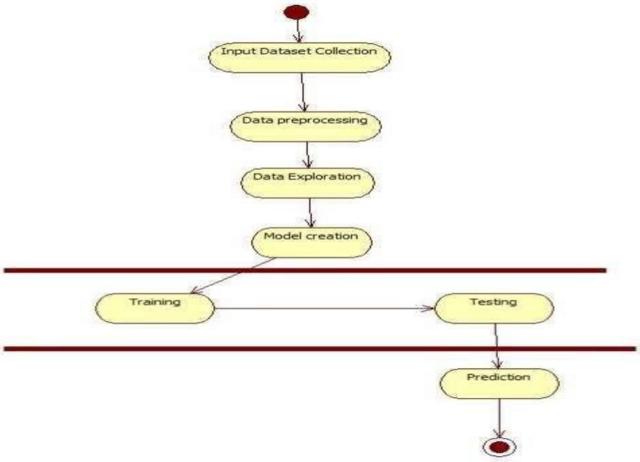
A **sequence diagram** in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



3.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

An activity diagram shows the overall flow of control.



Deployment diagram

CHAPTER 4

Fesablity Study

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economical Feasibility

4.TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?
- Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.
- Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

CHAPTER 5

OPERATIONAL FEASIBILITY

- Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: - • Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits? This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

  The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

CHAPTER – 6

METHODOLOGY

6.1 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 preprocessing stage improves quality of the image and makes it suitable for feature extraction

6.2 Theoretical 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 I in which the signature part consisting of higher gray levels forms the foreground.

$Idi(i,j) = Io,max - Io(i,j)$ .... (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,

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

$Ira(i,j) = Ir(i,j)$ if $Ir(i,j) > 0$

$= 0$ otherwise ……(2)

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

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

6.2.1 Image was segmented using T.

6.2.2 Average gray level values µ1 and µ2 for the two groups of pixels was computed.

6.2.3 .... Based on step 3, new threshold value was computed. $T = 0.5 * (\mu1 + \mu2)$ .... (4)

Steps 2 through 4 were repeated until the difference in T in successive iterations was smaller than 0.5.

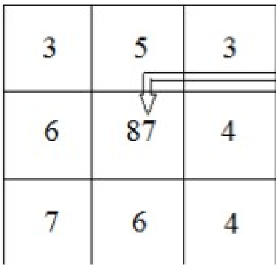
6.3 Preprocessing involoves the following stages:

6.3.1 **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.

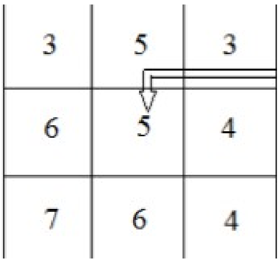
The median filter is a non-linear spatial filter that uses a sub-image area or window. This window is usually of square shape and is of fixed size. This window slides over complete image pixel by pixel and replaces the center value in the window with the median of all the pixel values in the window.

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 6.1 A 3*3 window



3	5	3
6	5	4
7	6	4

Fig 6.2: window after noise removed

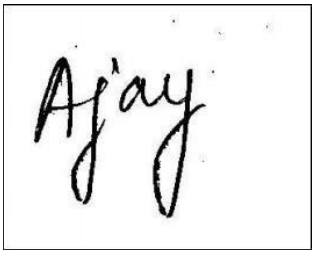


Fig. 6.3. Signature image with noise

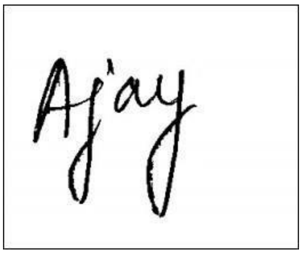


Fig. 6.4. Signature image after filtering

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

6.2.2Binarization:

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

6.3.0 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

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

Lightness method

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

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

I = Calculated gray level

R= Value of Red colour plan G= Value of Green colour plan B= Value of Blue colour plan

(ii) 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

(iii) 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

6.2.2.0 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. 6.5. A gray level signature image

Fig. 6.6 .A binary signature image

6.3.2 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 signature part. Cropping is an essential preprocessing step for all types of classification techniques.

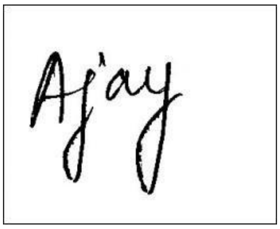


Fig.6.7. An uncropped image

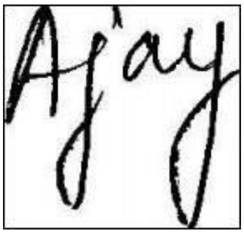


Fig.6.8. A cropped image

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

6.3.3 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].

