CHAPTER-1

INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

Handwritten signatures have been used for centuries as a means of authenticating documents and transactions. In today's digital age, the verification of handwritten signatures plays a crucial role in ensuring the integrity and security of various processes across different sectors. Offline signature verification, a subset of biometric authentication, focuses on the analysis and comparison of handwritten signatures without the need for real-time connectivity or online databases.Signature verification is one of the most utilized pattern recognition tasks. This task is being done by humans till now. Forensic experts involved in signature verification are given with special trainings on the verification process. But still in case of professionally forged signatures, their recognition rate is in between 93% to 99.5%.Machine recognition can give a new dimension in the field of signature verification.In machine recognition, signatures can be analyzed more in detail from a variety of aspects. Throughout this project, we will explore various methodologies, algorithms, and evaluation metrics commonly employed in offline signature verification research. We aim to contribute to the advancement of signature verification technology and its practical implementation in real-world scenarios. Offline signature verification approach can replace the existing manual verification method without disturbing its present mechanism.

This project aims in detecting the forged signature by comparing it with the original signature. It will works where handwritten identification is need without any online interaction. An offline signature verification project can be helpful in various contexts, including, Document authentication, Fraud prevention, Identity verification, Access control, Forensic analysis. Offline signature verification projects can be utilized in various sectors, including, Banking and Finance, Legal compliance, Education, Real estate, Healthcare, Corporate, Government agencies such as legal documents, permits, licenses, and other official paperwork. Overall, offline signature verification projects can enhance security, reduce fraud, and streamline authentication processes in various industries and applications.

1.2 ABOUT ORGANISATION

Karpagam Academy of Higher Education (KAHE) established under Section 3 of UGC Act 1956 is approved by Ministry of Human Resource and Development, Government of India. Dr. R. Vasanthakumar, the president of the trust a philanthropist, industrialist, entrepreneur and culture promoter Contemporary infrastructure, modern teaching methodologies, career-oriented training, excellent placements and the finest faculty have always been Karpagam's hallmark.

Besides technical expertise, the Karpagam Academy of Higher Education (KAHE) has made a mark for itself since its inception by developing communication and soft skills, ensuring enlightening knowledge, extending holistic education and creating a strong value system, Today, with a strength of 6000 students and over 750 teaching & non-teaching staff, the Karpagam Academy of Higher Education (KAHE) is setting new benchmarks in the educational.

MERITS OF KARPAGAM:

Karpagam strives to offer a package of value-added benefits that are tailored to nurture the educational experience of the students.

- Well experienced and trained faculty including 108 doctorates and Post Doctoral Fellows. Visiting faculty from premier institutes like IIM, IISc, IIT, NIT etc.,
- A professional placement department enduring training for overall personality development of students.
- A vibrant Karpagam Research Centre marching towards fruition of innovations and patents.43 patents were filed and 3 patents are granted. 2 of the granted patents are being commercialized.
- Scope to work on projects funded by government & other agencies 10.
- Industrial MoUs and career-oriented courses for enhancing employability.
- Highly vibrant and encouraging academic ambience aiding an enriched education.
- State of the art laboratories and Wi-fi enabled campus with 1Gbps internet connectivity.

RECOGNITION:

- Karpagam Academy of Higher Education was established under Section 3 of UGC Act, 1956
- Approved by the Ministry of Human Resource and Development, Government of India
- Approved by UGC-AICTE to conduct Engineering & Technology programs
- Approved by Council of Architecture to conduct Architecture program
- Approved by Pharmacy Council of India (PCI) New Delhi to conduct Pharmacy courses
- Accredited by NAAC

CHAPTER-2

SYSTEM STUDY

2.1. EXISTING SYSTEM

Existing system contain the many drawbacks. The existing system feature extraction can be done by using LBP (Local Binary Pattern) algorithm. The best feature where selected by using BFS (Breadth First Search) algorithm. Addressing the mentioned drawbacks requires ongoing research and development efforts to improve the accuracy, robustness and efficiency of offline signature verification systems.

2.1.1. DRAWBACKS OF THE EXISTING SYSTEM

- Limited accuracy
- Dependency on quality of signature images
- Waste of time for manual process
- Feature extraction challenges
- Limited dataset to train our algorithm

2.2. PROPOSED SYSTEM

This proposed system we have using CNNs (Convolutional Neural Networks) algorithm for feature extraction. The best subset where selected by using feature selection algorithm.

2.2.1. ADVANTAGES OF PROPOSED SYSTEM

- User Friendly
- Processing time is less
- Enhanced accuracy
- Scalability
- More dataset to train our algorithm
- Best feature selected by feature selection algorithm

2.3. MODULES DESCRIPTIONS

- Database Preparation
- Pre-Processing
- Feature Extraction
- Verification using extracted Features

DATABASE PREPARATION

Signature database plays a significant role in the process of signature verification. But there is no standard offline signature database for the researchers. The major setback faced by the researchers in offline signature verification is the non-availability of a sufficiently large signature database with required quality. Because of privacy issues, not many people agree to make their signatures available to others and that too for practicing forgeries. Therefore, it is still not easy to develop a sufficiently large database with quality signatures. Some limited numbers of offline signature databases are available. But their sizes are smaller and they lack in quality.

In pattern classification, there are two phases - (i) training and (ii) testing. In training phase, the signature classifier is modeled. Sufficient numbers of signature samples are required to model the classifier. The larger database represents a population properly and it helps in producing more reliable results. In most of the pattern recognition techniques, large datasets help in achieving better results. A small training set may result unsatisfactoryverification performance. Lesser number of signature samples against high number of extracted features is a snag in signature verification. Widely accepted and standard benchmark signature database is not yet available for offline signature verification. Therefore, our own signature databases for was developed experimentation.

Data Acquisition:

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

Lighting:

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

Focus:

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

Quality:

In case of a document, the image quality of a scanner is always better than that of a camera.

Scanner Specification:

A scanner is specified by many factors such as Scanner Type, Scanner Resolution, Maximum Resolution, Maximum Scan Area, Scanning Speed, Light Source, Color Bit Depthetc. Our concern was only the Scanner Resolution. Because other parameters do not affectthe essential quality of the signature images required for our purpose.

Scanner Resolution:

Scanner Resolution is a measurement of the resolving power of a scanner. Resolution of a scanner is the measurement of number of pixels that it can sample in the scanned image. It is expressed in dots per inch (dpi). 200 dpi means 200×200 or 40,000 dots per square inch.

If resolution of a scanner is more, it implies that more numbers of dots or pixels are captured per inch of the image by the scanner. Image scanned with a higher resolution can be enlarged more. Resolution of scanned images is also an important factor that influences the process of signature verification. High resolution results more detailed images. But, they need more storage space and may contain noise. Thus computational cost is also higher. On the other hand, computational cost is lower with low resolutions.

Scanner resolution required for offline signature:

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

Signature Collection Procedure:

There must be a proper way of collecting the signature samples. Quality of the signature database largely relies on the signature collection protocol. For producing the forged signatures, the forgers were provided the genuine signatures.To reproduce the signatures, they were allowed to practice the signature as long as they wished. Like the genuine signature samples, the forged signature samples were also collected. But the forged signatures are highly inconsistent. Genuine signature where collected from the genuine person. Forged signatures are collected from lay forgers. We have follow these steps while collecting the signature samples,

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

(ii) Signature images were scanned by a digital scanner with 200dpi resolution.

(iii) All signature images were stored in JPEG format.

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.

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. Following preprocessing steps were used in this project:

(i) 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 unavoidablepreprocessing 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. 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 overcomplete image pixel by pixel and replaces the center value in the window with the medianof all the pixel values in the window.

The pixel value of the window in Fig 2.3.1 (a) is in ascending order. The center value in the window which is possibly a noise, is replaced with the median value, the following new window in fig 2.3.1 (b) is found, where the noise is removed.

