

UNLOCKING SECURITY: ENHANCED SIGNATURE VERIFICATION SYSTEM USING IMAGE PROCESSING

A PROJECT REPORT

Submitted by

KARTHIKEYAN. P (310120104040)

MURALIDHARAN. U (310120104055)

*in partial fulfillment for the award of the degree
of*

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



ANAND INSTITUTE OF HIGHER TECHNOLOGY, CHENNAI

ANNA UNIVERSITY:: CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY:: CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report “**UNLOCKING SECURITY:ENHANCED SIGNATURE VERIFICATION SYSTEM USING IMAGE PROCESSING**” is the bonafide work of “**KARTHIKEYAN.P (310120104040) , MURALIDHARAN.P (310120104055)**” who carried out the project work under my supervision.

SIGNATURE

Dr.S.Roselin Mary,Ph.D.

HEAD OF THE DEPARTMENT PROFESSOR

Department of Computer Science
and Engineering
Anand Institute of Higher
Technology
Kazhipattur
Chennai-603103

SIGNATURE

Mrs.R.Pratheeba, M.E.,

SUPERVISOR ASSISTANT PROFESSOR

Department of Computer Science
and Engineering
Anand Institute of Higher
Technology
Kazhipattur
Chennai-603103

Submitted to Project Viva Voce Examination held on.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost, we thank the **Almighty**, for showering his abundant blessings on us to successfully complete the project. Our sincere thanks to our Honorable Founder “**Kalvivallal**” **Late Thiru.T. Kalasalingam, B.Com.**

Our sincere thanks and gratitude to our “**SevaRatna**”, **Dr. Sridharan K. Com, M.B.A., M.Phil., Ph.D.**, Chairman, **Dr. S. Arivazhagi, M.B.B.S.**, Secretary for giving us the support during the project work. We convey our thanks to **Dr.K.Karnavel,Ph.D.**, principal Incharge for the support towards the successful completion of this project.

We wish to thank **Head of the Department Dr. S. Roselin Mary, Ph.D.**, our **Project Coordinator Mr.A.S.Balaji, M.E(Ph.D).**, **Assistant Professor** and our **Project Guide Mrs.R.Pratheeba, M.E., Assistant Professor** for the co- ordination and better guidance and constant encouragement in completing in this project.

We also thank all the **Staff members** of the Department of Computer Science and Engineering for their commendable support and encouragement to go ahead with the project in reaching perfection.

Last but not the least our sincere thanks to all our parents and friends for their continuous support and encouragement in the successful completion of our project.

ABSTRACT

This signature verification system utilizing image processing techniques, with a focus on the Structural Similarity Index (SSIM) algorithm. The system aims to authenticate signatures by comparing a reference signature with a query signature. Through the integration of image processing methodologies, the system extracts relevant features from the signatures to facilitate comparison and authentication. The SSIM algorithm serves as a key component in quantifying the structural similarity between signatures, enabling robust and accurate verification. The project encompasses the design, development, and testing phases, culminating in the creation of a functional signature verification system capable of providing reliable authentication results. Through experimentation and evaluation, the effectiveness and performance of the system are assessed, demonstrating its potential applications in security-sensitive environments requiring dependable signature authentication mechanisms. The Structural Similarity Index (SSIM) algorithm assesses the likeness between two images by comparing their structural information, luminance, and contrast. It dissects images into patches, computing local statistics such as mean, variance, and covariance for each. These statistics facilitate the computation of individual SSIM values for patch pairs, which are then aggregated to produce an overall SSIM value for the entire image. Higher SSIM values indicate greater perceptual similarity between the images. SSIM's ability to capture structural details makes it suitable for tasks like image quality assessment.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF TABLES	vii
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1.	INTRODUCTION	1
	1.1 OBJECTIVE	5
	1.2 SCOPE	5
2.	LITERATURE REVIEW	6
3.	ANALYSIS	11
	3.1 SYSTEM ANALYSIS	11
	3.1.1 Problem Definition	11
	3.1.2 Existing System	11
	3.1.3 Proposed System	12
	3.2 REQUIREMENTS ANALYSIS	13
	3.2.1 Functional Requirements	13
	3.2.2 Non-Functional Requirements	13
	3.2.3 Hardware Specification	14
	3.2.4 Software Specification	14
4.	SYSTEM DESIGN	15
	4.1 OVERALL PROCESS DESIGN	15
	4.2 UML DIAGRAM	16
	4.1.1 Use Case Diagram	16
	4.1.2 Class Diagram	17

4.1.3	Sequence Diagram	18
4.1.4	State Diagram	19
4.1.5	Collaboration Diagram	19
4.1.6	Activity Diagram	20
5.	IMPLEMENTATION	21
5.1	MODULES	21
5.1.1	Login Module	21
5.1.2	Signup Module	21
5.1.3	Home Module	21
5.1.4	Verification Module	21
5.1.5	Result Module	22
6.	SYSTEM TESTING	23
6.1	Testing and Validation	23
6.2	Different Stages of Testing	24
6.3	Build the Test Plan	25
7.	RESULT AND DISCUSSION	28
8.	USER MANUAL	29
9.	CONCLUSION	30
10.	FUTURE ENHANCEMENT	31
	APPENDICES	
	APPENDIX I : BASE PAPER	
	APPENDIX II : SCREEN SHOTS	
	APPENDIX III : PUBLICATIONS	
	REFERENCES	

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
6.1	Test Cases	25
6.2	Test Case log	27

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	SSIM Algorithm Structure	2
4.1	Overall Architecture of Entire System	15
4.2	Use Case Diagram of Entire System	16
4.3	Class Diagram of Entire System	17
4.4	Sequence Diagram of Entire System	18
4.5	State Diagram of Entire System	19
4.6	Collaboration Diagram of Entire System	19
4.7	Activity Diagram of Entire System	20
7.1	Line Graph for the sample result of the two image	28

LIST OF ABBREVIATIONS

SYMBOL	ABBREVIATION
UML	Unified Modeling Language
DB	Data Base
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JS	JavaScript
OOPs	Object Oriented Programming
CMD	Command Prompt
SSIM	Structural Similarity Index

CHAPTER 1

INTRODUCTION

In today's digital age, the verification of signatures holds immense importance across numerous sectors, including finance, legal affairs, and administrative processes. Signatures serve as a fundamental method of authentication, validating the identity of individuals and confirming their consent or approval on various documents. However, traditional methods of manual signature verification are labor-intensive, time-consuming, and susceptible to errors. As a result, there is a growing need for automated systems that can accurately and efficiently verify signatures, leveraging the capabilities of image processing technology. Manual signature verification typically involves visual inspection by trained professionals, who compare a submitted signature with reference samples to determine its authenticity. However, this process is subject to several limitations:

Subjectivity: Human judgment can vary, leading to inconsistencies in the evaluation of signatures.

Time-Consuming: Verifying signatures manually can be a time-consuming task, particularly when dealing with large volumes of documents.

Expertise Required: Skilled personnel are needed to accurately assess the validity of signatures, adding to the operational costs.

The Role of Image Processing

Image processing techniques offer a promising solution to the challenges associated with manual signature verification. By analyzing digital representations of signatures, these techniques can extract and quantify various visual features, enabling automated comparison and authentication processes. Key advantages of using image processing for signature verification include

Objective Analysis: Image processing algorithms can objectively analyze signature images based on predefined criteria.

Efficiency: Automated signature verification systems can improve operational efficiency and reduce processing times.

Accuracy: By leveraging advanced image analysis techniques, such as pattern recognition, these systems can achieve high levels of accuracy in signature authentication.

Components of Signature Verification Using Image Processing

Image Acquisition: Signature images are captured using digital scanners or cameras, ensuring high-quality input for the verification process.

Preprocessing: The captured images undergo preprocessing techniques, such as noise removal, binarization and normalization, to enhance their quality and suitability for analysis.

Feature Extraction and Representation: Relevant features are extracted from the preprocessed signature images, including stroke patterns, edge profiles, curvature, and texture and Extracted features are transformed into a suitable representation format, such as feature vectors or histograms, for further analysis.

Comparison Algorithm: A comparison algorithm is applied to measure the similarity between the features of the input signature and reference samples.

ALGORITHM USED:

The Algorithm used in this system is **SSIM**, SSIM is a perceptual image quality assessment algorithm that measures the similarity between two images. Three factors in SSIM: luminance, contrast, and structural information. It compares the mean and variance of the luminance and contrast of the two images, as well as the correlation.

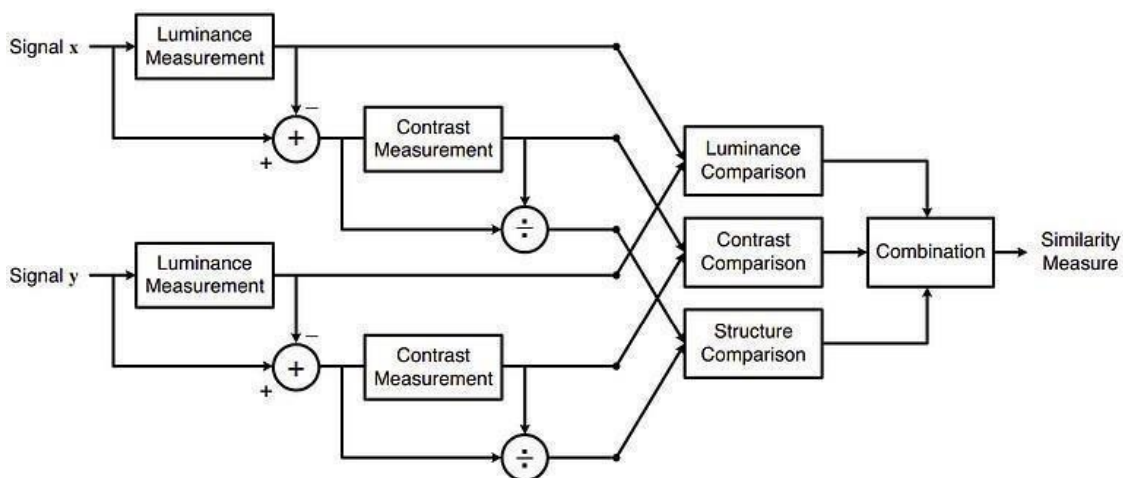


Fig 1.1 SSIM Algorithm Structure

The SSIM algorithm produces a similarity score between 0 and 1, where 1 indicates that the two images are identical and 0 indicates that they are completely different. This system calculates the Structural Similarity Index between 2 given images which is a value between -1 and +1. A value of +1 indicates that the 2 given images are very similar or the same while a value of -1 indicates the 2 given images are very different.

- **Luminance:** Luminance is measured by averaging over all the pixel values. Its denoted by μ (Mu) and the formula is given below,

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i.$$

The Luminance Comparison Function is,

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

- **Contrast:** It is measured by taking the standard deviation (square root of variance) of all the pixel values. It is denoted by σ (sigma) and represented by the formula below,

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}}$$

The Contrast Comparison Function is,

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

- **Structure:** The structural comparison is done by using a consolidated formula (more on that later) but in essence, we divide the input signal with its standard deviation so that the result has unit standard deviation.

$$(\mathbf{x} - \mu_x) / \sigma_x$$

The Structure Comparison Function is,

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3}.$$

And finally, the SSIM score is given by,

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma$$

where $\alpha > 0$, $\beta > 0$, $\gamma > 0$ denote the relative importance of each of the metrics.

To simplify the expression, if we assume, $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$, we can get,

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}.$$

1.1 OBJECTIVE

The primary objective of this project is to design and implement an automated system capable of verifying signatures by analyzing their visual characteristics.

- **Feature Extraction:** To extract relevant features from signature images, such as stroke patterns, curvature, texture, and spatial relationships between key points.
- **Comparison and Matching:** Design methodologies for comparing and matching the extracted features of an input signature with those of a reference signature and matching by using SSIM Algorithm.
- **Implementation:** Implement the signature verification system using image processing libraries and frameworks.
- **Evaluation:** Evaluate the performance of the developed system through extensive testing on genuine and forged signatures. Measure key performance metrics such as accuracy, precision, recall.