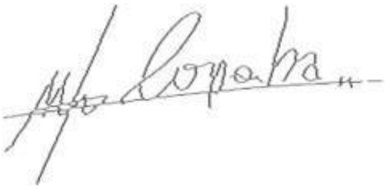


Fig 6.9 Thinning Image

6.3.4 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 6.10.Resized image

6.3.5 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.

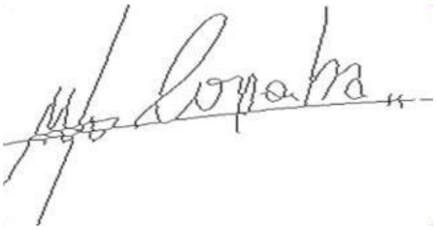


Fig 6.11.Bounding box of an Image

6.3.5.0.1 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].

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

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.

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].

***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].

***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

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:

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

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.

6.4 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

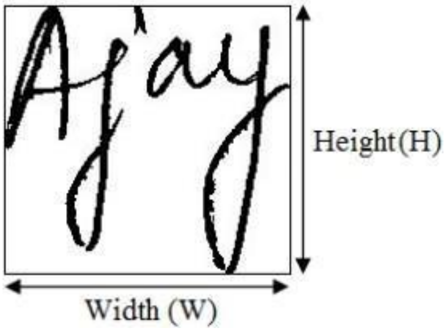


Fig.6.12. A signature image showing height and width

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. Width is 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.

6.5 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 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

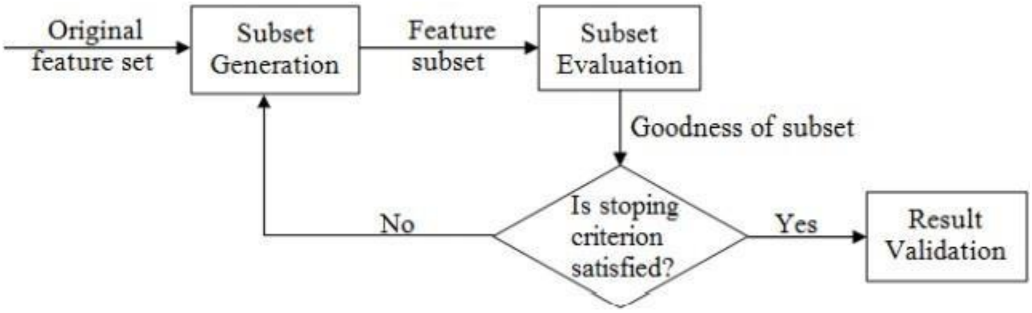


Fig 6.17 Flow chart of Feature selection process

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

6.6 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.