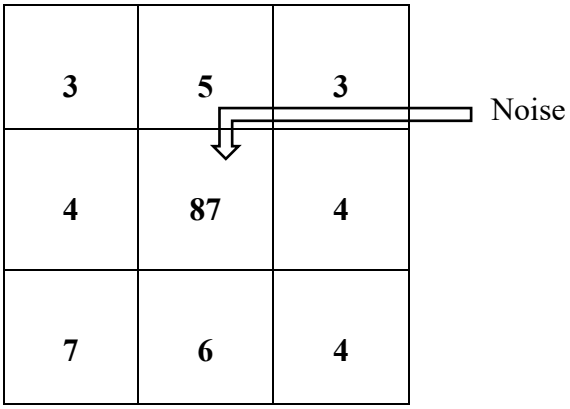

Fig 2.3.1 (a) A 3*3 window

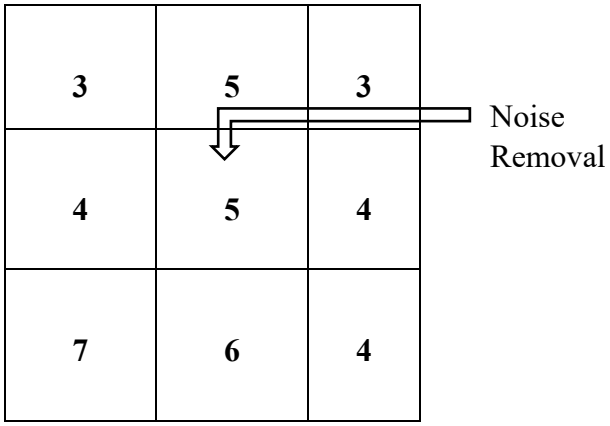

Fig 2.3.1 (b) window after noise removal

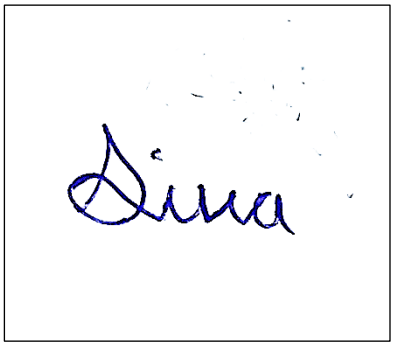

Fig 2.3.2 (a) signature image with noise

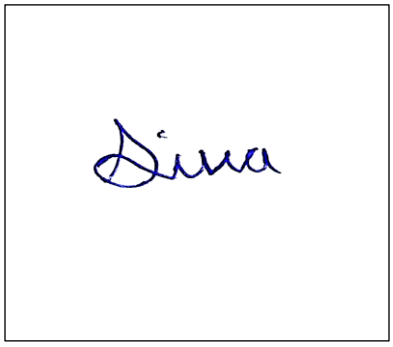

Fig 2.3.2 (b) signature image after noise

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

(ii) Conversion from Color image (RGB) into Gray level image:

There are several algorithms for converting a color image into a gray level image. The following four algorithms are found to be more common in used.

(a) Lightness method:

Here, the most prominent and least prominent colors are averaged. That is

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

I = Calculated gray level

R= Value of Red color plan

G= Value of Green color plan

B= Value of Blue color

(b) Average method:

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

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

(c) Luminosity method:

Human brain is more sensitive to green color than red and it is least sensitive to blue color. Accordingly in this method, different weights are given to these colors.

$$I=0.21R+0.72G + 0.07\text{B}$$

(d) 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 color image in to a gray level image.

$$I=0.2989R+0.587\ G + 0.114\ B$$

(iii) Cropping:

Scanned, signature image contains the signature and some white colored non-signature regions. Those superfluous non-image portions are removed by cropping the image to the bounding rectangle of the signature part.

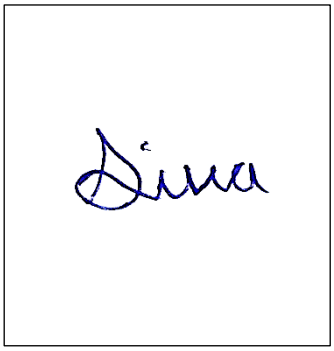

Fig 2.3.3 (a) A un cropped image

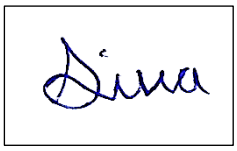

Fig 2.3.3 (b) A cropped image

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

(iv) 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.

v) Rotation for Skew Correction:

Many times, it is seen that during scanning of the signature images, the images are not properly oriented. This angular tilt in the signature image is called ‗skew'. Skew may result poor classification (depending on the classification technique used). Therefore there may be a need for skew correction of the signature images by rotating them. After skew correction, the final image is made parallel to the horizontal axis.

Following are the steps of skew correction,

1. Move the signature to origin by using the co-ordinates of centre of mass of the signatureimage.

2. Calculate the minimum Eigen Value of the matrix formed by using the new co-ordinates ofthe signature image.

3. Calculate the Skew angle using the Eigen vector.

$$\phi = \left[M(1) / M(2) \right]$$

M = the image matrix

M (1) = the first element in the first row of the image matrix

M (2) = the first element in the second row of the image matrix

After skew angle is found, skew correction is performed by applying rotation transformation to every pixel in the signature image.

Fig. 2.3.4 (a) Signature image with skew

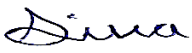

Fig. 2.3.4 (b) Signature image after skew correction

(vi) Slant Correction:

Slant is the tilt that an inclined signature makes with the vertical axis.Sometimes, the slant angle needs to be corrected before feature extraction.

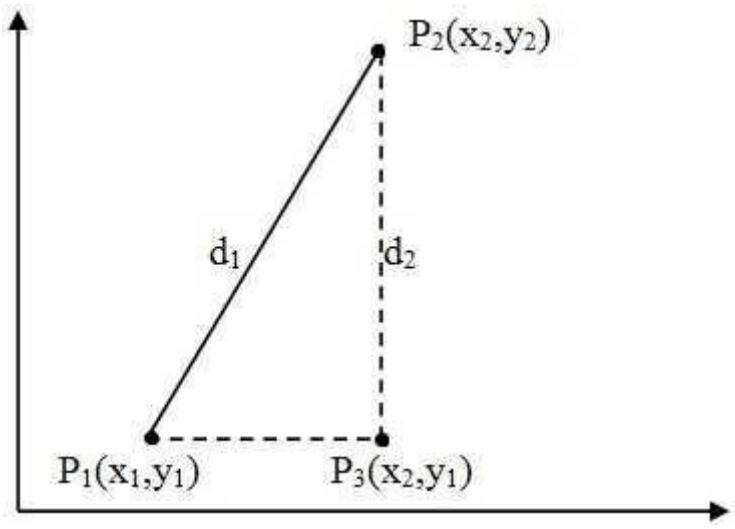

Fig. 2.3.5 Slant angle estimation

Following are the steps of a slant correction algorithm as discussed:

1.Obtain the left most pixel value as $P1 = (x1, y1)$.

2.Get the first maxima from the left most as ( ) $P2= x2 , y2$.

3.Set the value of ( ) $P = x2 ,y1$.

4.Calculate the distance between *P1* and *P2* (= d1 say).

5.Calculate the distance between *P2* and *P3* (= d2 say).

6.Find the slope 'm' between lines formed by joining *P1, P2* and *P2, P3*, If $m < 0$, then $k = -1$ else $k = 1$.

7.Slant angle is calculated by 1 ( ) 2 1 q = *k* sin- d / d

8.For every pixel in the signature image, slant correction is performed by applying the transformation:

$$( )1x= x - y \tan 90 -q \text{ and } 1\ y = y \qquad (5.3.4)$$

(vii) 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 becomesimportant. 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 position 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 neighbor approach. In this method, gray level is assigned accordingto the pixel which is the nearest to the mapped pixel.

FEATURE EXTRACTION

Feature extraction is a process of deriving some characteristic parameters or functionsfrom 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. 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.

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.

Types of Features in Offline Signatures:

Features extracted from an offline signature are basically classified into two categories.

(i) Local Features and (ii) Global Features (iii) Geometrical feature (iv) Statisticalfeatures (v) Pressure Features (vi) Shape 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 imageas 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. Two additional types of signature features are also found in literatures. They are (i)Geometrical features and (ii) Statistical features.

(iii) Geometrical Features:

Geometric features are derived from the geometrical parametersof 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.

(iv) 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.

(v) Pressure Features:

These represent variations in the pressure applied during the signing process, which can be detected through sensors embedded in digital pens or inferred from pen strokes in scanned images.

(vi) Shape Features:

These capture the geometric properties of the signature, such as contour information, Fourier descriptors, or chain code.