1.2 SCOPE

The scope of this project encompasses several key aspects of signature verification using image processing techniques:

- **Document Authentication:** Implement the system to verify signatures on important documents such as contracts, legal agreements, and financial transactions to ensure their authenticity.
- **Banking and Finance:** Integrate the system into banking applications for verifying signatures on checks, loan documents, and other financial instruments to prevent fraud and unauthorized transactions.
- **Forensic Analysis:** Apply the system in forensic investigations and criminal cases for verifying signatures on evidence, crime scene documents, and witness statements to support legal proceedings.
- **Education:** Implement the system in educational institutions for verifying signatures on exam papers, certificates, and academic transcripts to prevent cheating and ensure academic integrity.

CHAPTER 2

LITERATURE SURVEY

Title : New Textural Features Handwritten Signature Image Verification

Authors : Suriya Soisang, Suvit Poomrittigul

Publication : IEEE Conference Proceedings

Concept Discussed:

The paper proposes a new textural feature, Local Binary Gradient Quantization Angle Patterns (LBGQAP), for offline handwritten signature verification. The concept integrates Local Binary Patterns (LBP) with Gradient Quantization Angle (GQA) to enhance the precision of signature verification.

Work Done:

The authors develop the LBGQAP method by combining LBP with GQA and utilize an Artificial Neural Network (ANN) classifier for signature verification.

Problem Identified:

The need for improved offline handwritten signature verification methods due to the prevalence of non-genuine signatures in various applications such as bank transactions and official documents.

Knowledge Gained:

It demonstrating the effectiveness of LBGQAP compared to traditional methods. The study highlights the importance of feature extraction techniques in enhancing the accuracy of signature verification systems.

Gaps:

While the proposed LBGQAP method shows promising results, further research could investigate its performance across diverse datasets and evaluate its robustness against various types of forgeries and exploring the scalability and real-world applicability of the method could provide valuable insights for its implementation in practical settings.

Title : Signature Recognition Using Image Processing

Authors : Mirudu Basini K S, Gopinath R

Publication : IJARIIIE-ISSN(O)-2395-4396

Concept Discussed:

The development of a Scale Invariant Feature Transform (SIFT) based Automatic Signature Recognition (ASR) system for identifying particularly for traffic control purposes.

Work Done:

The method for efficient license plate localization in images with complex backgrounds, utilizing techniques such as image enhancement, edge extraction, and morphological filtering. It also discusses character segmentation and recognition using OCR methods.

Problem Identified:

The challenge of identifying document owners in traffic situations where traditional methods may be impractical due to factors like high vehicle speed and non-uniform signature details.

Knowledge Gained:

The application of image processing techniques for signature recognition and character analysis, along with discussing the challenges and approaches in this domain.

Gap:

While the paper presents a method for signature recognition, it does not extensively discuss the evaluation of the proposed system's performance or compare it with existing methods in the literature.

Title: Signature Verification Using Image Processing Techniques

Author: Amina Khatra

Publication: GRA - GLOBAL RESEARCH ANALYSIS

Concept Discussed:

Offline signature verification using image processing techniques

Work Done:

Proposing a method for offline verification of signatures using simple shape-based geometric features, preprocessing of scanned images, feature extraction, and comparison with a template signature.

Problem Identified:

Limited availability of signature data for robust parameter estimation, difficulty in developing a general system to classify various signature styles.

Knowledge Gained:

Importance of offline signature verification, challenges in feature extraction, classification, and achieving high accuracy across different signature styles.

Gap:

Performance deterioration in detecting skilled forgeries, potential improvements using higher-dimensional feature spaces and dynamic information from the signing process.

Title: Offline Signature Verification Using Image Processing

Authors: Bushra Shaik, Jyothi Manohar Katikireddy, Vamsidhar Kambham, K Sravani

Publication: E3S Web of Conferences 391, 01074 (2023)

Concept Discussed:

Offline signature verification using Convolutional Neural Networks (CNN), image processing techniques, deep learning, and recurrent neural networks (RNNs).

Work Done:

The authors proposed a method for offline signature verification using CNNs, focusing on feature extraction and classification. They trained various CNN architectures such as VGG16, Inception, ResNet50, and Xception using different optimizers to achieve high accuracy in signature recognition.

Problem Identified:

The challenge of accurately verifying handwritten signatures, which are prone to variations due to factors like age, behavior, and environment. Traditional methods of signature verification may not be efficient in detecting forgeries, leading to potential fraud.

Knowledge Gained:

Understanding of how CNNs and other deep learning techniques can be applied to offline signature verification, the importance of feature extraction, model training, and evaluation for achieving accurate results.

Gap:

Despite advancements in deep learning techniques, the authors noted that separating authentic signatures from expert forgeries remains a challenging task. They also highlighted the need for further research to improve the accuracy and efficiency of offline signature verification systems.

Title: Signature Verification using Image Processing & Neural Network

Author: Sayali Gowre

Publication: International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Volume 2, Issue 2, April 2022

Concept Discussed:

Offline signature verification using image processing techniques and neural networks.

Work Done:

To developing an offline signature verification system by extracting shape-based geometry features from digitized signature images. Various image processing techniques such as grayscale conversion, median filtering, intensity adjustment, thresholding, and canny edge detection were employed to preprocess the signature images. The processed images were then fed into a neural network for classification.

Problem Identified:

The challenges in offline signature verification include the absence of stable dynamic characteristics, difficulty in segmenting signature strokes due to unconventional writing styles.

Knowledge Gained:

The study explored techniques for preprocessing signature images, feature extraction, and classification using neural networks.

Gap:

Additionally, exploring the integration of online and offline verification methods could enhance the overall performance of signature verification systems.

CHAPTER 3

ANALYSIS

3.1 SYSTEM ANALYSIS

3.1.1 PROBLEM DEFINITION

The problem addressed in this project is the need for an efficient and reliable method of signature verification. Traditional manual methods of signature verification are prone to errors and are time-consuming, particularly in scenarios requiring rapid authentication, such as financial transactions or access control. Additionally, manual verification methods lack scalability and consistency, making them unsuitable for large-scale applications. Therefore, there is a demand for automated systems capable of accurately verifying signatures with minimal human intervention.

3.1.2 EXISTING SYSTEM

Existing signature verification systems predominantly rely on manual inspection or basic rule-based algorithms. Manual verification methods involve visual inspection of signatures by trained personnel, which can be subjective and time-consuming. Rule-based algorithms often use simple metrics such as stroke thickness or geometric features for verification, but these methods may lack robustness and are susceptible to forgeries.

Disadvantages of existing system:

- **Variability:** Signatures can differ significantly each time they are written, leading to inconsistencies in verification accuracy.
- **Forgery on Training data:** With only a dataset for training. When a forgery adds their signature in the dataset, the system will accurately identify signatures, especially those from individuals not included in the training set.
- **Privacy Concerns:** Collecting and storing handwritten signature data raises privacy concerns, particularly regarding the security of sensitive personal information.

- **Legal Uncertainty:** The legal validity of electronically verified signatures may not be universally recognized, leading to potential challenges in legal contexts.

3.1.3 PROPOSED SYSTEM

The proposed system aims to address the limitations of existing methods by leveraging image processing techniques for signature verification. By analyzing the visual characteristics of signatures, such as stroke patterns, curvature, and texture, the system will automate the verification process. Key components of the proposed system include feature extraction algorithms, comparison methodologies, and decision-making criteria for authentication. The system will offer a scalable, efficient, and reliable solution for signature verification across various applications.

Advantages of proposed system:

- **Versatility:** It can authenticate the handwritten signature with powerful image processing technique.
- **Efficiency:** Utilizes advanced image processing techniques for quick and accurate signature verification, enhancing the efficiency of the authentication process.
- **Balanced Performance:** Employs a lightweight processes that balances computational efficiency with powerful verification capabilities, ensuring both speed and accuracy.
- **Enhanced Security:** Helps enhance document security by reducing the risk of unauthorized access or forgery, safeguarding sensitive information effectively.

3.2 REQUIREMENT ANALYSIS

3.2.1 FUNCTIONAL REQUIREMENT

- **Image Input:** The system should be capable of accepting input in the form of scanned signature images.
- **Preprocessing:** Preprocessing techniques such as noise reduction and normalization should be implemented to enhance image quality.
- **Feature Extraction:** Algorithms for extracting relevant features from signature images, such as stroke patterns and texture, should be developed.
- **Comparison Methodologies:** The system should employ methods for comparing extracted features and determining the similarity between signatures.
- **Decision Making:** Criteria for making decisions regarding the authenticity of signatures based on comparison results should be established.
- **User Interface:** A user-friendly interface should be designed to facilitate input of signature images and display verification results.

3.2.2 NON-FUNCTIONAL REQUIREMENT

- **Accuracy:** The system should achieve a high level of accuracy in verifying signatures, minimizing false acceptance and rejection rates.
- **Robustness:** The system should be robust to variations in signature style, writing conditions, and image quality.
- **Speed:** The verification process should be efficient, providing timely responses to authentication requests.
- **User Experience:** The user interface should be intuitive and easy to navigate, ensuring a positive user experience.
- **Security:** Measures should be implemented to protect sensitive signature data and prevent unauthorized access to the system.

3.2.3 HARDWARE SPECIFICATION

- Hard-Disk: 20GB and above
- RAM: 4GB or Above
- Processor: P IV and Above
- Input Device: Camera
- Display Device: Monitor

3.2.4 SOFTWARE SPECIFICATION

- Operating System: Windows 10 or Higher
- Code Editing Tools: VS Code
- Programming Language:
 - Front-End: HTML, CSS, JS
 - Back-End: Python
 - Packages Needed: Numpy, SciKit, Tkinter, Flask, OpenCV, OS, Matplotlib, Sqlite3
 - Database: Sqlite3

CHAPTER 4

SYSTEM DESIGN

4.1 OVERALL ARCHITECTURE

An architectural diagram is used to abstract the overall outline of the software system. It involves a user registering and logging in, with their details stored in a database. The user can upload a signature image, which is captured using a camera or select from the device the image is pre-processed then performs histogram and curvature and edge detection, followed by feature extraction. The extracted features are compared using the SSIM (Structural Similarity Index) algorithm. The result of the verification is then displayed.