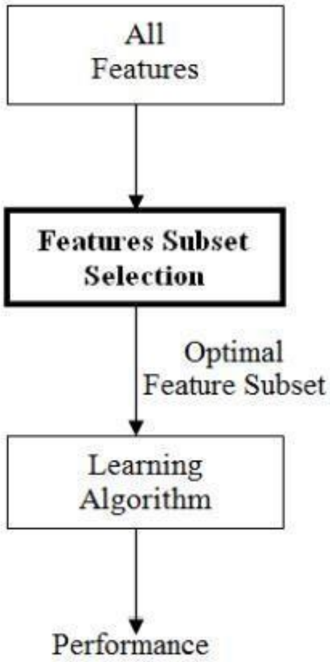
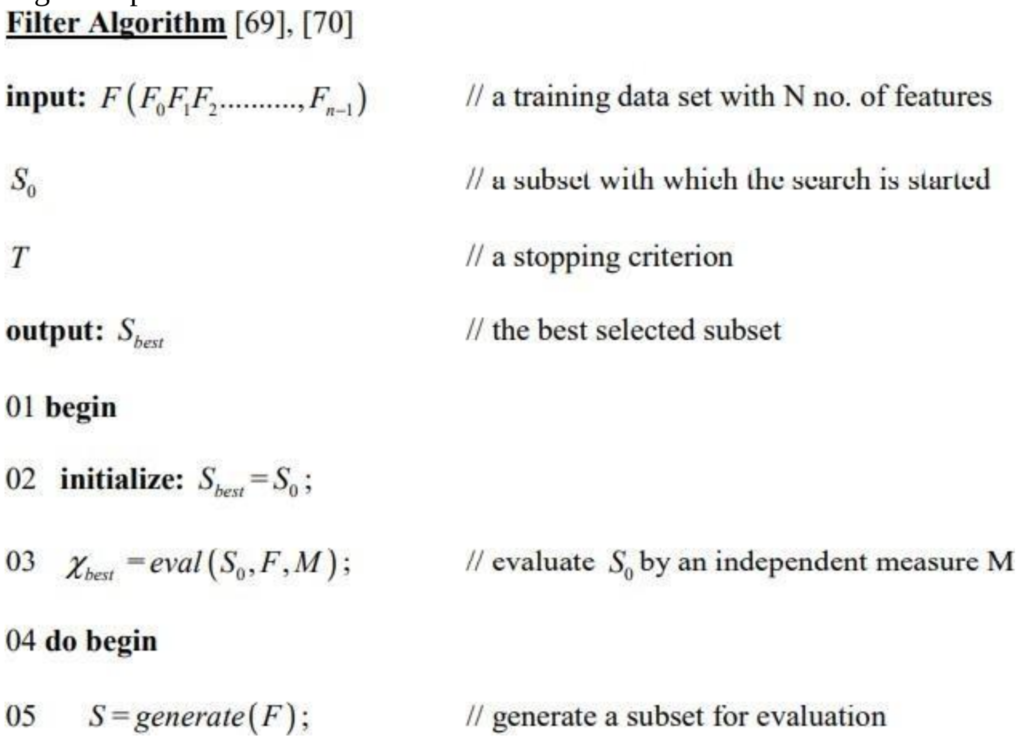
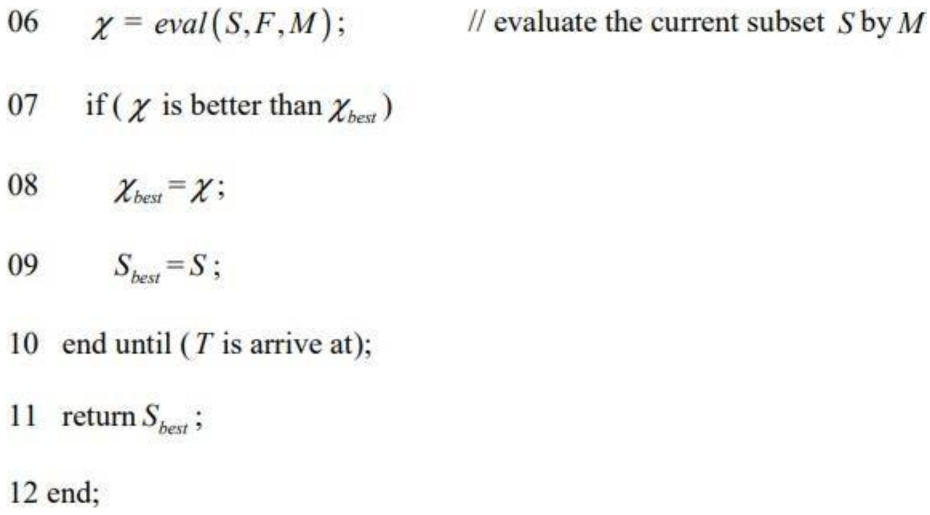