Features in Offline Signatures:

Global features produced better recognition results in offline signature verification. The following features from the signature samples present in our datasets:

(i) 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

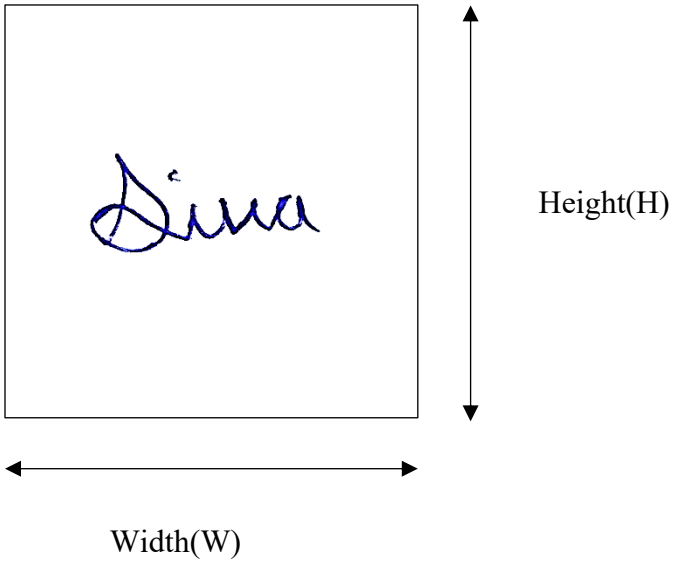

Fig 2.3.6 A signature image showing height and width

(a) 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 *b*y

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

The vertical center (Cy) 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)

(b) Horizontal and Vertical Projections (Horizontal and Vertical Histograms):

Horizontal projection or histogram is found by counting the number of signature image pixels in each row in a signature image and plotting it horizontally with a line.The row with maximum value gives the maximum horizontal projection. Similarly, counting and plotting signature pixels in vertical direction for every column, maximum vertical projection or histogram is found.

Horizontal and vertical projections can be calculated as follows:

$$P_v[x] = \sum_{y=1}^{m} \text{black pixel}(x,y)$$

$$P_h[y] = \sum_{x=1}^{n} \text{black pixel } (x,y)$$

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 aspects that 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

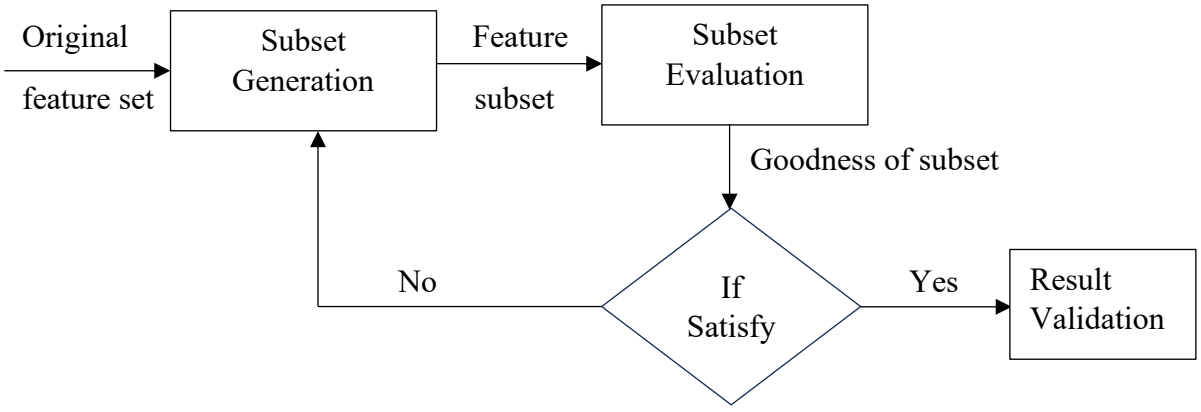

Fig. 2.3.7 Flow chart of Feature Selection Method

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 limitation of features does not contribute to significant increase in error.

Evaluation Process of Subset of Features:

There are three main methods for the evaluation of subset of features.

(i) Filter Methods (ii) Wrapper Methods (iii) Embedded Methods.

(i) Filter Methods:

This method was first proposed by John, G. H., Kohavi, R, &Pfleger. Filter method of feature selection uses different statistical tests (e.g. T-test, F-test, i-test, Euclidean distance, (C2) Chi-squared test, ANOVA, Information gain, Correlation coefficient score sets) 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.

Correlation-based Feature Selection (CFS):

Selects features that are highly correlated with the class labels but uncorrelated with each other.

Information Gain (IG) or Mutual Information:

Measures the amount of information provided by a feature regarding the class labels.

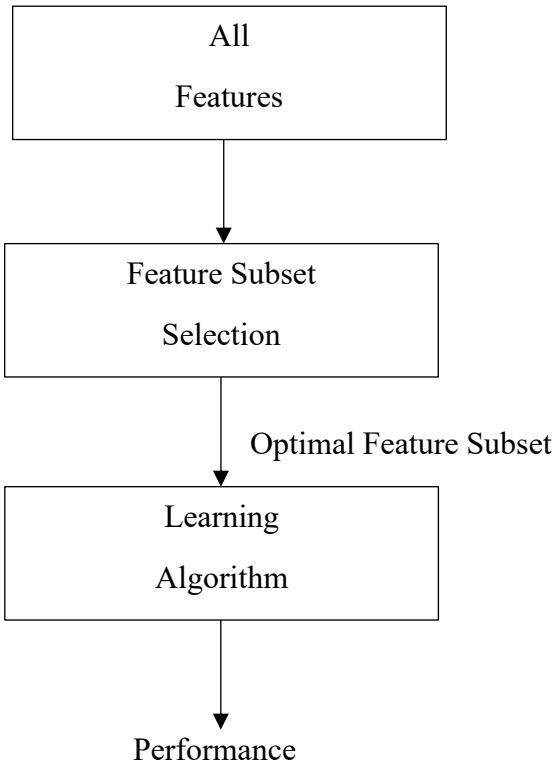

Fig 2.3.8 Operations in filter method

Advantages of Filter Methods:

Filter method is computationally easy and fast. There is no chance of over fitting 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.

(ii) Wrapper Methods:

This method was first presented by John, G. H., Kohavi, R, &Pfleger. Wrapper Method divides the feature set into some subsets using a search procedure.

I tapplies the classifier on them and based on the classification accuracy, feature subset is evaluated using cross validation. This method evaluates various possible combinations of feature subsets and classifiers (learning algorithm or induction algorithm) and finds the optimal combination that produces the best classification rate. Wrapper methods are sentially search algorithms that use the classifier as input the classification rate is the output to be optimized.

Forward Selection:

Starts with an empty set of features and iteratively adds the most relevant feature until a stopping criterion is met.

Backward Elimination:

Starts with the full set of features and iteratively removes the least relevant feature until a stopping criterion is met.

Recursive Feature Elimination (RFE):

Selects features by recursively removing the least important features based on the classifier's performance.

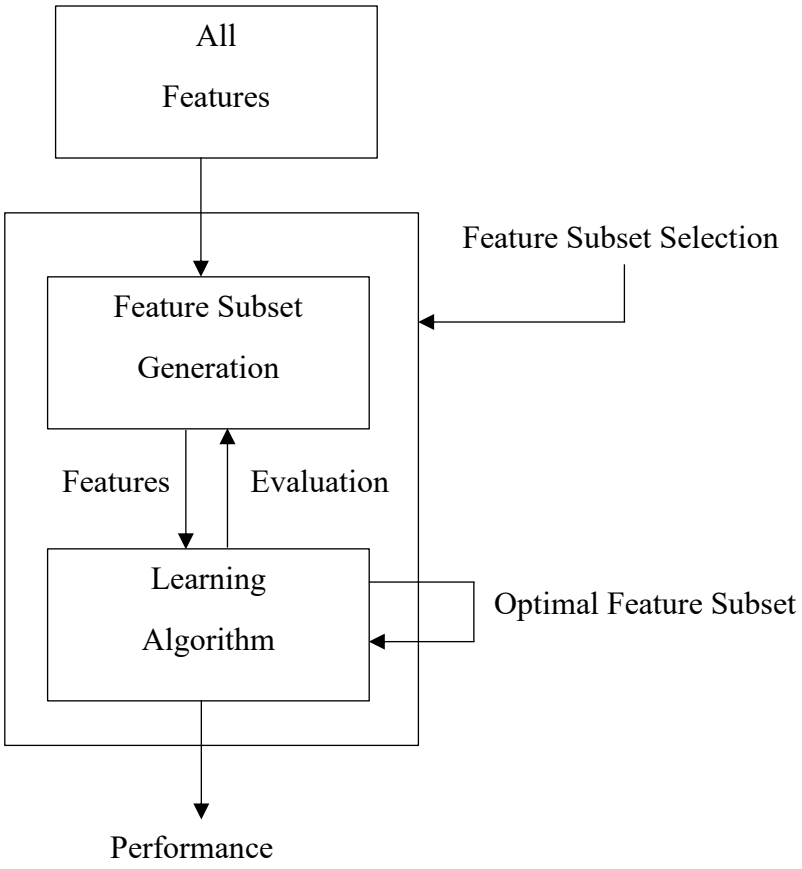

Fig 2.3.9 Operations in wrapper method

Advantages of Wrapper Methods:

Wrapper methods consider the inter dependencies of thedifferent features in a feature set. So, redundant features are eliminated. Therefore, they provide more accurate result as compared to filter methods

Drawbacks of Wrapper Methods:

There is a chance of over fitting in wrapper method. Wrapper methods are computationally costly.

(iii) Embedded Methods:

Embedded methods perform selection of features during the training of a specific classifier. The criteria for feature selection are found out during training. In this method, feature selection and training of the classifier are combined together. That means, selected features are linked with the concerned classifier.

Lasso (L1 regularization):

Encourages sparsity in the feature space by penalizing the absolute size of the feature coefficients.

Tree-based Methods:

Decision trees and ensemble methods like Random Forest and Gradient Boosting automatically select features based on their importance scores.

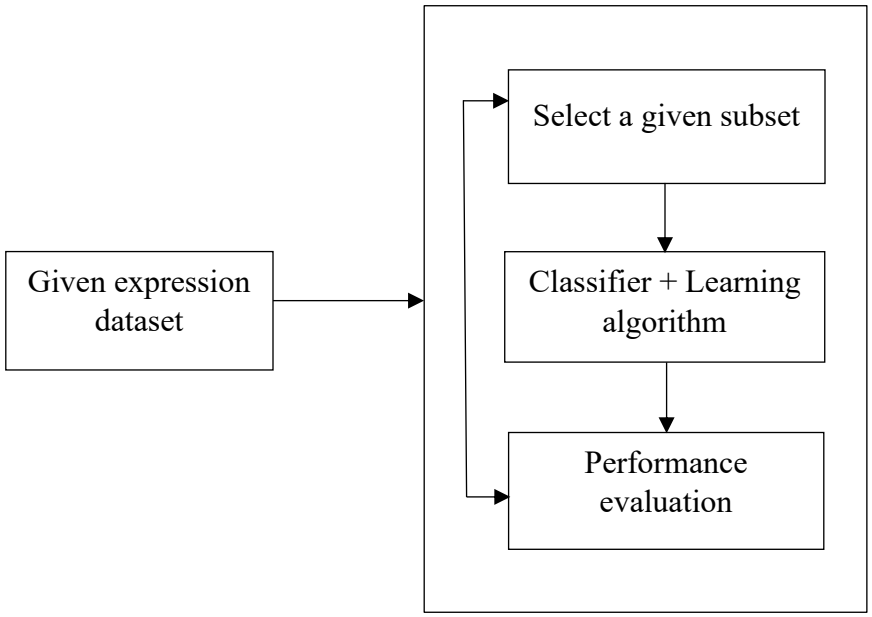

Fig 2.3.10 Operation in embedded method

Drawbacks of Embedded Methods:

These methods are classifier specific.

Advantages of Embedded Methods:

Like wrapper methods, embedded methods considerthe classifier while selecting the features; but it has less computational cost.

VERIFICATION USING EXTRACTED FEATURES

Features extracted are stored in a Vector. Minimum error threshold values is to be fixed for each feature implemented using which it classifies genuine signatures. Machine Learning Algorithm will minimize the error threshold. But since we have not been taught Machine Learning, we have done this step manually by data for input and selecting error threshold values. The more images this step is applied to the more accurate algorithm becomes.

Threshold technique:

Thresholding is a technique used in image processing and computer vision to convert grayscale or color image into binary images. In the context of comparing two images, thresholding can be used to determine whether the difference between corresponding feature extracted from the images is significant enough to consider them similar or dissimilar.

Before thresholding, features are extracted from both images. These features represent distinctive characteristics of the image and can include points, edges, textures or other relevant information.

This technique can define the given signature is genuine or forged.

Performance evaluation:

False rejection rate (FRR) or Type I error

$$FRR = \frac{\text{Total no. of genuine signatures classified as forged}}{\text{Total no. genuine signature samples}} \times 100\%$$

False acceptance rate (FAR) or Type II error

$$FAR = \frac{\text{Total no. of forged signatures classified as genuine}}{\text{Total no. forged signature samples}} \times 100\%$$

Recognition rate

$$\text{Recognition rate} = \frac{\text{Total no. of corectly classified signature samples}}{\text{Total no.of signature samples}} \times 100\%$$

CHAPTER-3

SYSTEM SPECIFICATIONS

3.1 SOFTWARE SPECIFICATIONS

- Language : MATLAB (MATRIX LABORATORY)
- Software : MATLAB R2021a
- Operating system : Windows 10

3.2 HARDWARE SPECIFICATIONS

- Processor : AMD Ryzon 5 Processor
- RAM : 4 GB
- Hard disk : 256 GB

3.3 LANGUAGE SPECIFICATIONS

MATLAB:

MATLAB (MATRIX LABORATORY) is a high-level language and interactive environment for numerical computation, visualization, and programming. Using MATLAB, you can analyze data, develop algorithms, and create models and applications.

The language, tools, and built-in math functions enable you to explore multiple approaches and reach a solution faster than with spreadsheets or traditional programming languages, such as C/C++ or Java.

MATLAB is used for a range of applications, including signal processing and communications, image and video processing, control systems, test and measurement, computational finance, and computational biology. More than a million engineers and scientists in industry and academia use MATLAB, the language of technical computing.

MATLAB (matrix laboratory) is a numerical computing environment and fourth-generation programming language, developed by Math Works. MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi domain simulation.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to- use environment where problems and solutions are expressed in familiar mathematical notation.

Justification for Utilizing MATLAB:

i) Image processing toolbox:

MATLAB provides comprehensive image processing toolbox with wide range of built-in functions and algorithms specifically designed for image processing tasks. It can provides extensive library files for image processing.

ii) Visualization:

MATLAB offers powerful visualization tools that enables user to visualize and analyze image data effectively.

iii) Integration with other toolbox:

MATLAB integrates with other toolbox for image processing such as signal processing, computer vision, machine learning and deep learning. This integration can helpful for complex image processing tasks.

iv) Community support:

MATLAB provides extensive documentation, tutorials and online forums, where users can seek, help, share knowledge and collaborate on image processing projects. It can provides references and details about the project.

v) Performance:

MATLAB offers high-performance computing capabilities with optimized algorithms and support parallel processing enabling efficient execution of image processing tasks.

Uses of MATLAB:

- Math and computation
- Algorithm development
- Modeling, simulation, and prototyping
- Data analysis, exploration, and visualization

Application development, including Graphical User Interface building MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take towrite a program in a scalar non interactive language such as C or FORTRAN.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACKprojects, which together represent the state-of-the-art in software for matrix computation.

MATLAB Software:

This is a high-level matrix/array language with control flow statements, Functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

MATLAB Working Environment:

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

Handle Graphics:

This is the MATLAB graphics system. It includes high-level commands for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level commands that allow you to fully customize the appearance of graphics as well as to build complete Graphical User Interfaces on your MATLAB applications.

MATLAB Mathematical Function Library:

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions likematrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms. It can also provides many library functions for image processing.

MATLAB API:

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It include facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files. MATLAB is an interpreted language for numerical computation. It allows one to perform numerical calculations, and visualize the results without the need for complicated and time consuming programming. MATLAB allows its users to accurately solve problems, produce graphics easily and produce code efficiently. Because MATLAB is an interpreted language, it can be slow, and poor programming practices can make it unacceptably slow.