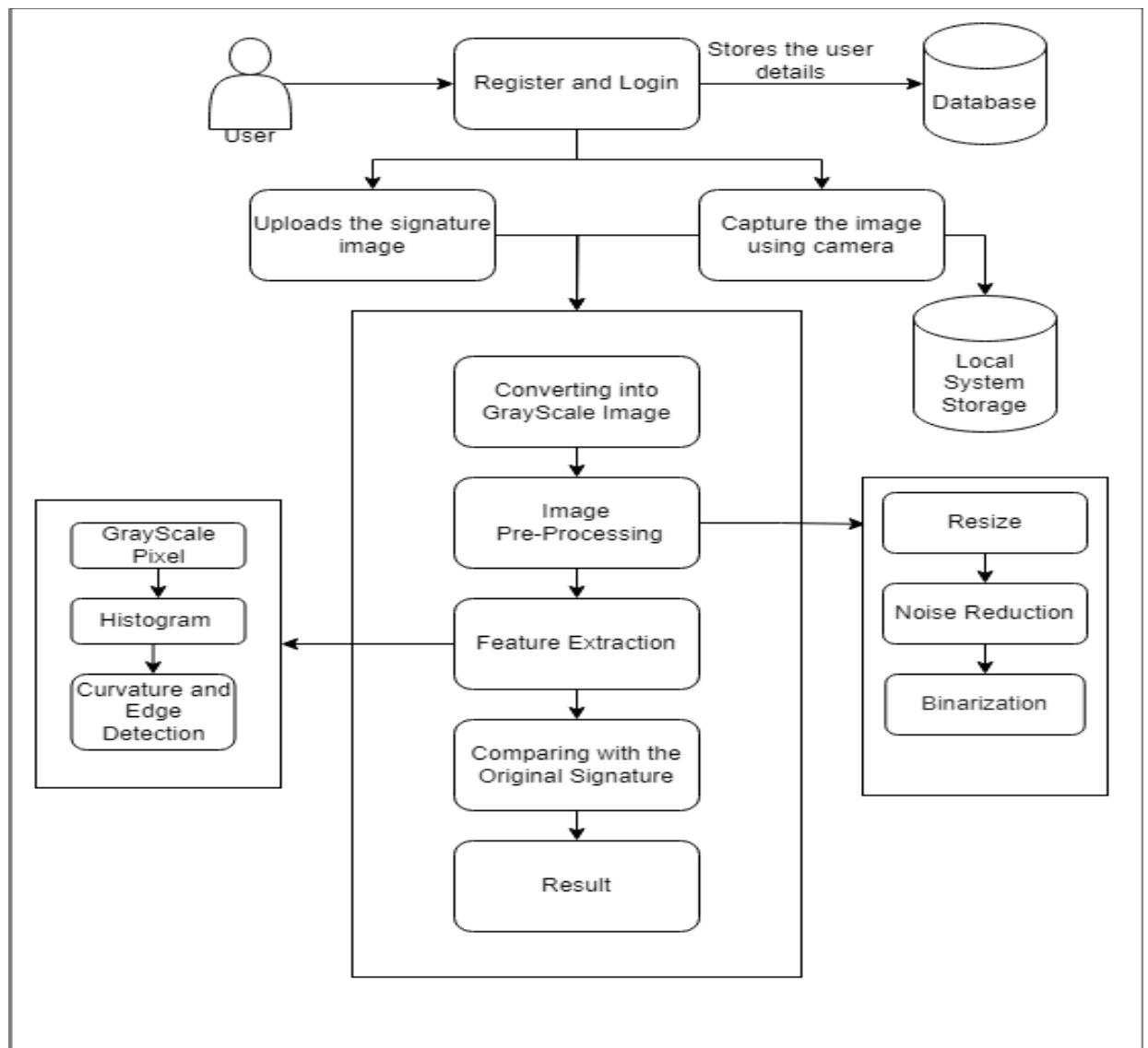


Fig 4.1 Architecture Diagram for Entire System

4.2 UML DIAGRAM

Unified Modelling Language (UML) is simply another graphical representation of a common semantic model. The proposed system has been designed by using use case diagram, class diagram, sequence diagram, collaboration diagram, state chart diagram and component diagram.

4.2.1 USE CASE DIAGRAM

This diagram illustrates the interactions between users and the system, outlining various actions users can perform and how the system responds to those actions. For this project, it might include actions like uploading signature images, selecting action types, Clicking the buttons and viewing results.

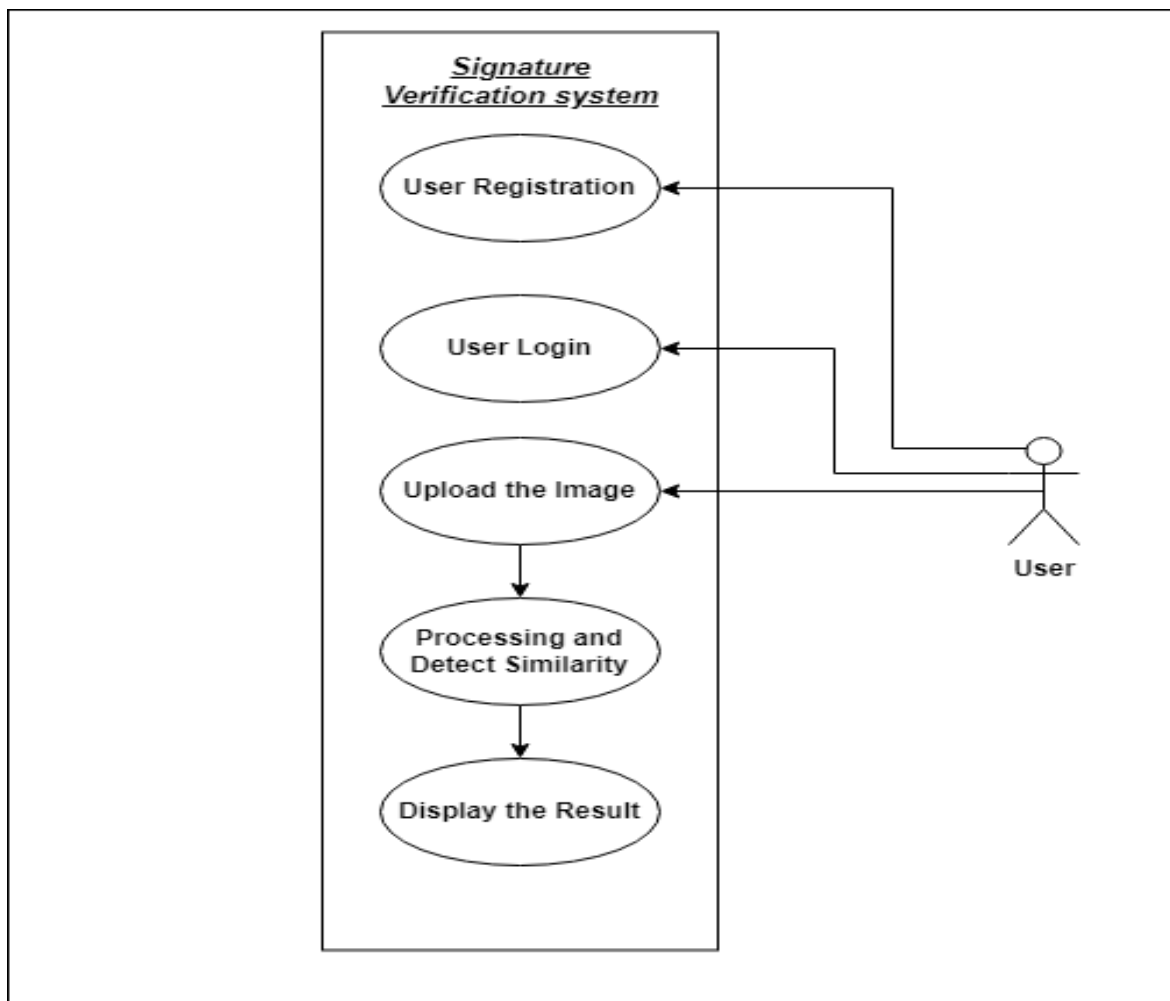


Fig 4.2 Use-Case Diagram for entire system

4.2.2 CLASS DIAGRAM

Class diagram is to model the static view of an application. Class diagrams represent the structure of the system by showing the classes, attributes, methods, and relationships between them. For this project, it might include classes like Register, Login, Image Pre-Processing, Feature Extraction and Recognize and displaying the result.

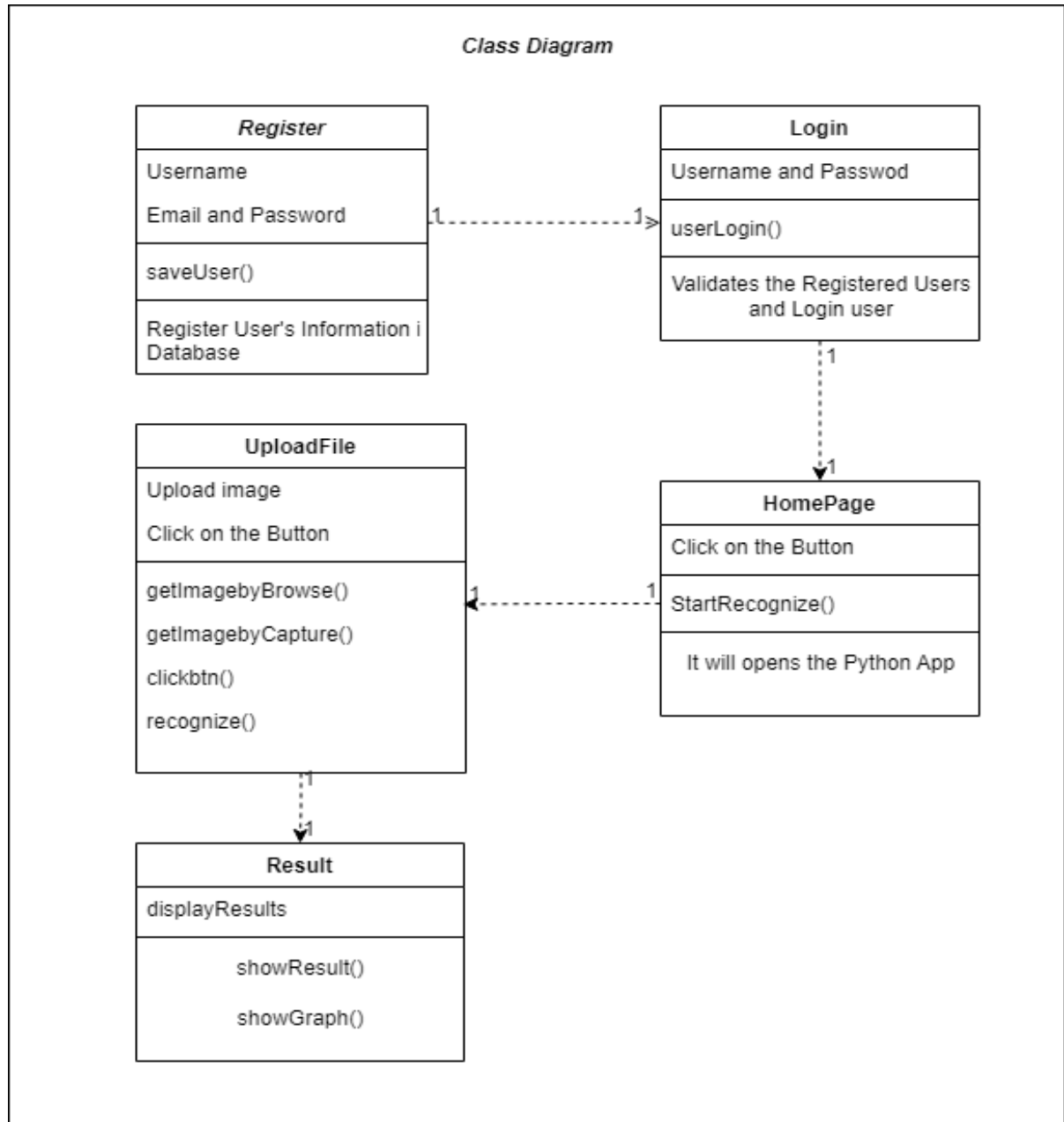


Fig 4.3 Class Diagram for Entire system

4.2.3 SEQUENCE DIAGRAM

It shows the control flow between various participants or entity roles of the corresponding system in the form of messages. Sequence diagrams illustrate the interactions between objects in a sequential manner, showing the order of messages exchanged between them. It could demonstrate how a fundus image is processed, segmented, and classified using various components in your system.