Figure Operations in filter method



Filter Algorithm [69], [70]

```
input: $F(F_0F_1F_2.........., F_{n-1})$        // 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
begin
  initialize: $S_{best} = S_0$;
  $\chi_{best} = eval(S_0, F, M)$;              // evaluate $S_0$ by an independent measure M
do begin
    $S = generate(F)$;                          // generate a subset for evaluation
```



```
        $\chi = eval(S,F,M)$;        // evaluate the current subset $S$ by $M$
    if ( $\chi$ is better than $\chi_{best}$ )
        $\chi_{best} = \chi$;
        $S_{best} = S$;
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.

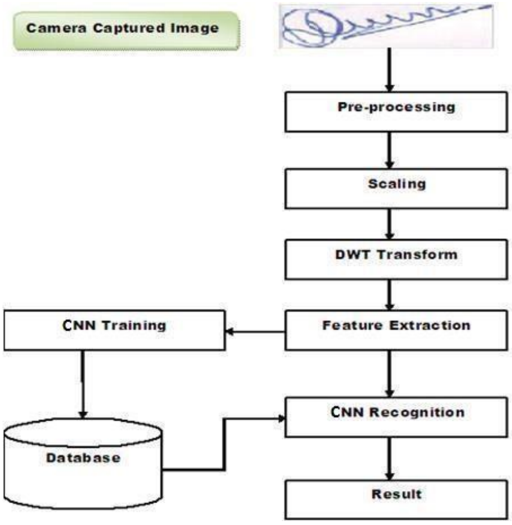
Drawbacks of filter methods: This method doesn't check the relationships between two different features, thus feature correlation is ignored. Therefore there is a chance that redundant features may get selected in filter method; this results poor classification performance

6.4 CNN Algorithm

captured input image is taken for processing and after pre-processing with basic image operation same image is mapped to CNN trained model to get the prediction of which user the image is and if signature is authorized or not.

The proposed system is designed to help these organizations, which are depending on the signatures to complete the documentation and transactions. The basic ideology is integrating the principle of CNN with the input signature image to verify and identify accurate result of signatures.

The proposed system is as shown in fig. It consist of various activities. First the camera



o overcome the problems faced in existing system we are going to implement our system. In this system we are going to use CNN algorithm to get correct output. In this system we are going to minimize the human interpretation error. By using this system, it should be easy to find out the frauds and duplicates signature easily.

A. Algorithm:

CNN is the basic algorithm used in the project which is as follows,

1. Classify dataset under labeled folders with signature samples as CNN is supervised algorithm
2. Read dataset and prepare dataset in one file as pickle or numpy.
3. Read features of all images and label (here name of dataset folder) of it using following functions,

a. Conv2D

b. Maxpool2D

c. Relu activation for layers

d. Sigmoid activation for dense layer

e. Binary Crossentropy for loss calculation

4. Store it in model file
5. Get input image
6. Read features of input image
7. Compare features of stored features
8. Show label as prediction of nearly matched features.

B. Mathematical Model:

Let S be the closed system defined as, S = {Ip, Op, A, Ss, Su, Fi}

Where, Ip=Set of Input, Op=Set of Output, Su= Success State, Fi= Failure State and A= Set of actions, Ss= Set of user's states.

- Set of input=Ip={username, password}

- Set of actions =A={F1,F2,F3,F4,F5,F6} Where,

  - F1= Authentication of user ○ F2 = input the signature image

  - F3 = system detect the object this image ○ F4 = Perform operation Image Processing and Machine Learning

  - F5= Detecting of various sources

  - F6= This result show and stored the database

- Set of user's states=Ss={registration state, login state, selection signature image, classified image, logout}
- Set of output=Op={Show results}

- Su=Success state={Registration Success, Login Success }