CHAPTER-4

SYSTEM ANALYSIS

4.1 DATAFLOW DIAGRAM

Offline signature verification can be executed in these stages.

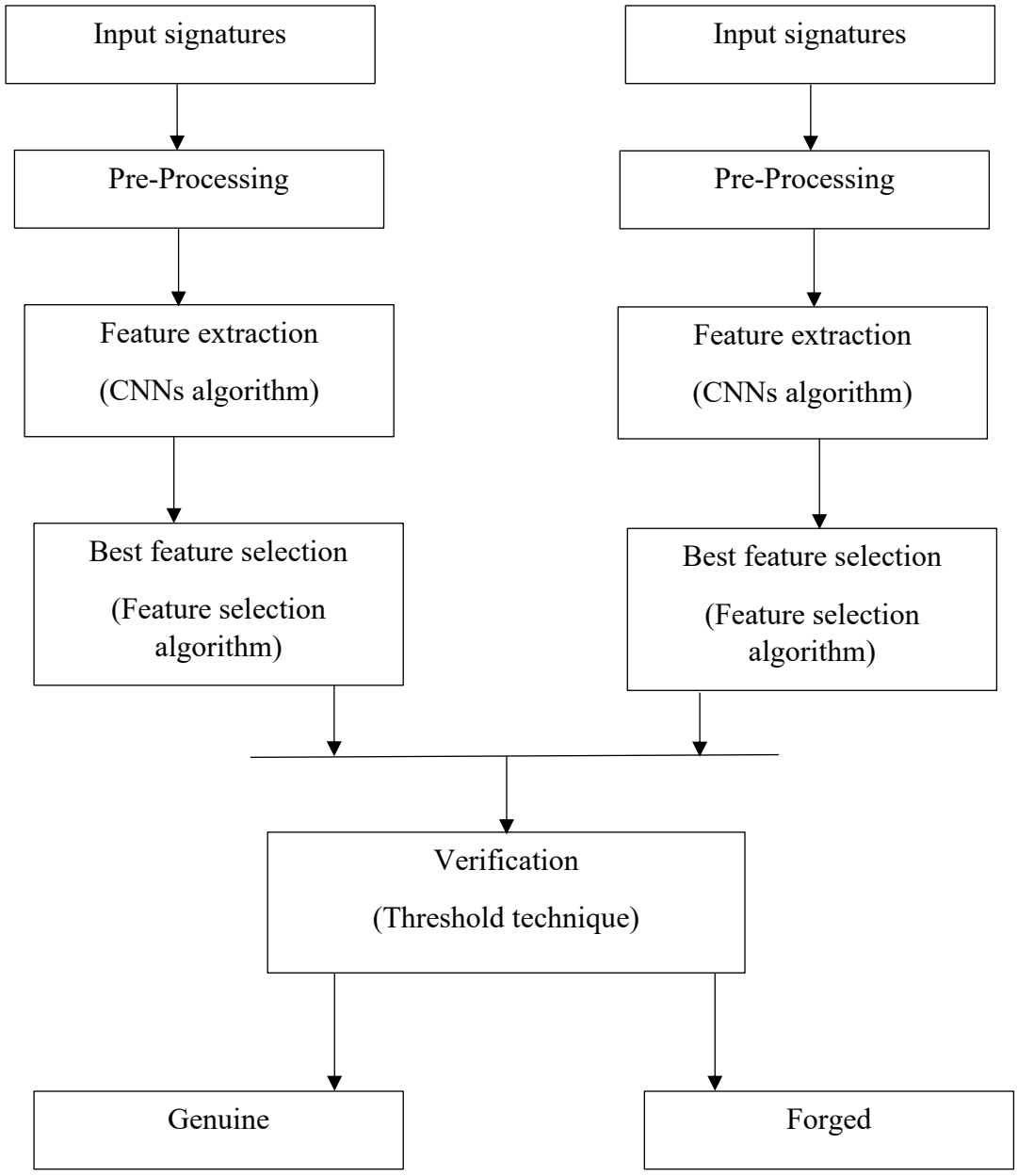

Fig 4.1 Dataflow diagram

CHAPTER-5

SYSTEM TESTING AND IMPLEMENTATION

5.1 SOFTWARE TESTING

Software testing is a systematic process of evaluating software applications or systems to ensure that they meet specified requirements and quality standards. It involves executing the software with the intent of finding defects or bugs and verifying that it functions correctly under various conditions.

Key aspects of software testing include, verification, validation. Identifying defects, preventing defects and improving quality.

Overall, software testing plays a crucial role in ensuring the reliability, quality, and success of software applications and systems in meeting the needs of users and stakeholders.

System testing:

System testing for an offline signature verification project would involve various stages to ensure its functionality, accuracy, and reliability.

Types of testing:

i) Unit testing:

Unit testing is a level of software testing where individual units/components ofsoftware are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived child class. In the offline signature verification each modules (units) are tested to work properly. After the unit testing, the modules of the signature verification are tested under integration testing.

ii) integration testing:

The objective is to take unit tested components and build a program structure that has been dictated by design. Integration testing is testing in which a group of components are combined to produce output. In this testing, the first unit of signature verification that have already been tested are combined into component and the interface between them are tested.

iii) Functional testing:

Verify that the system correctly captures signature images from input sources such as scanned documents or digital devices. Ensure that the signature preprocessing steps, such as noise removal, normalization, and segmentation, are performed accurately. Test the feature extraction process to ensure that relevant features are extracted from signature images effectively. Validate the signature matching or classification algorithms to accurately verify the authenticity of signatures.

iv) Non-Functional testing:

Evaluate the system's performance in terms of processing speed and memory usage, especially for large datasets of signature images. Test the system's robustness against variations in signature styles, sizes, orientations, and quality. Assess the system's usability by examining the user interface, input methods, and feedback mechanisms. Verify the system's security measures to prevent unauthorized access to sensitive signature data.

v) Block-box testing:

In this testing, software is tested such that it works fine for different inputs In this we just focus on required input and output without focusing on internal working.In this we have security A testing, recovery testing, stress testing and performance testing.

vi) White-box testing:

White-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT).

vii) System testing:

System testing ensures that the entire integrated software system meets requirements. It tested a configuration to ensure known and predictable results, an example of system testing is the configuration oriented system testing. System testing is based on process description and flows, emphasizing pre-defined process links and integration points.

viii) Regression testing:

Every time new module added leads to changes in program. This type of testing make sure that whole component works properly even after adding components to the complete program. Any changes that made in the offline signature verification may affect functionality of detection. This error are rectified in regression testing.

ix) Robustness testing:

Assess how the system handles variations in lighting conditions, image quality, and background noise. Test the system's resilience to common challenges such as smudging, overlapping signatures, or partial signatures.

5.2 SYSTEM IMPLEMENTATION

System implementation is the process of defining how the information system should be built (i.e., physical system design), ensuring that the information system is operational and used, ensuring that the information system meets quality standard (i.e., quality assurance).Offline signature verification system is implemented in the MATLAB software (version MATLAB R2021a) and the modules of signature offline verification systems are implemented for giving the functionality to genuine or forgery signature in an image.

We have implemented the CNNs algorithm and feature selection algorithm to classify the signature samples and generate the best feature. Then the threshold technique is used to convert the image in binary format and analyze the image. Many built in MATLAB function are used to create verification system. Images from the datasets are used to check the genuine signature.

CHAPTER-6

CONCLUSION & FUTURE ENHANCEMENT

6.1 CONCLUSION

In conclusion, the offline signature verification project aims to develop a robust and reliable system for authenticating handwritten signatures. Through the implementation of advanced deep learning techniques, such as convolutional neural networks (CNNs) and adversarial training, the proposed system offers significant improvements over traditional methods in terms of accuracy, robustness, and efficiency. The best feature were selected by feature selection algorithm such as filter method, wrapper method, embedded method. The final stage of verification has been done by threshold technique.

The project addresses the limitations of existing systems by leveraging deep learning models to automatically extract discriminative features from raw signature images, reducing dependency on manual feature engineering and enhancing adaptability to new signatures. Furthermore, the system's real-time verification capabilities and scalability make it suitable for a wide range of applications requiring prompt and secure authentication.