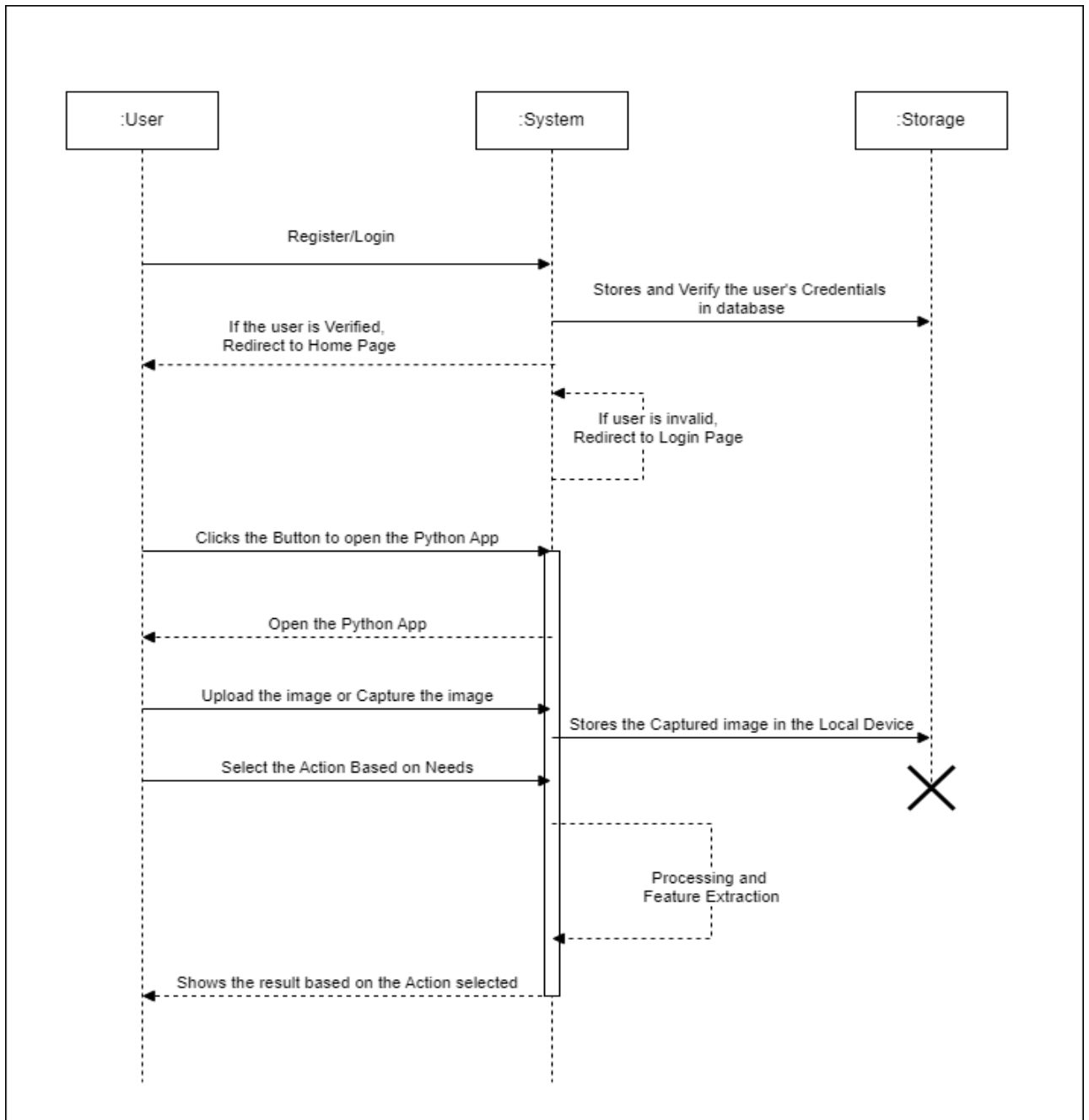


Fig 4.4 Sequence Diagram for Entire System

4.2.4 STATE DIAGRAM

This diagram models the different states a system can be in and how it transitions between them in response to events. In this project, it could depict states such as idle, processing image, feature Extraction and displaying results.

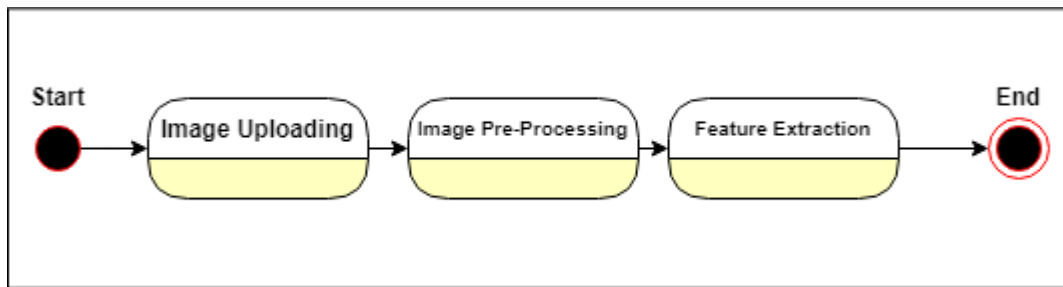


Fig 4.5 State Diagram for Entire System

4.2.5 COLLABORATION DIAGRAM

Collaboration diagram is defined as one of the interaction diagrams. The collaboration diagram is also called as the set of message exchange among the objects within the collaborative nature of message exchange between the corresponding.

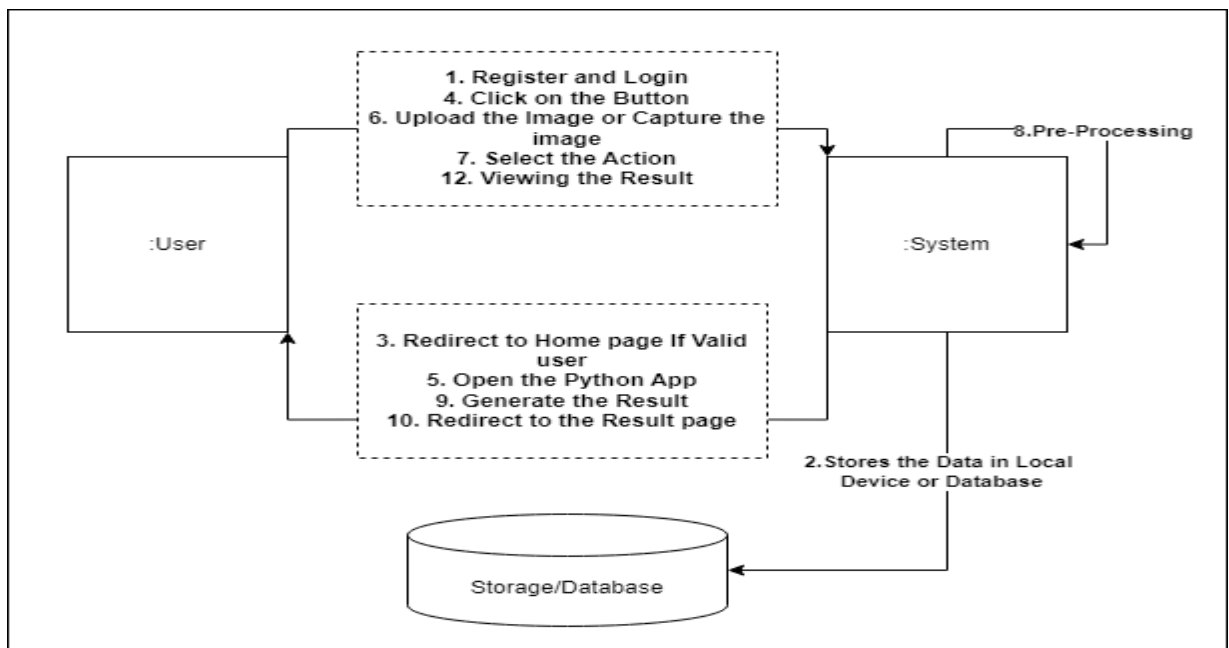


Fig 4.6 Collaboration diagram for Entire System

4.2.6 ACTIVITY DIAGRAM

Activity diagrams depict the flow of activities within the system, showing how different processes or actions are interconnected. In this case, it could illustrate the steps involved in Registration and login process, preprocessing of image, segmenting them, extracting features, and justify the similarity and flow of various page navigation.

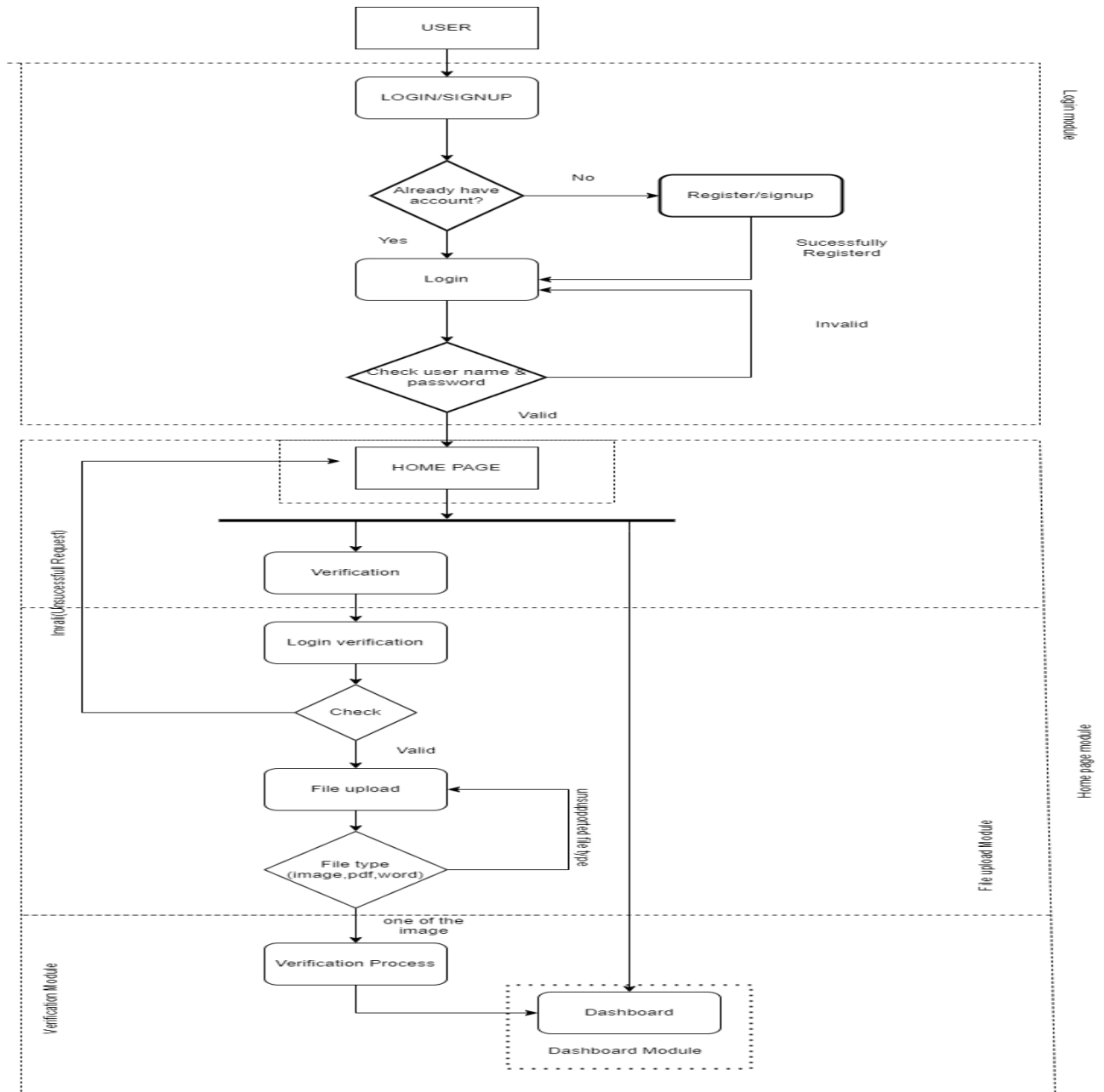


Fig 4.7 Activity Diagram for Entire System

CHAPTER 5

IMPLEMENTATION

5.1 MODULES

5.1.1 LOGIN MODULE

- **Login Form:** The login form serves as the entry point for users to access the system. It typically includes fields for entering the username and password.
- The login module is a crucial component of the enhanced signature verification system.

5.1.2 SIGNUP MODULE

- **Registration:** Users need to register their credentials to create an account within the system. This information is typically stored securely in a database.
- **Password Security:** Passwords should be stored securely using cryptographic hashing algorithms.

5.1.3 HOME MODULE

- The homepage module serves as the initial landing page of the enhanced signature verification system.
- It provides users with an overview of the system's capabilities, access to relevant functionalities, and navigation options.