- Fi=Failure State={Registration failed, Login failed}

- Set of Exceptions= Ex ={Null Pointer Exception while registration state, Record Not Found (Invalid Password) while login state , Null Values Exception while showing state}

CHAPTER 7

SOFTWARE ENVIRONMENT

USED LANGUAGES

In the implementation of proposed system, we used python as programming language. Python is beginner’s language, which provides various applications. In recent years, python had set the new trend because it is easy to use, interpreted, object-oriented, high-level, scripting language. Python is one of the best languages for the implementing machine learning. It provides rich packages and libraries that used in machine learning.

7.1 What is Python

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

7.2 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.

- **A broad standard library** − Python's bulk of the library is very portable and cross- platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.

- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Advantages of Python Over Other Languages

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

. 3.Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language

Install Python Step-by-Step in Windows and Mac :

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high- level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.The object-oriented approach and language construct provided by

Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

7.3 How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version

3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here.The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

7.4 Packages and libraries used

Some of the python library and packages used in proposed system are as follows:

- ***Numpy***

Numpy is a python library. Abbreviation of Numpy is numerical python library. Numpy package is used for multidimensional arrays and linear algebraic operations. 37 A. RB and S.K. KR Global Transitions Proceedings 2 (2021) 35–41

- ***Pandas***

Pandas is a python library. Pandas is used for data analysis and data manipulation tool. It is used to read the dataset and load the dataset. It is fast, flexible when working with data.

- ***Keras***

Keras is advanced stage of neural network application programming interface (API). It is able of run on top of tensor flow. Keras is mainly used while implementing deep learning algorithms such as CNN, RNN because its user friendly, modularity, and easy to extensibility. It runs on both CPU and GPU. In the experiment of finding the fraud or non fraud credit card transaction we had used Keras along with backend running tensor flow. This Keras along with tenor flow backend makes excellent choice for training neural network architecture.

- ***MySQL***

MySQL is database which is used for storage purpose. In the experiment of fraud identification in card transaction we had used MySQL for storing the user details namely user name, password, email-id and phone number. While entering into application, user needs to register by providing the credential. These credentials are stored in database. Thereafter, user needs to login by giving username and password. The application will validate the login and registered information than user is moved to next window.