In India, forgery of signatures is a criminal offense punishable under various sections of the Indian Penal Code (IPC) and other relevant laws. The punishment for forgery depends on the severity of the offense and the specific circumstances involved. If anybody involved in the signature forgery crime they will punish under these IPC sections 420, 463, 464, 465, 467, 468, and 471.

These sections of the Indian Penal Code (IPC) outline the legal provisions and punishments for forgery of signatures and related offenses. Depending on the specific circumstances of the case, the court may impose imprisonment, fines, or both as penalties for forgery offenses.

6.2 FUTURE ENHANCEMENT

Future enhancements for an offline signature verification project includes several function and methodologies.

Multimodal Biometric Fusion:

Integrating multiple biometric modalities such as signatures, fingerprints, and iris scans can enhance authentication accuracy and security. Fusion techniques can combine the strengths of each modality while compensating for their individual limitations.

Feature extraction algorithm:

We have use some other algorithms to done feature extraction process. There are several algorithm is there for feature extraction in image processing such as Local Binary Patterns (LBP), Scale-Invariant Feature Transform (SIFT), speeded up Robust Features (SURF), Gabor Filters, Convolutional Neural Networks (CNNs), Principal Component Analysis (PCA).

Enhance dataset:

Here, We have Enhance the dataset more to train our algorithm. We can use multiple algorithm for feature extraction and feature selection function.

Dynamic Signature Analysis:

Incorporating dynamic features such as pen pressure, stroke speed, and trajectory into the verification process can provide additional layers of security and authenticity assessment. Analyzing the temporal aspects of signature execution can further improve the detection of forgeries.

Continuous Authentication:

Implementing continuous authentication mechanisms that monitor user behavior throughout a signing session can enhance security and mitigate risks of impersonation or unauthorized access. Real-time analysis of signature dynamics can dynamically adjust authentication thresholds based on user behavior.

Blockchain Integration:

Leveraging blockchain technology for signature verification can enhance security, transparency, and auditability. Storing signature verification logs on a decentralized ledger can provide tamper-proof records of authentication attempts and improve accountability.

Cross-Domain Verification:

Extending the verification system to support cross-domain authentication, such as verifying signatures across different documents or mediums (e.g., digital signatures on electronic documents versus physical documents), can enhance versatility and usability.

Explainable AI (XAI):

Integrating XAI techniques into the verification system can enhance transparency and interpretability of model decisions. Providing explanations for why a signature was classified as genuine or forged can improve user trust and facilitate forensic analysis in case of disputes.

CHAPTER-7

APPENDIX

7.1 SAMPLE CODE

Feature extraction

```
function [Feat_Val,im_processed] = featureExtraction(I)
%FEATUREEXTRACTION Summary of this function goes here
%   Detailed explanation goes here
%I=imread('image71.jpeg '); % Load the image file and store it as the variable I.
% figure(1),imshow(I);
% pause
I2=imresize(I,[512 ,512]);
% figure(2),imshow(I2);
% pause
I3=rgb2gray(I2);
%I3=medfilt2(I3]);
I3=im2double(I3);
I3=im2bw(I3);                    %converting image to black and white
I3 = bwmorph(~I3, 'thin', inf);                %thining the image
I3=~I3;
im1 = I3;
% figure(3),imshow(I3);
% pause
%extracting the black pixels
k=1;
for i=1:512
   for j=1:512
      if(I3(i,j)==0)
         u(k)=i;
         v(k)=j;
         k=k+1;
         I3(i,j)=1;
```

```
        end
    end
end
C=[u;v];
N=k-1;%the number of pixels in the signature
oub=sum(C(1,:))/N;   %the original x co-ordinate center of mass of the image
ovb=sum(C(2,:))/N;   %the original y co-ordinate center of mass of the image

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%********ROTATE******%%
%%%%%%%%%%%%%%%%%%%%%%%%
%moving the signature to the origin
for i=1:N
    u(i)=u(i)-oub+1;
    v(i)=v(i)-ovb+1;
end
% the new curve of the signature
C=[u;v];
ub=sum(C(1,:))/N;
vb=sum(C(2,:))/N;
ubSq=sum((C(1,:)-ub).^2)/N;
vbSq=sum((C(2,:)-vb).^2)/N;
for i=1:N
    uv(i)=u(i)*v(i);
end
uvb=sum(uv)/N;
M=[ubSq uvb;uvb vbSq];
%calculating minimum igen value of the matrix
minIgen=min(abs(eig(M)));
%the eigen vector
MI=[ubSq-minIgen uvb;uvb vbSq-minIgen];
theta=(atan((-MI(1))/MI(2))*180)/pi;
thetaRad=(theta*pi)/180;
rotMat=[cos(thetaRad) -sin(thetaRad);sin(thetaRad) cos(thetaRad)];
%% rotating the signature and passing the new co-ordinates
```

```
for i=1:N
   v(i)=(C(2,i)*cos(thetaRad))-(C(1,i)*sin(thetaRad));
   u(i)=(C(2,i)*sin(thetaRad))+(C(1,i)*cos(thetaRad));
end
C=[u;v];
%moving the signature to its original position
for i=1:N
   u(i)=round(u(i)+oub-1);
   v(i)=round(v(i)+ovb-1);
end
%after rotating the image the signature might go out of the boundry (128x128) therefore
%we have to move the signature curve
mx=0;%the moving x co-ordinate
my=0;%the moving y co-ordinate
if (min(u)<0)
   mx=-min(u);
   for i=1:N
      u(i)=u(i)+mx+1;
   end
end
if (min(v)<0)
   my=-min(v);
   for i=1:N
      v(i)=v(i)+my+1;
   end
end
C=[u;v];
for i=1:N
   I3((u(i)),(v(i)))=0;
end
% figure(4),imshow(I3);
% pause
% removing extra white space in sides
xstart=512;
```

```
xend=1;
ystart=512;
yend=1;
for r=1:512
   for c=1:512
      if((I3(r,c)==0))
         if (r<ystart)
            ystart=r;
         end
         if((r>yend))
            yend=r;
         end
         if (c<xstart)
            xstart=c;
         end
         if (c>xend)
            xend=c;
         end
     end
   end
```