5.1.4 VERIFICATION MODULE

- Provide a user-friendly Python app where users can upload their signature images and a file input field allowing users to select files from their local device or capture the image using camera
- **Verification:** Verification of uploaded file by using image processing technique and feature extraction process and verify that the image.

5.1.5 RESULT MODULE

- The dashboard module in the enhanced signature verification system provides users with an interactive and informative interface to monitor and manage their activities within the system.
- Display a list or grid of the user's previous signature uploads, along with relevant details such as upload date, file name, verification status, and any associated notes or comments. Users can review their upload history and track the progress of verification requests.

CHAPTER 6

SYSTEM TESTING

6.1 TESTING AND VALIDATION

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment. Validation Testing - Workflow: Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.

ACTIVITIES:

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Test

TESTING LEVELS

Functional Testing

Functional testing is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing, and it is not concerned about the source code. Every functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual result with the expected results.

Example of Functional Testing Types:

Unit Testing

User Acceptance Testing

Integration Testing

Regression Testing

Non-Functional Testing

Non-functional testing is a type of testing to check non-functional aspects of a software application. It is explicitly designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. A good example of non-functional test would be to check how many people can simultaneously login into a software. Nonfunctional testing is equally important as functional testing and affects client satisfaction.

Example of Non-Functional Testing Types:

Performance Testing

Scalability

Usability Testing

Load Testing

6.2 DIFFERENT STAGES OF TESTING

UNIT TESTING

Unit testing is a level of software testing where individual units/ components of a software 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. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In OOPs, a unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Unit Testing Benefits

Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, it will be able to promptly catch any defects introduced due to the change. The unintended impact of changes to any code is less. Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

Integration Testing

Integration testing for signature verification involves testing the interactions between different components or modules of the signature verification system to ensure they work together seamlessly. Test cases are designed to verify the integration points where these components interact, such as input/output interfaces and data flows. The goal is to identify any issues related to data exchange, communication protocols, or compatibility between components. Integration testing ensures that the system functions correctly as a whole, validating its ability to process signatures accurately across all integrated modules.

System Testing

System testing for signature verification encompasses testing the entire system as a cohesive unit, focusing on its functionality, performance, and reliability. Test cases are designed to evaluate the system's behavior against its requirements and specifications. This involves testing various aspects such as signature validation accuracy, response time, scalability, and error handling under different conditions. System testing verifies that the signature verification system meets user expectations and operates correctly within its intended environment. It also includes testing non-functional requirements like security and usability.

Regression Testing

Regression testing for signature verification involves verifying the functionality of a signature verification system after changes or updates. It begins with identifying test cases covering various signature scenarios. Baseline testing establishes the system's initial performance. After implementing changes, a regression test suite is created and executed to compare results against the baseline. Any discrepancies are debugged and fixed, with documentation of issues and resolutions. Automating the process can streamline future testing, ensuring the system maintains its accuracy and reliability over time.

6.3 BUILD THE TEST PLAN

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identify the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

TEST CASES

Table 6.1 Test cases design

S NO	Test case ID	Test Description	Test procedure	Test Input	Expected Result	Actual Result
01	S101	To check whether the python is running or not	Open Cmd in your system	Type Python3 in the Cmd	It should opens the python in the Cmd.	Opens python as expected
02	S102	To check whether the Flask server is running or not	Open IDE (VS code) and Run the Flask	Open Ide and Run the python code App.py	open the server http://127.0.0.1:5000	Server running as expected
03	S103	To Sign up and Login using the credentials.	User have to Sign up and Login into the page.	Login Credentials	It should redirect to HomePage	Redirect to Home page as expected
04	S104	To run the Python App by clicks the trigger in the Homepage	The User have to Click the Trigger.	Clicks the trigger to start the python app	It opens the Python App.	Open the Python App as expected
05	S105	Upload the image and Select the Action	User should submit the image in the App	Selects the input and Choose the Option	It should shows the Result on the Result Page	It shows result in Result Page as expected

TEST LOGS

Table 6.2 Test log

S. No	Test ID	Test Description	Test Status (PASS/FAIL)
01	S101	To check whether the python is running or not	PASS
02	S102	To check whether the Flask server is running or not	PASS
03	S103	To Sign up using username, email and Password and Login using the credentials.	PASS
04	S104	To run the Python App by clicks the trigger in the Homepage	PASS
05	S105	Upload the image and Select the Action and Get the Result on the Result Page.	PASS

CHAPTER 7

RESULT AND DISCUSSION

The signature verification system utilizing image processing compares two signature images by computing a similarity score. If the similarity score surpasses a predetermined threshold, the system categorizes the signatures as genuine; otherwise, they are deemed forged. Upon receiving input, the system proceeds with comparing the extracted features of the signatures and calculates their resemblance using a suitable similarity measure, such as Euclidean distance and using the Structural Similarity Index (SSIM) Algorithm. The obtained similarity score is then contrasted with the predefined threshold to make the authentication decision.

The system's performance is extensively elaborated. This includes detailing the similarity score computation process, highlighting any preprocessing steps involved in standardizing the input images, and specifying the similarity measure employed. Accuracy metrics are presented, such as true positive rate, false positive rate, precision, and recall, to provide a comprehensive evaluation of the system's effectiveness. Additionally, visual representations, like ROC curves can further elucidate the system's performance.

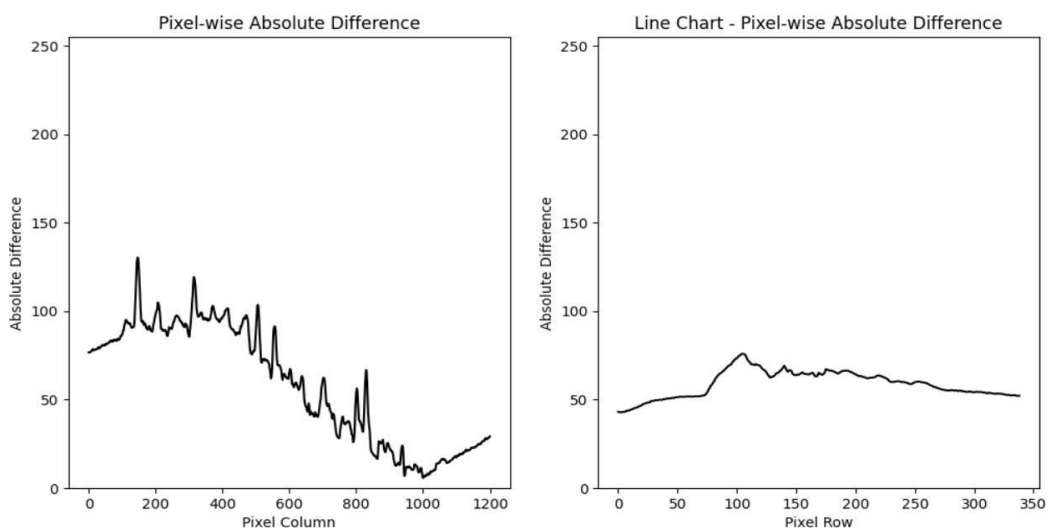


Fig 7.1 Line Graph for the sample result of the two image

CHAPTER 8

USER MANUAL

Installation of Python in Windows

Step 1: A Python 3.xx setup pop-up window will appear.

Step 2: Ensure that the Install for all user radio button is pressed.

Step 3: Click Next > button.

Step 4: A new Python 3.xx setup pop-up window will appear Select Destination

Step 5: Click the Next > button.

Step 6: Use the default customization, which selects the Python Interpreter and all its libraries (about 250 Mb).

Step 7: Click the Next > button.

Step 8: Click the Yes button on the following window.

Step 9: Click the Finish button.

Installation of Python Package:

Step 1: Connect System with Internet and Open Command Prompt

Step 2: Type “pip install –upgrade <Package-Name>” and click Enter.

Step 3: Verify Module Installation by Importing into the Code.

“import <Package-Name>”

Installing of SQLite3:

Step 1 – Download sqlite-shell-win32-*.zip and sqlite-dll-win32-*.zip zipped files.

Step 2 – Create a folder C:\>sqlite and unzip above two zipped files in this folder, which will give you sqlite3.def, sqlite3.dll and sqlite3.exe files.

Step 3 – Add C:\>sqlite in your PATH environment variable and finally go to the command prompt and issue sqlite3 command, which should display the following result.

CHAPTER 9

CONCLUSION

The signature verification system utilizes image processing to effectively distinguish between genuine and forged signatures. Through preprocessing steps like normalization and noise reduction, input image quality is enhanced, bolstering the reliability of verification. Leveraging feature extraction and comparison algorithms such as cosine similarity, the system accurately quantifies the resemblance between signatures, enabling precise authentication decisions. Its performance metrics, including accuracy and precision, underscore its efficacy across various handwriting styles and conditions. The system's real-time capability enhances its practicality for security-sensitive applications. However, challenges like signature style variability and susceptibility to skilled forgeries necessitate ongoing refinement. Future enhancements, such as integrating deep learning-based techniques, hold promise for further improving system robustness and accuracy, ultimately advancing identity authentication technology and bolstering security measures.

CHAPTER 10

FUTURE ENHANCEMENT

Incorporating digital signature and document verification capabilities holds promise for further advancing the signature verification system. By integrating digital signature analysis, the system can expand its authentication capabilities beyond traditional handwritten signatures, accommodating modern electronic documents and transactions. Additionally, document verification functionality can be introduced to authenticate the integrity and authenticity of entire documents, complementing signature verification with a holistic approach to document security. This expansion would not only enhance the system's versatility but also address emerging needs in digital document management and authentication. By combining these features, the system can offer comprehensive security solutions for a wide range of applications, ensuring robust identity verification and document integrity in an increasingly digitized world.

APPENDIX-1
BASE PAPER

New Textural Features for Handwritten Signature Image Verification

Suriya Soisang¹

¹Electrical Engineering Department
Faculty of Engineering
Pathumwan Institute of Technology,
Bangkok, Thailand
email: suriya_si@cwss.ac.th

Suvit Poomrittigul²

²Software Engineering and Information System Department,
Faculty of Science and Technology
Pathumwan Institute of Technology,
Bangkok, Thailand
email: suvit@pit.ac.th

Abstract—In this paper, a new textural feature for solving offline handwritten signature verification is proposed. A new textural features method is developed by combining a Local Binary Patterns (LBP) method and a Gradient Quantization Angle (GQA) method. This proposed method is called Local Binary Patterns with Gradient Quantization Angle (LBPGQA), as developed by heuristic method to improve the precision of verification the offline signature image. The hypothesis for this study is to classify the distinctive handwritten signature individually with the actual signature angle and refraction for enhancing the signature fraud detection. The verification step is achieved by Artificial Neural Network (ANN) classifier and trained on genuine signatures. Furthermore, the test stage is performed on genuine signatures and skilled forgeries. The experiments are conducted on CEDAR datasets. The experimental results show that in the LBPGQA method outperforms classical features such as Histogram of oriented gradients and local binary patterns. Conclusively, this proposed method can verify the individual and distinctive handwritten signature and help to protect the signature fraud by skilled forgeries.

Keywords—Handwritten Signature Verification, Gradient Quantization Angle, ANN

I. INTRODUCTION

Biometrics technology is operated to secure variety of safety systems. The objective is to recognize personal appearance on physiological or behavioral traits. The first factor, the recognition is based on the assessment of biological traits, for instance, the fingerprint, face, iris, signature, etc. On the other hand, Biometrics technology which is known as the most popular tool and the lowest investment biometrical identification is hand handwritten signature for a wide range of applications, for example, bank transactions, formal contracts, and organization forms.

However, the personal indication via handwritten signature is still required a specific application for the clarification process due to the frequency of non-genuine signatures such as bank transection, contracts, official documents. This is why it is necessary to develop the solution of handwritten signature verification which is formed both an online pattern and an offline pattern [1].