CHAPTER 8

SYSTEM TESTINGAND IMPLEMENTATION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

Credit card Fraud Detection using Predictive Modeling

Banks collect a lot of historical records corresponding to millions of customer's transactions. They are credit card and debit card operations, but unfortunately, only a small portion, if any, is open access. Fraud detection is a critical problem affecting large financial companies that have increased due to the growth in credit card transactions . The proposed method consists of the Predictive modeling and Logistic Regression. Now a day's bank transactions as well as credit card frauds increased. One of the most target frauds are credit card fraud, the fraud can occur any type of credit products, such products are retail, home loan and personal loan. During the last few decades as technology has changed, dramatically the face of fraud also changed. To detect credit card fraud, data mining techniques Predictive modeling and Logistic Regression are used. In prediction model to predict the continuous valued functions. Credit card of CSV files will be analyzed to predict the outcome.

In this paper, we propose to detect credit card transaction using available data set and data mining techniques of predictive modeling, Decision tree, and Logistic Regression. Predictive modeling splits the data into two partitions 70% of testing and 30%of training check output class distribution to predict the outcome. The decision tree to get the result as a tree with root node describes the best predictor in the data, the combination of two or more branches is denoted by decision node (non leaf nodes) and each branch represents a value for the attribute which is tested. The leaf node may be 1 in the case of fraud and 0 otherwise. Logistic regression or logistic model is a regression model, where the dependent variable is categorical of a linear generalized model.The rest of the paper is organized as explained. Section

I describes fraud detection methods. Section

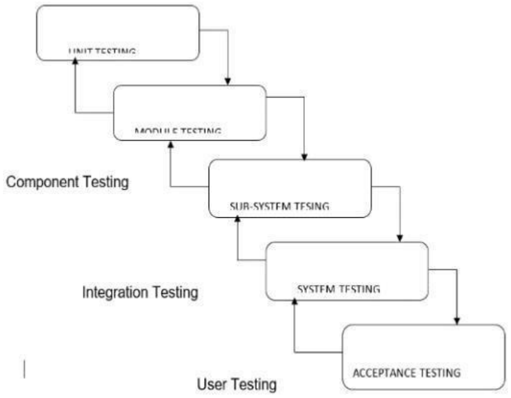
II explains Dataset description for credit card transaction. Section

III consists of experimental results of fraud detection methods, and finally, the conclusion of this work included in section V.

8.1 STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing.



8.2 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel. ***8.2.1. WHITE BOX TESTING***

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

8.2.2. BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were: Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula: V(G)=E-N+2 or V(G)=P+1 or

V (G) =Number Of Regions

Where V (G) is Cyclomatic complexity, E is the number of edges, N is
the number of flow graph nodes, P is the number of predicate nodes.
Determine the basis of set of linearly independent paths.

8.2.3. CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

8.2.4. DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

8.2.5. LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just abo
- ve them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of
- connected loop.Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

8.2.6 TEST CASES

TC ID	Condition Being Tested	Expected Result	Result

1	Check if dataset is taken as input	If taken process next step else throw error	passed