GUI

```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input arguments.
%
```

```
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help GUI
% Last Modified by GUIDE v2.5 26-Nov-2018 23:59:47
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

7.2 SCREENSHOTS

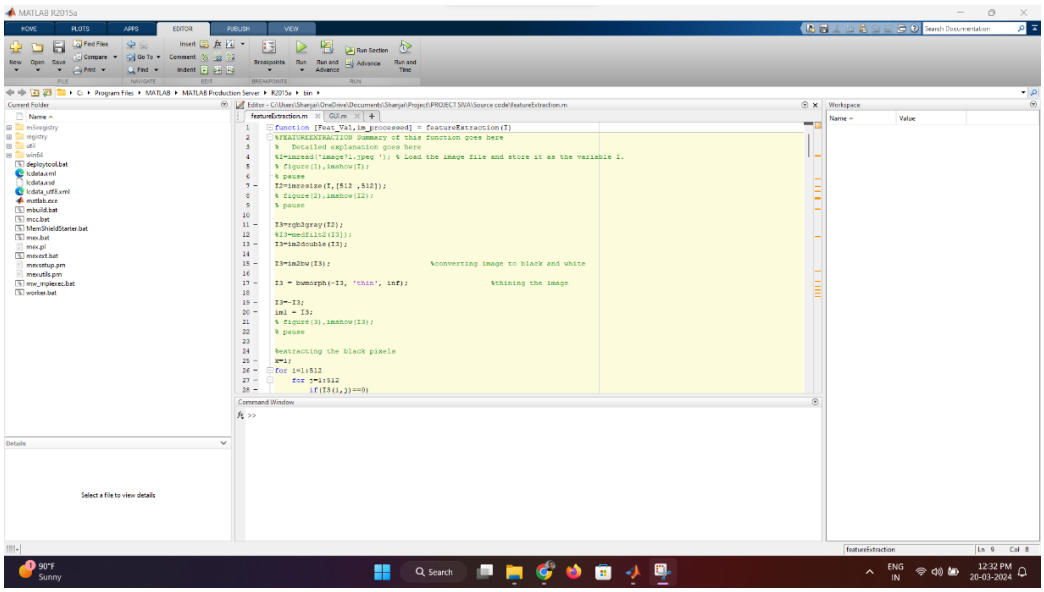

Fig 7.2.1 Source code screen

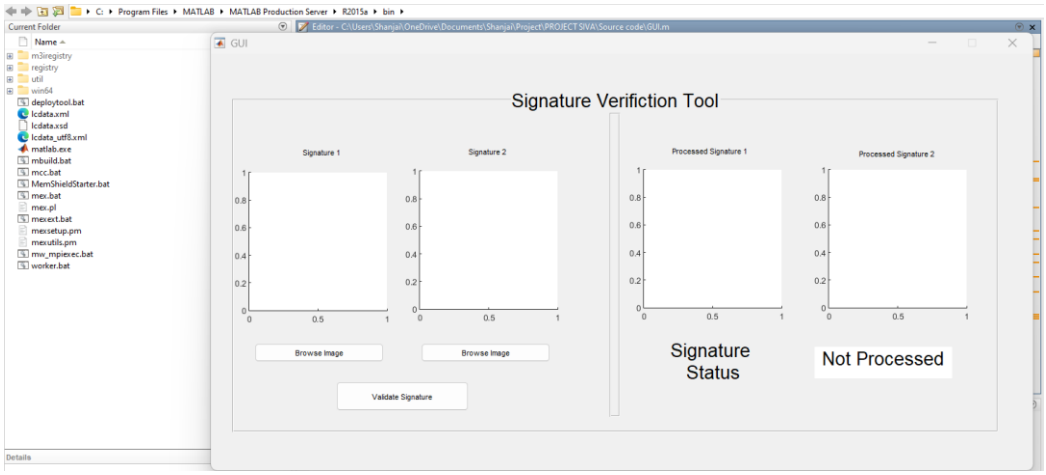

Fig 7.2.2 Open GUI screen

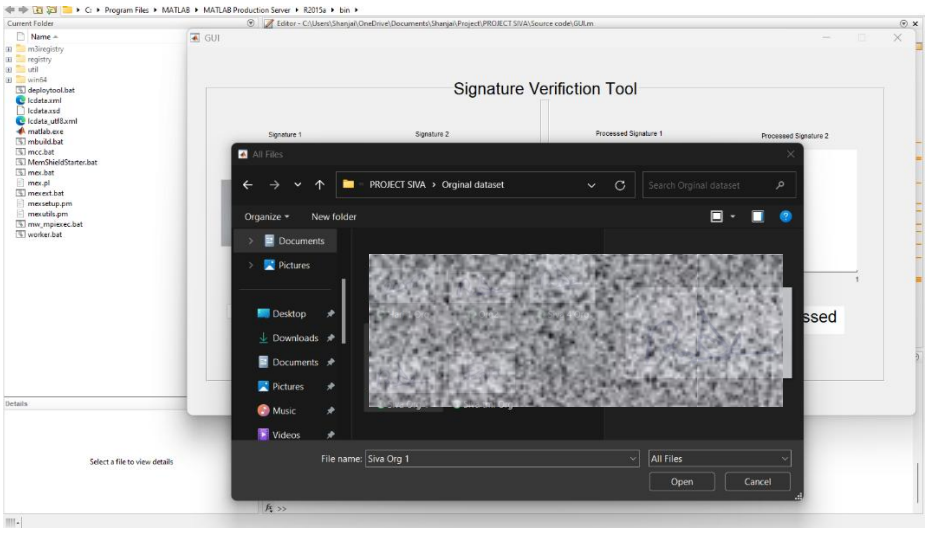

Fig 7.2.3 Input the signature from the dataset

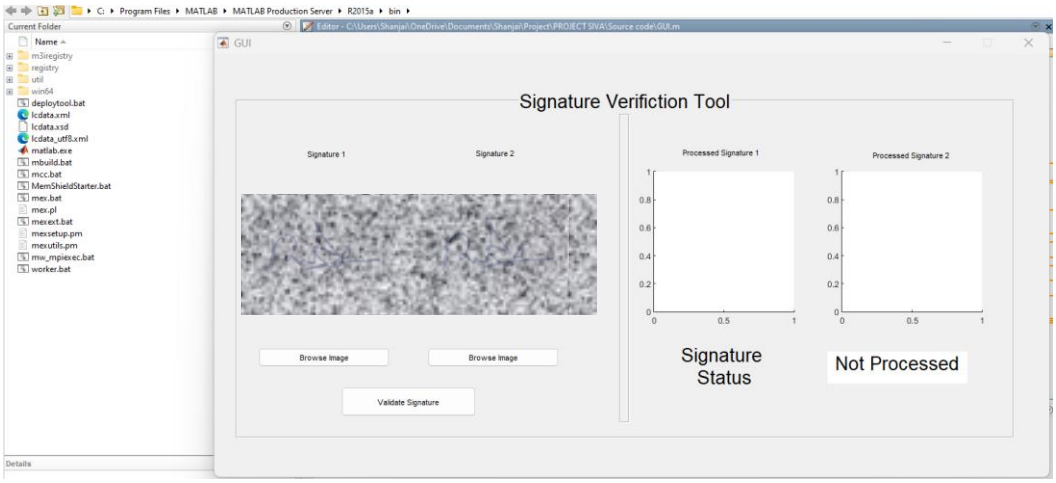

Fig 7.2.4 Uploaded both signatures

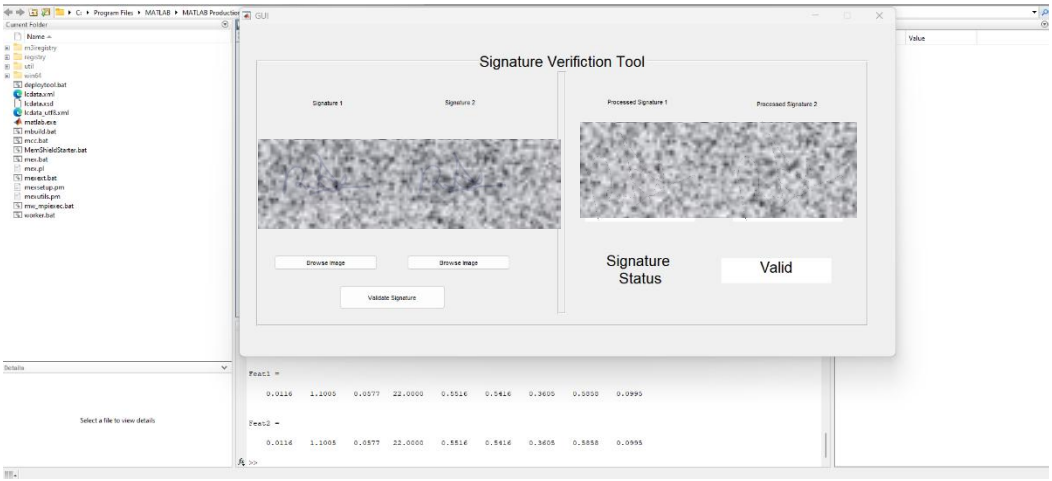

Fig 7.2.5 Same signature valid

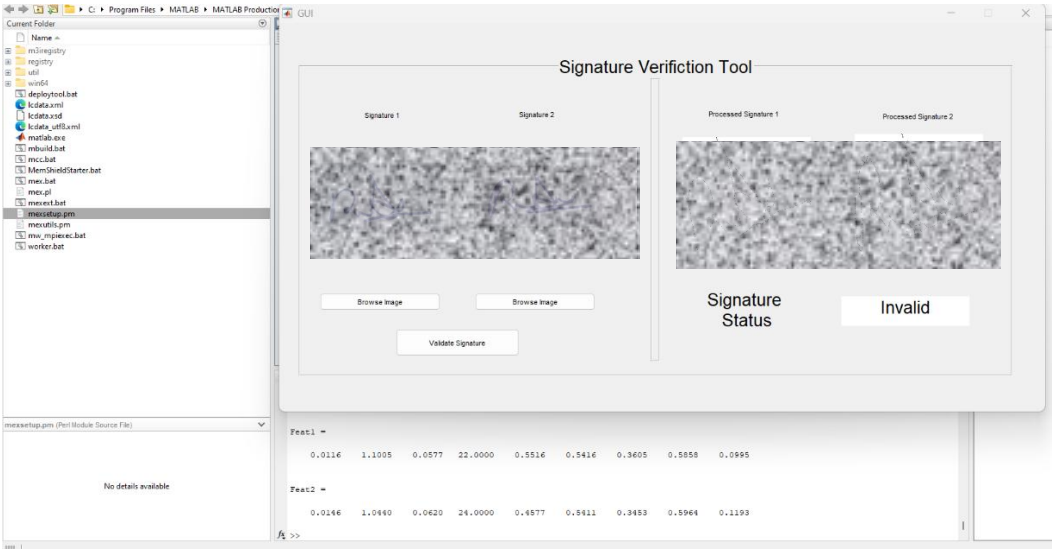

Fig 7.2.6 Different signature invalid

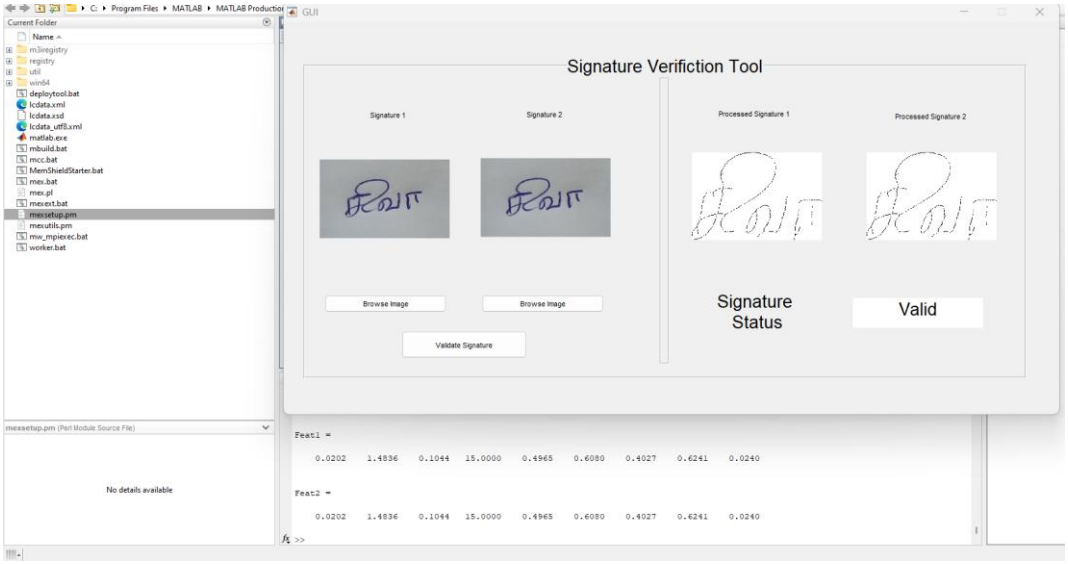

Fig 7.2.7 Other language same signature valid

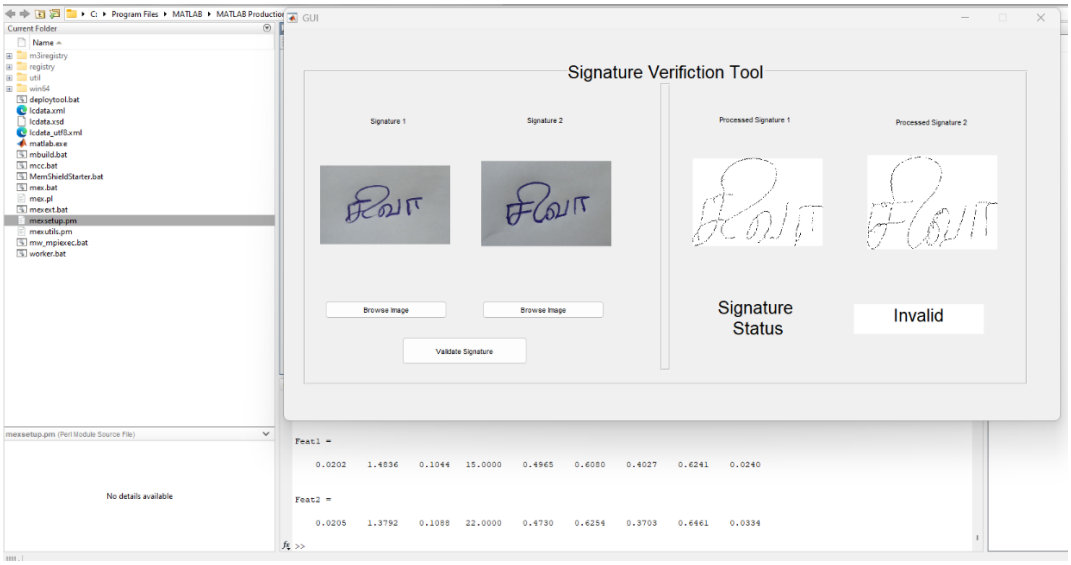

Fig 7.2.8 Other language different signature invalid

CHAPTER-8

BIBLIOGRAPHY

REFERENCE BOOKS

[1] “Biometrics: Concepts, Methodologies, Tools, and Applications" edited by Jason P. Poovey (2020) - This book provides insights into various biometric technologies, including signature verification, along with methodologies and applications.

[2] “Deep Learning for Biometrics" edited by Bir Bhanu and Nalini K. Ratha (2019) - This book explores the application of deep learning techniques in various biometric modalities, including signature verification.

[3] “Convolutional Neural Networks for Visual Recognition“ edited by Fei-Fei Li, Justin Johnson, Serena Yeung (2020) - This book provides a comprehensive introduction to Convolutional Neural Networks (CNNs) for visual recognition tasks. It covers topics such as image classification, object detection, and feature extraction using CNNs.

[4] “Deep Learning for Computer Vision” edited by Rajalingappaa Shanmugamani (2019) - This book provides a comprehensive overview of deep learning methods for computer vision tasks, with a focus on CNNs. It covers CNN architectures, feature extraction techniques, and applications in image recognition and classification.