Moreover, signatures are written through splendid equipment with an instrumented stylus that access information about the pen tip, such as the position, velocity or acceleration[2]. The online signature verification system is accommodated properly with higher accuracy than the offline system. However, the offline pattern should be taken when the signatures are written in advance over documents, for example bank check.

Hence, the offline signature verifications are still being researched [3],[4] for improvement the process of signature verification. The researchers [3],[4] can operated into two situations in order to deal with the intra-variability between an authentic and an imitated signatures. Their process of verifications use distance of statistics and surroundedness feature.

Over the period of years, the textural qualification especially Local Binary Patterns (LBP) is widespread using for identification the variety signature characteristics and recognizing an authentic signatures the patterns [5]. Moreover, LBP is operated to the gradient level of pixels information on signature images such as, wavelet transforms and Gabor filters [6],[7]. There is the notably development of LBP feature encoding by comparing the neighborhood pixels using gradient calculation[8]. It was called this qualification by Gradient Local Binary Patterns (GLBP).

For over the years, a number of works are performed offline signature verification improvement with LBP and the Support Vector Machines (SVM) [3],[9]. Many researches also use the combination LBP with other method such as Hidden Markov model based system, HMN-based structures with Neural network and a fusion of statistical probability form. It is eminent to provide remarkable effects of the results [10].

In this work, we observe the benefit of novel LBP with the hybrid algorithm for handwritten signature verification system. A New Textural Features by Local Binary Gradient Quantization Angle Pattern is proposed with ANN classifier to enhance the offline signature verification efficiency. The rest of this paper is prepared as the following below.

II. SIGNATURE VERIFICATION SYSTEM

The handwritten signature verification method depends on the feature and characteristic of handwritten signature by the owner in order to decide whether the asked signature is genuine or non-genuine. Therefore, we propose a new textural feature for offline handwritten signature verification system as show in Figure 1.

A new textural features technique is Local Binary Gradient Quantization Angle Patterns (LBGQAP). This proposed method is to display the prominent point of Local Binary Patterns (LBP) which is the most effective technique to using gradient calculation in order to calculate histogram of gradient as developed by heuristic method to improve the precision of verification the offline signature images with Gradient Quantization Angle (GQA) to classify hypothetically distinctive in each personal signatures can be explained this theory as follows.

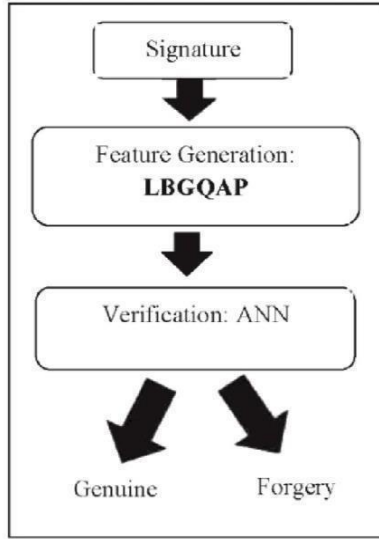


Fig. 1. The system of the signature verification.

A. Local Binary Patterns (LBP)

LBP is the most efficient technique to store display property of surface. It is remind that LBP is operated to increase the statistical and structural analysis of textural patterns by considering the pixel scale of the center area to refer the calculation. The result will be transformed consequently into binary number and adapted to histogram displaying the feature surface qualification as Figure 2 as show how LBP function operates.

LBP has been developed to the mare effective by a circular formatting in order to be applied more flexible. By all means, it is provided two variables. P is number of positions around the center which may 8 or 16 point per one center. R is represented the radius [11] between P point and the center point of the circle as Figure 3.

From the initial process, it can calculate the LBP in each given area as this equation (1).

$$LBP(x_c, y_c) = \sum_{n=0}^{N-1} f(i_n - i_c) 2^n \quad (1)$$

When i_c is the point of the pixel at the center. i_n is the point of the pixel surrounded the center any x_c, y_c is the amount of bit which is 8 and $f(x)$ can be replaced as this equation (2).

$$f(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2)$$

3x3 Pixel Data	Threshold	Weight																											
<table border="1"><tr><td>8</td><td>5</td><td>2</td></tr><tr><td>9</td><td>5</td><td>4</td></tr><tr><td>1</td><td>7</td><td>6</td></tr></table>	8	5	2	9	5	4	1	7	6	<table border="1"><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td></td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr></table>	1	0	0	1		0	0	1	1	<table border="1"><tr><td>1</td><td>2</td><td>4</td></tr><tr><td>128</td><td></td><td>8</td></tr><tr><td>64</td><td>32</td><td>16</td></tr></table>	1	2	4	128		8	64	32	16
8	5	2																											
9	5	4																											
1	7	6																											
1	0	0																											
1		0																											
0	1	1																											
1	2	4																											
128		8																											
64	32	16																											

Pattern=10110001

LBP=128+32+16+1=177

Fig. 2. The process of Local Binary Pattern.

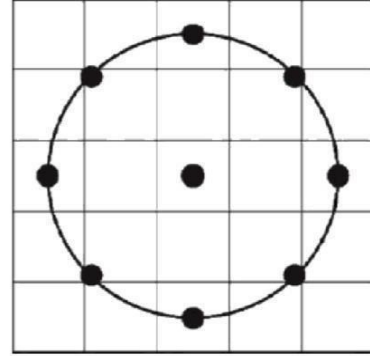


Fig. 3. Local Binary Pattern Circular when P=8, R=2.

B. Gradient Quantization Angle

The method developed by quantization index modulation (QIM) [12] by collecting distribute of the angle of refraction of the gradient signature of LBP function quantization shown by $Q(\theta)$ in the range of real angles and binary numbers as this equation (3).

$$Q(\theta) = \begin{cases} 0, & \text{if } \lfloor \theta / \Delta \rfloor \text{ is even} \\ 1, & \text{if } \lfloor \theta / \Delta \rfloor \text{ is odd} \end{cases} \quad (3)$$

Where the positive real number Δ indicated represents the angular quantization $\lfloor \cdot \rfloor$ step size and the size of function indicate the angle of the signature.

• If, $Q(\theta) = \omega$, then $Q(\theta)$ takes the value of angle at the center of the sector it lies in.

• If, $Q(\theta) \neq \omega$, then $Q(\theta)$ takes the value of angle at the center of one of the two adjacent sectors, whichever is closer to $Q(\theta)$.

These rules can be show as equation (4).

$$\theta^o = \begin{cases} \Delta\theta/\Delta - \Delta/2, & \text{if } Q(\theta) = \omega \\ \Delta\theta/\Delta + \Delta/2, & \text{if } Q(\theta) \neq \omega \text{ and } \theta > (\Delta\theta/\Delta - \Delta/2) \\ \Delta\theta/\Delta - \Delta/2, & \text{if } Q(\theta) \neq \omega \text{ and } \theta \leq (\Delta\theta/\Delta - \Delta/2) \end{cases} \quad (4)$$

When we get gradient angles, they will be considered and rotated every angle of gradient feature and divided quantization angle into 8 ranges of 360 degree each range is consist of 45 degree by calculating as this equation (4) all angle in which angle each gradient falls show as Figure 4.

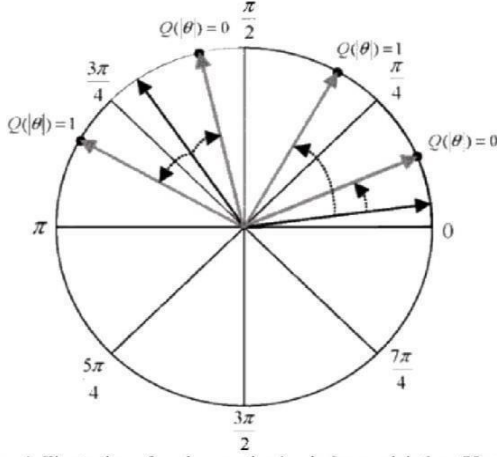


Fig. 4. Illustration of angle quantization index modulation.(Vectors before and after the angle of refraction of the gradient signature are represented by black and gray arrows, respectively.)

C. Classification

The classifier we have applied is the Artificial Neural Network structure duplicated by the mathematical and circumstance of the structure known as the Back Propagation Network that has been received from the generalized Delta Learning rule which aims to reduce error in posterior reiteration during the training phase [13]. This is the most prevalent and generalized neural network presently in use.

The attributive feature vector is given as input to the ANN. These inputs can be considered in conformity to their importance to give an input vector that is a priority oriented considered version of the one, which the feature attribution section primarily presents [14]. We have applied three grades of importance as concluded from the statistics, 34 input neuron nodes, dues to the feature size of signature is 34 input, it occurs from the compilation of LBP feature and GQA feature. In the testing, the hidden layer of 11 is the most appropriate accuracy value and one output neuron compose the connected neural network applied to classification.

III. EXPERIMENTAL RESULTS

The performance evaluation for the proposed signature verification system is assessed on the CEDAR dataset contains the signatures of 55 volunteer signers belonging to versatile cultural backgrounds. For each writer, there are 24 genuine signatures and 24 skilled forgeries. So, this yields 1320 genuine and the same number for skilled forgeries. All signatures were scanned at 300 dpi in 8-bit gray-level as Figure 5(a) and Figure 5(b), which depicts some samples from adopted datasets. After that take the signatures picture LBP feature encoding by comparing the neighborhood pixels using gradient calculation and perform feature extraction to classification as Figure 6 as show how LBP image conversion.

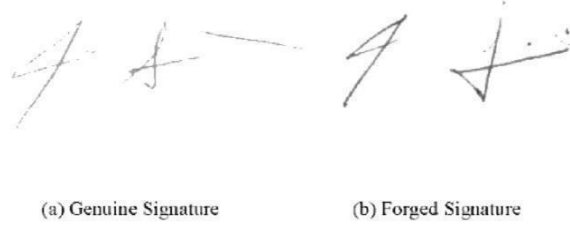
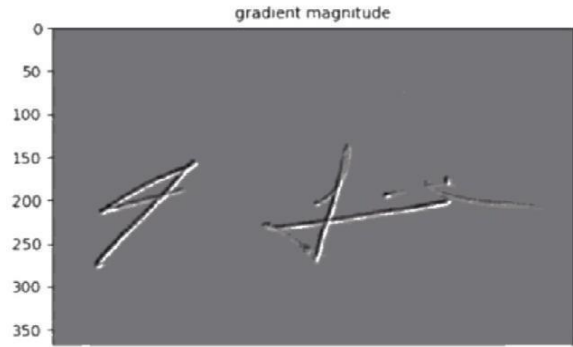
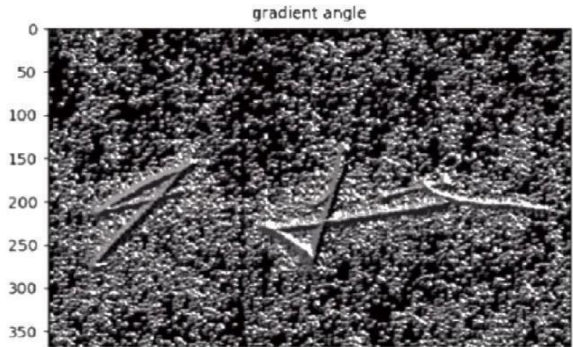


Fig. 5. Samples from adopted datasets.



(a) gradient magnitude



(b) gradient angle

Fig. 6. LBP image conversion.

TABLE I. THE CONFUSION MATRIX FOR LBP.

	ACTUAL POSITIVE	ACTUAL NEGATIVE
PREDICTED POSITIVE	53	40
PREDICTED NEGATIVE	19	108

TABLE II. THE CONFUSION MATRIX FOR LBGQAP.

	ACTUAL POSITIVE	ACTUAL NEGATIVE
PREDICTED POSITIVE	51	22
PREDICTED NEGATIVE	8	139

Table I and Table II shows a confusion matrix for LBP and LBGQAP, whose entries have the following meanings:

- *TP* is the number of correct positive predictions;
- *FP* is the number of incorrect positive predictions;
- *FN* is the number of incorrect negative predictions;
- *TN* is the number of correct negative predictions.