2	check for null values in the dataset	If null values drop the null records	passed
3	Check for futur e extraction and save the file in. pkl format	If file not saved throw the error	passed

CHAPTER 9

Results & Screen shots

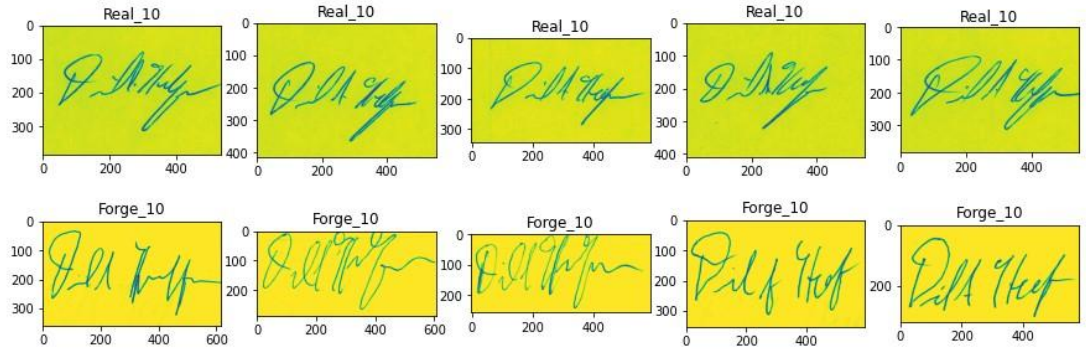
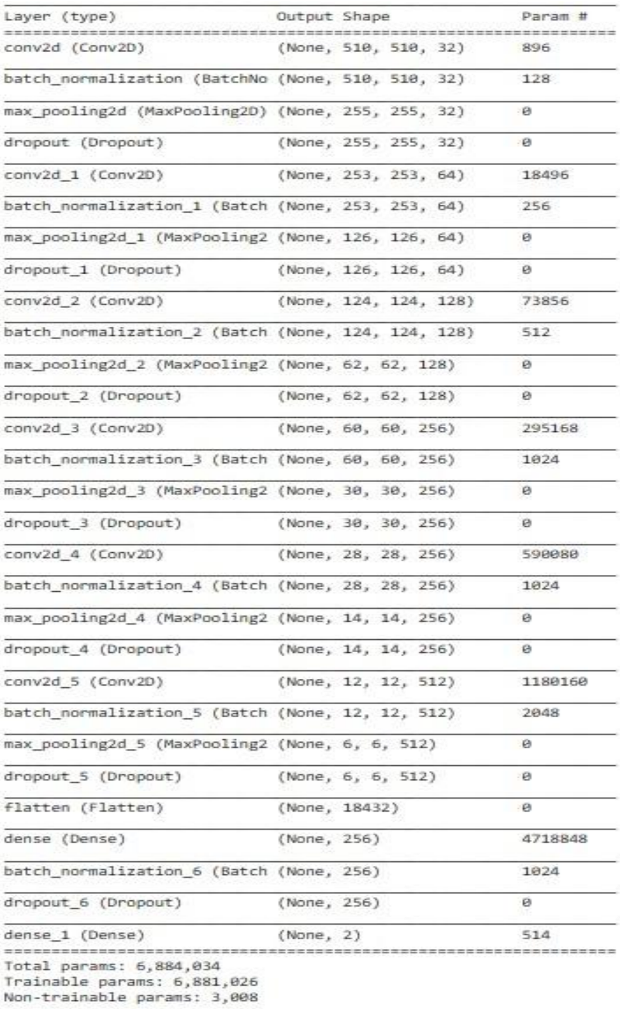
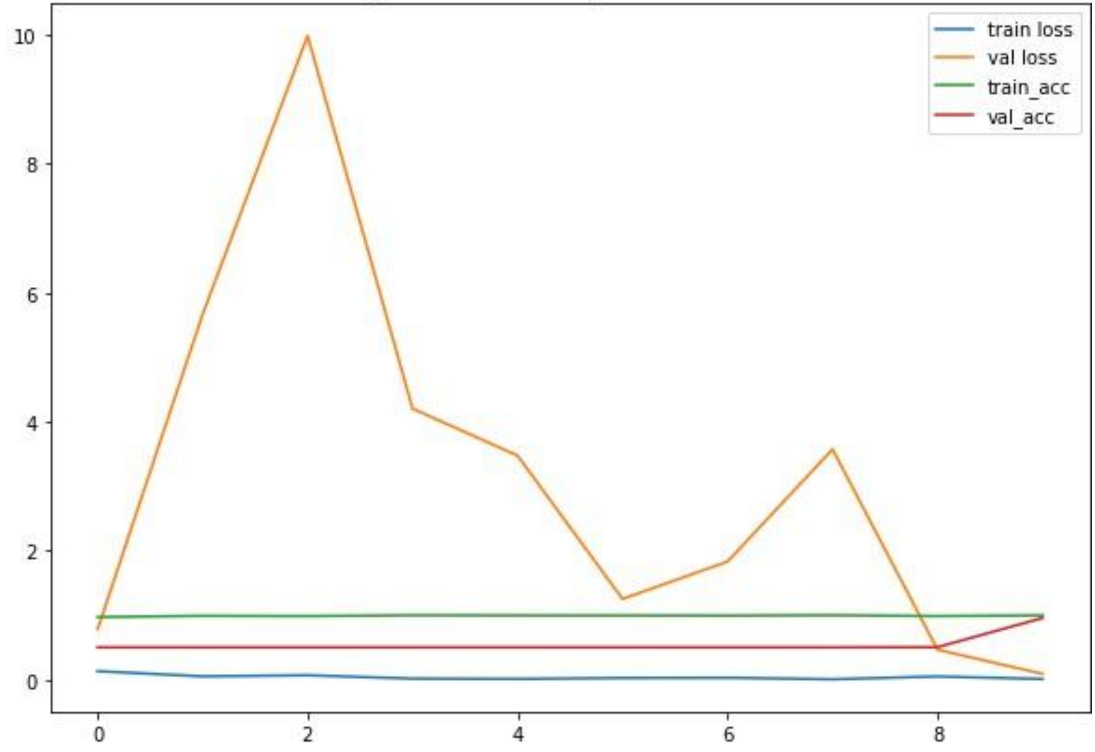


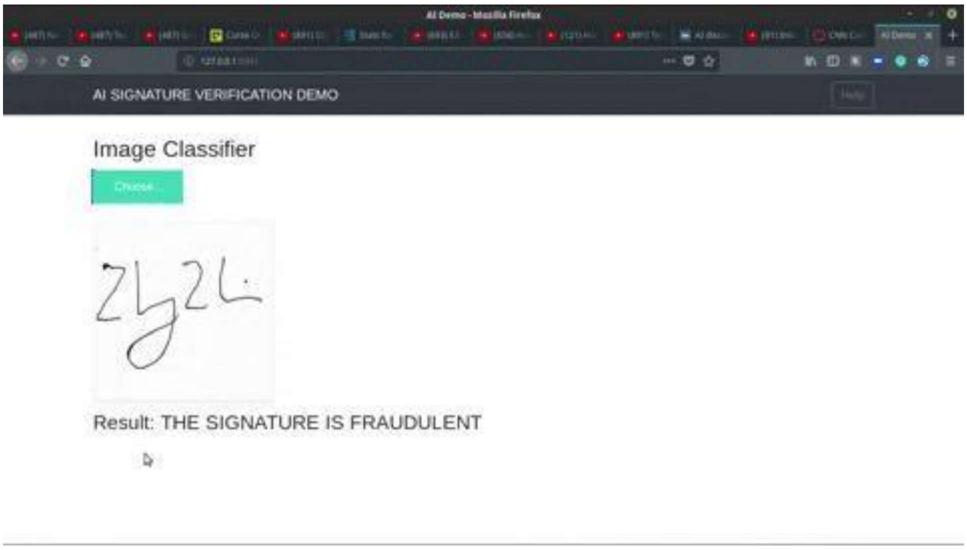
Fig :- sample images for signature in dataset



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 510, 510, 32)      896
_________________________________________________________________
batch_normalization (BatchNo (None, 510, 510, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 255, 255, 32)      0
_________________________________________________________________
dropout (Dropout)            (None, 255, 255, 32)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 253, 253, 64)      18496
_________________________________________________________________
batch_normalization_1 (Batch (None, 253, 253, 64)      256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 126, 126, 64)      0
_________________________________________________________________
dropout_1 (Dropout)          (None, 126, 126, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 124, 124, 128)     73856
_________________________________________________________________
batch_normalization_2 (Batch (None, 124, 124, 128)     512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 62, 62, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 62, 62, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 60, 60, 256)       295168
_________________________________________________________________
batch_normalization_3 (Batch (None, 60, 60, 256)       1024
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 30, 30, 256)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 30, 30, 256)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 28, 28, 256)       590080
_________________________________________________________________
batch_normalization_4 (Batch (None, 28, 28, 256)       1024
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 256)       0
_________________________________________________________________
dropout_4 (Dropout)          (None, 14, 14, 256)       0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 12, 12, 512)       1180160
_________________________________________________________________
batch_normalization_5 (Batch (None, 12, 12, 512)       2048
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 6, 6, 512)         0
_________________________________________________________________
dropout_5 (Dropout)          (None, 6, 6, 512)         0
_________________________________________________________________
flatten (Flatten)            (None, 18432)             0
_________________________________________________________________
dense (Dense)                (None, 256)               4718848
_________________________________________________________________
batch_normalization_6 (Batch (None, 256)               1024
_________________________________________________________________
dropout_6 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 514
=================================================================
Total params: 6,884,034
Trainable params: 6,881,026
Non-trainable params: 3,008
```