[5] “Deep Learning for Vision Systems“ edited by Mohamed Elgendy (2020) - This book explores deep learning techniques for vision systems, including CNNs. It covers topics such as image preprocessing, feature extraction, and transfer learning using CNNs.

[6] “Digital Image Processing Using MATLAB” edited by Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins (2019) - This practical guide focuses on digital image processing techniques implemented using MATLAB, including thresholding methods. It provides step-by-step explanations and MATLAB code examples for implementing various thresholding algorithms.

[7] “Digital Image Processing“ edited by Rafael C. Gonzalez, Richard E. Woods (2017) - This comprehensive textbook covers various aspects of digital image processing, including thresholding techniques. It provides a thorough explanation of thresholding methods and their applications in image segmentation.

[8] “Forensic Handwriting Identification: Fundamental Concepts and Principles“ edited by Ron Morris (2019) - This book provides a comprehensive introduction to forensic handwriting identification, including techniques for offline signature verification. It covers fundamental concepts, handwriting comparison methodologies, and case studies.

REFERENCE WEBSITES

[1] https://www.mathworks.in

[2] https://ieeexplore.ieee.org

[3] https://iopscience.iop.org

[4] https://indianexpress.com

[5] https://www.researchgate.net

[6] https://chat.openai.com

[7] https://m.economictimes.com/wealth/personal-finance-news/icici-bank-branch-manager-forged-signatures-siphoned-off-rs-11-95-crore-by-transfer-and-cash-case-in-eows-net/articleshow/108075852.cms

LIST OF ABBREVIATIONS

TERM	EXPANSION
MATLAB	MATRIX LABORATORY
RAM	RANDOM ACCESS MEMORY
LBP	LOCAL BINARY PATTERNS
BFS	BREADTH FIRST SEARCH
CNNs	CONVOLUTIONAL NEURAL NETWORKS
JPEG	JOINT PHOTOGRAPHIC EXPERTS GROUP
RGB	RED GREEN BLUE
NTSC	NATIONAL TELEVISION SYSTEM COMMITTEE
CFS	CORRELATIONS BASED FEATURE SELECTION
RFE	RECURSIVE FEATURE ELIMINATION
FRR	FALSE REJECTION RATE
FAR	FALSE ACCEPTANCE RATE
ICT	IN CIRCUIT TESTING
IPC	INDIAN PENAL CODE