Fig 2:-CNN Model with parameters





AI SIGNATURE VERIFICATION DEMO

Image Classifier

Result: THE SIGNATURE IS FRAUDULENT

Fig :- result of the input image

Fig training and validation accuracy and loss graph achived tranning accuracy of 95.88% and validation accuracy 95.55%

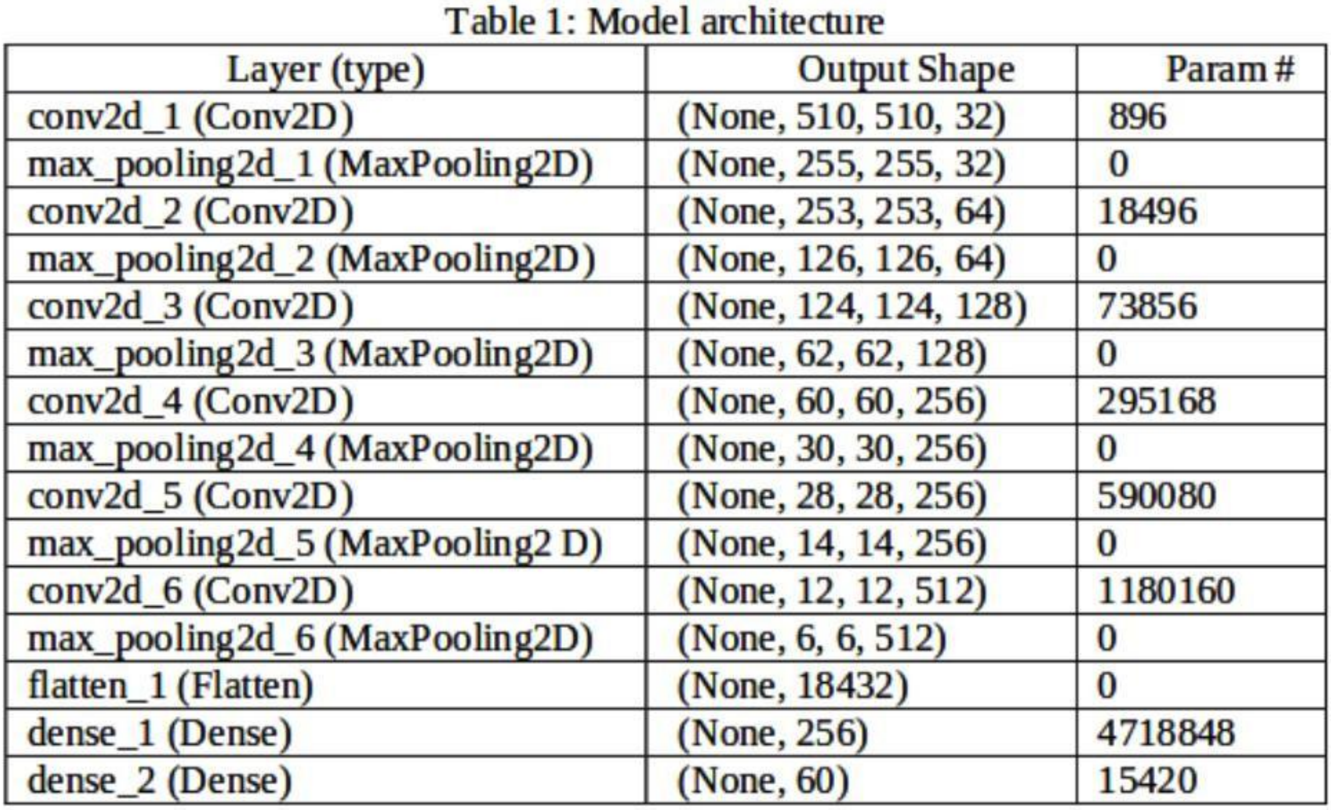


Table 1: Model architecture

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 510, 510, 32)	896
max_pooling2d_1 (MaxPooling2D)	(None, 255, 255, 32)	0
conv2d_2 (Conv2D)	(None, 253, 253, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 126, 126, 64)	0
conv2d_3 (Conv2D)	(None, 124, 124, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 62, 62, 128)	0
conv2d_4 (Conv2D)	(None, 60, 60, 256)	295168
max_pooling2d_4 (MaxPooling2D)	(None, 30, 30, 256)	0
conv2d_5 (Conv2D)	(None, 28, 28, 256)	590080
max_pooling2d_5 (MaxPooling2 D)	(None, 14, 14, 256)	0
conv2d_6 (Conv2D)	(None, 12, 12, 512)	1180160
max_pooling2d_6 (MaxPooling2D)	(None, 6, 6, 512)	0
flatten_1 (Flatten)	(None, 18432)	0
dense_1 (Dense)	(None, 256)	4718848
dense_2 (Dense)	(None, 60)	15420

Fig proposed model layers

CHAPTER 10

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 CNN, is its ability to learn nonlinear problem offline with selective training, which can lead to sufficiently accurate response.

Application of Convolution Neural Network (CNN) 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 CNN application are minimal. And the verification system based on CNN 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 11

References

1. Hafemann,L.G.;Sabourin,R.;Oliveira,L.S.Offline hand written signature verification Literaturere view.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.