The prediction accuracy, precision, recall and F-measure can be obtained from this matrix as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (5)$$

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (8)$$

- Accuracy is the correct prediction value.

- Precision is the precision that is only interested in the part that the model is predicting is the class under consideration.

- Recall is the precision interested in the part of the actual event that has occurred.

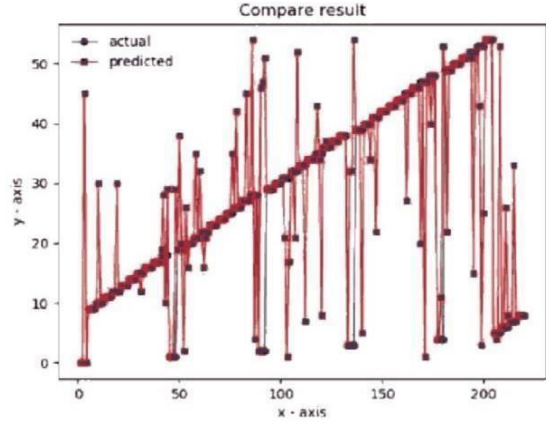
- F-measure is the calculation of a model is performance measure.

The present work uses ANN classify process can produce as from Table III, it is shown that the result of Train accuracy and Test accuracy of LBGQAP is better than LBP in 97.75%, 86.36% respectively. Moreover, the result of Precision, Recall, and F-Measure of LBGQAP is also better than LBP.

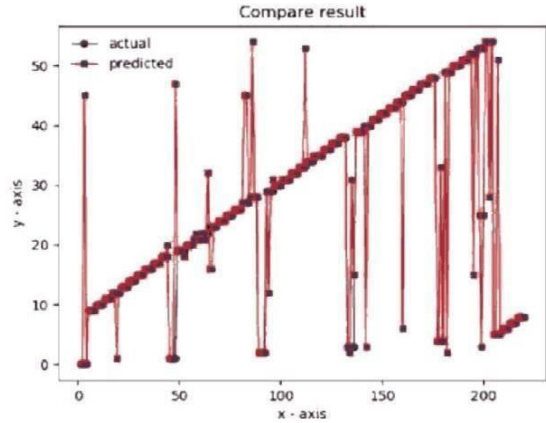
TABLE III. COMPARISON OF LBGQAP WITH LBP FEATURES

Features	Train accuracy	Test accuracy	Precision	Recall	F- Measure
LBP	95.41%	73.18%	0.571	0.732	0.732
LBGQAP	97.75%	86.36%	0.700	0.864	0.864

Furthermore, the results shows that our proposed is more accurate than the existing LBP, as shown in Figure 6(b), compared to LBP in Figure 6(a).



(a) LBP



(b) LBGQAP

Fig.6 signature verification results using the proposed system

IV. CONCLUSION

This work proposed a new texture features by using Gradient Quantization Angle (GQA) based on local binary patterns (LBP). A new textural features technique is called the Local Binary Gradient Quantization Angle Patterns (LBGQAP). We also combined ANN classifiers for solving offline handwritten signature verification. The experimental analysis was carried out on CEDAR datasets. The results showed that LBGQAP allows satisfactory performance with a train accuracy and test accuracy of over 2% and 13% respectively. Additionally, the LBGQAP's Precision Recall and F-Measure scores have more model accuracy than LBP. Motivated by these results, we plan as future work, to employ the LBGQAP features for implementing the own datasets with Thai signature verification.

REFERENCES

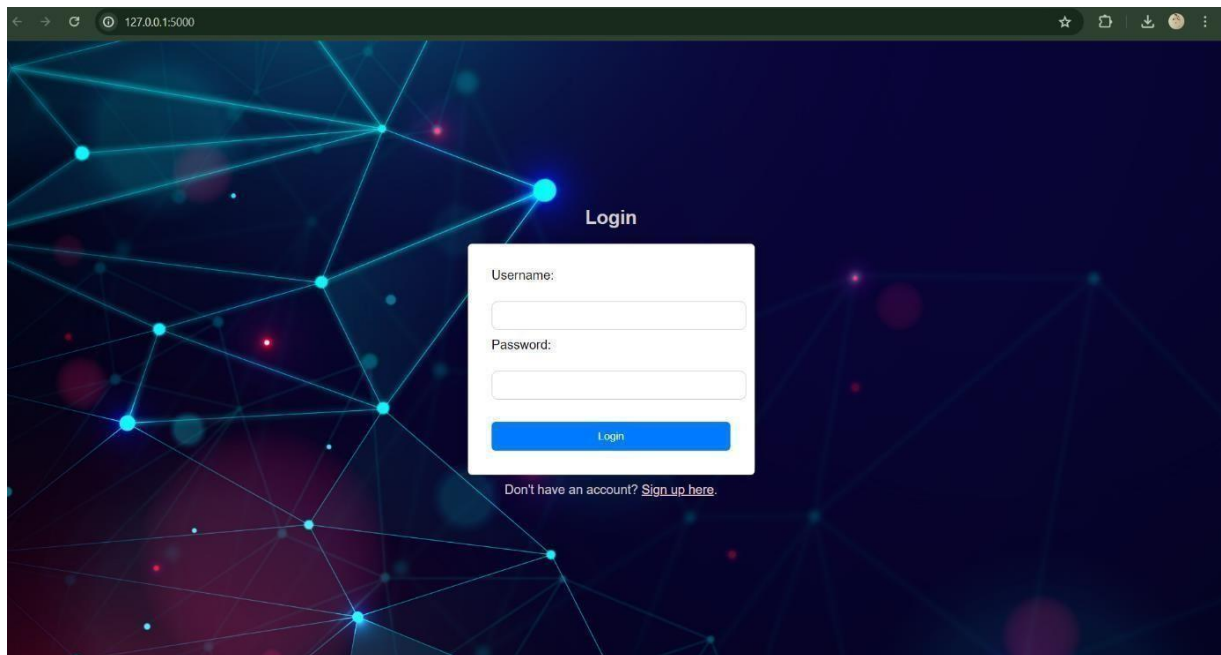
- [1] D. Impedovo, R. Modugno, G. Pirlo, and E. Stasolla, "Handwritten signature verification by multiple reference sets," *IEEE International Conference on Frontiers on Handwritten Recognition (ICFHR'08)*, Montreal, pp. 19–21, August 2008.
- [2] R. Plamondon and S. N. Srihari, "On-line and on-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 63–84, January 2000.
- [3] M. K. Kalera, S. Srihari, and A. XU, "On-line signature verification and identification using distance statistics," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, pp. 1339–1360, November 2004.
- [4] R. Kumar, J. D. Sharma, B. Chandra, "Writer-independent on-line signature verification using surroundedness feature," *Journal of Pattern Recognition Letters*, vol. 33, pp. 301–308, February 2012.
- [5] J. F. Vargas, M. A. Ferrer, C. M. Travieso, and J. B. Alonso, "On-line signature verification based on gray level information using texture features," vol. 44, pp. 375–385, February 2011.
- [6] J. F. Vargas, C. M. Travieso, J. B. Alonso, and M. Ferrer, "On-line signature verification based on gray level information using wavelet transform and texture features," in *Proc. IEEE International Conference on Frontiers in Handwritten Recognition (ICFHR'10)*, Kolkata India, pp. 587–592, November 2010.
- [7] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang Local, "Gabor binary pattern histogram sequence (1gbphs): A novel non-statistical model for face representation and recognition," in *Proc. IEEE International Conference on Computer Vision (ICCV'05)*, Beijing, pp. 786–791, October 2005.
- [8] N. Jiang, J. Xu, W. Yu, and S. Goto, "Gradient local binary patterns for human detection," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS'13)*, Beijing, pp. 978–981, May 2013.
- [9] I. Güler and M. Medhadi, "A different approach to on-line handwritten signature verification using the optimal dynamic warping algorithm," vol. 18, pp. 940–950, November 2008.
- [10] K. Sisodia and S. M. Anand, "On-line Handwritten Signature Verification using Artificial Neural Network Classifier," *International Journal of Recent Trends in Engineering*, vol. 2, no. 2, pp. 205–207, November 2009.
- [11] T. Ojala, M. Pietikinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," vol. 29, pp. 51–59, January 1996.
- [12] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inf. Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.
- [13] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 39, pp. 2101 – 2104, September 1991.
- [14] Carlos Gershen son, "Artificial Neural Networks for Beginners", arxiv.org

APPENDIX 2

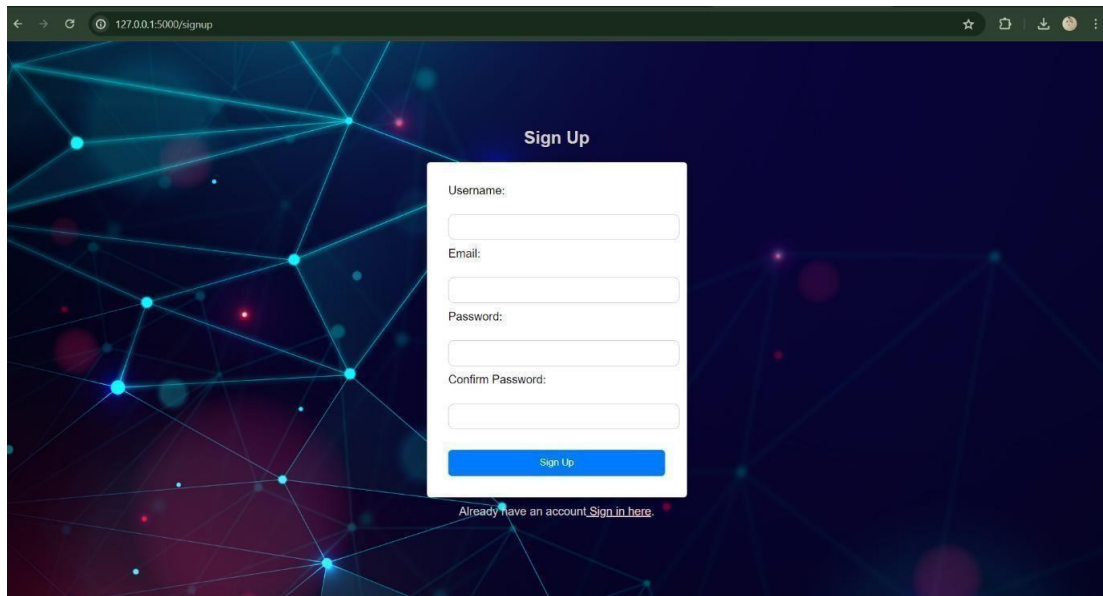
SCREENSHOTS

OUTPUT SCREENSHOTS:

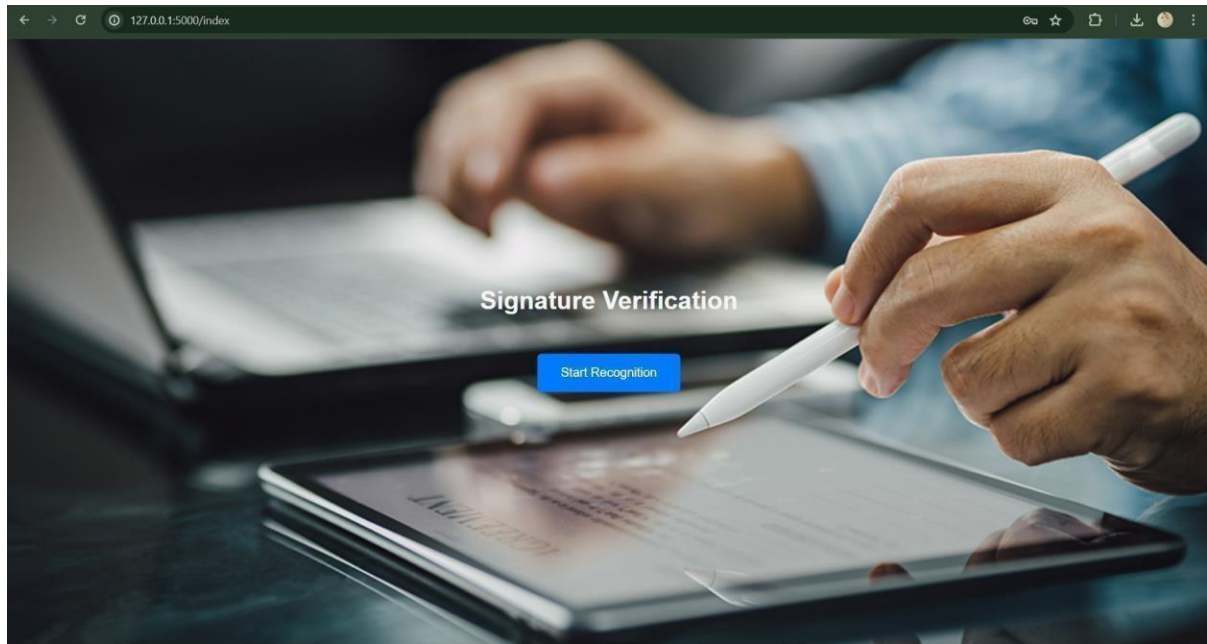
LOGIN PAGE



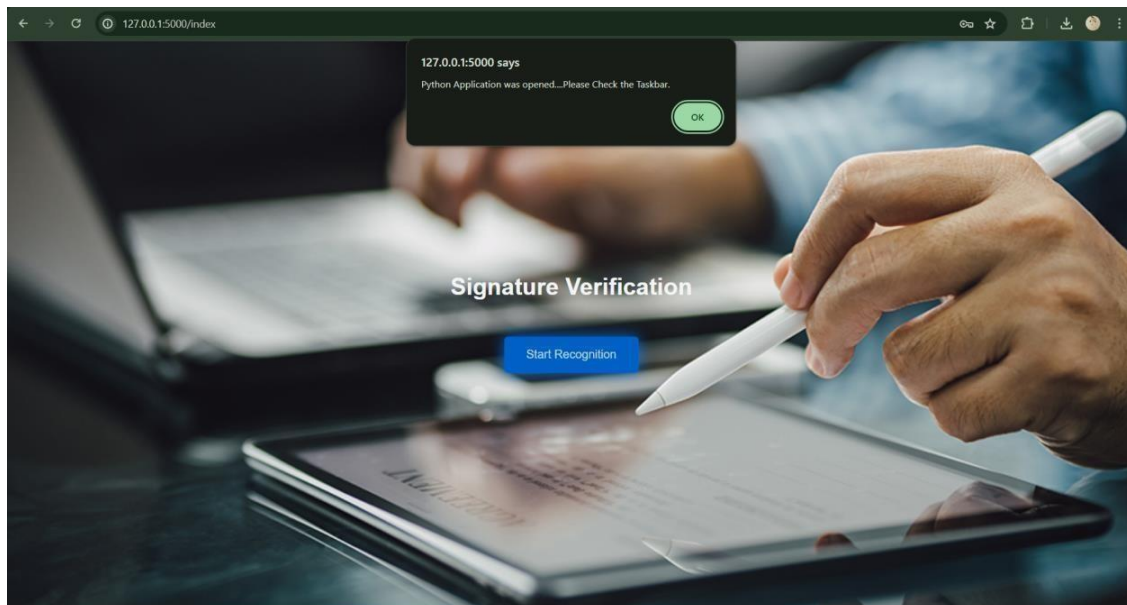
SIGNUP PAGE:



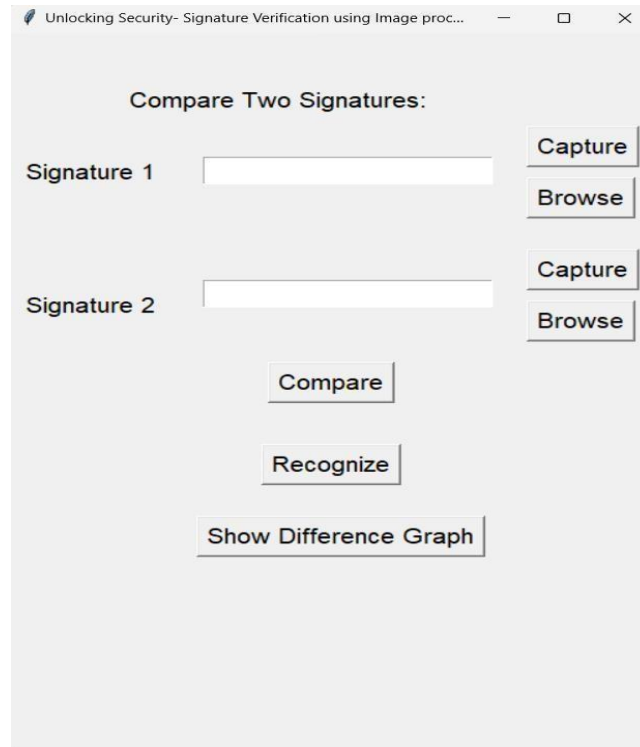
HOMEPAGE:



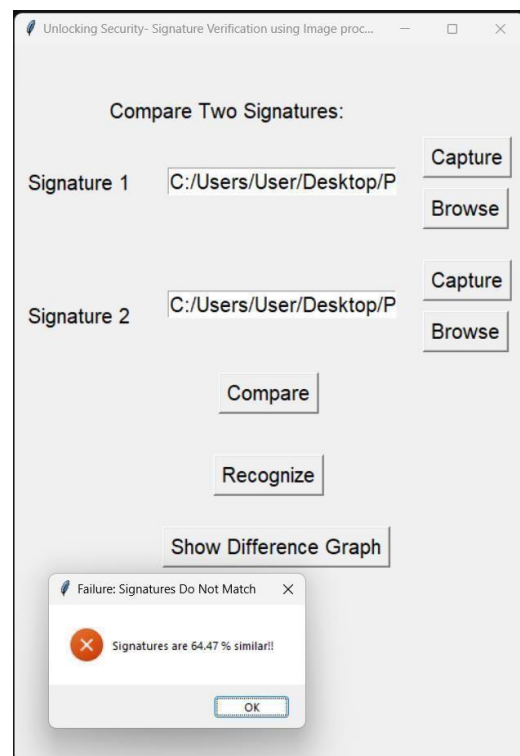
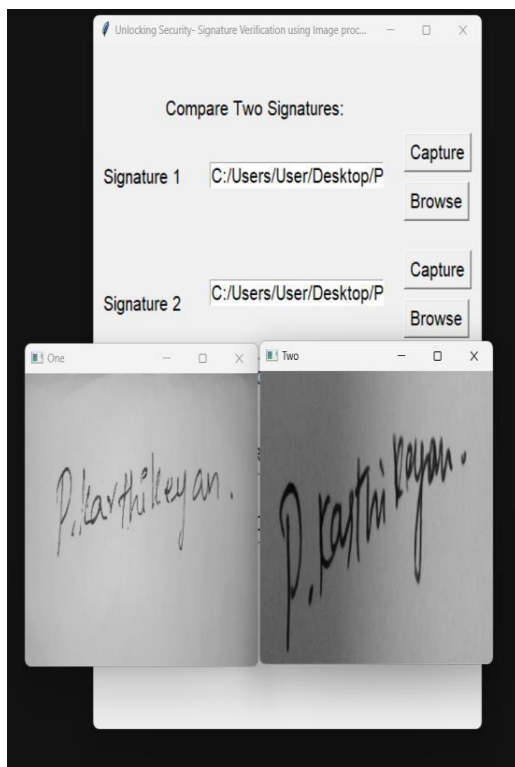
Home page with Notification Message:



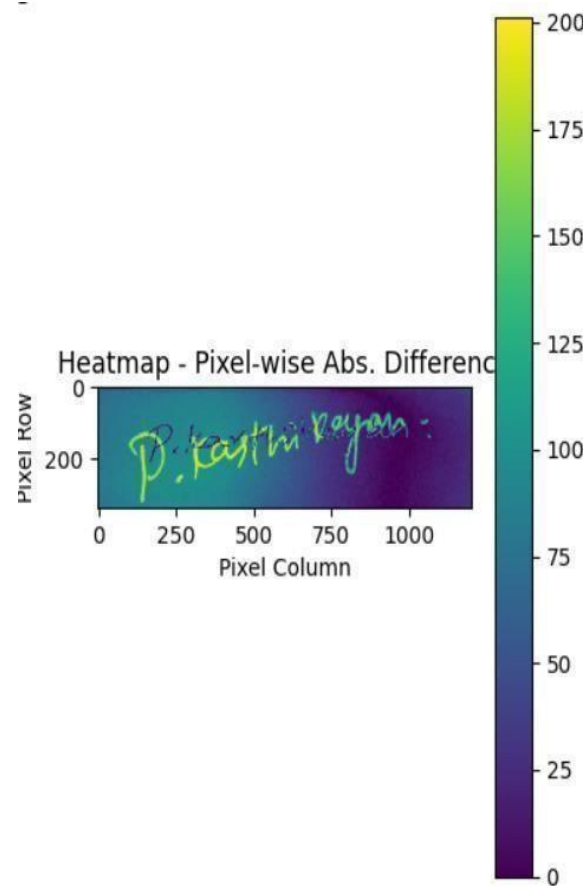
PYTHON APP:





RESULT FROM PYTHON APP:



OVERALL RESULTS:

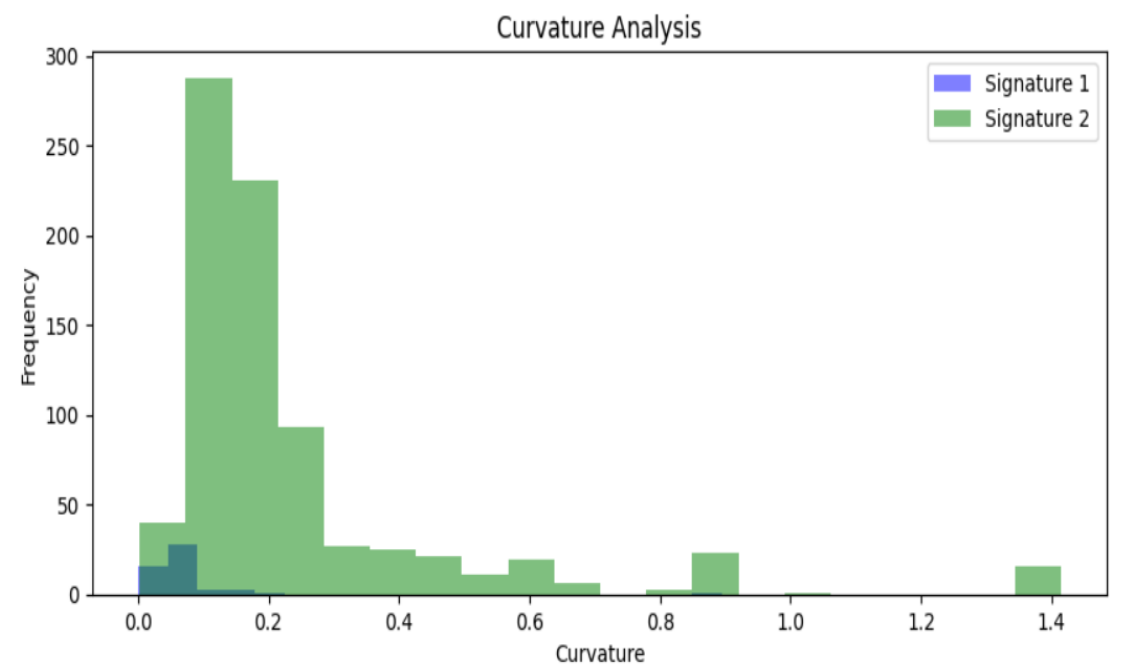
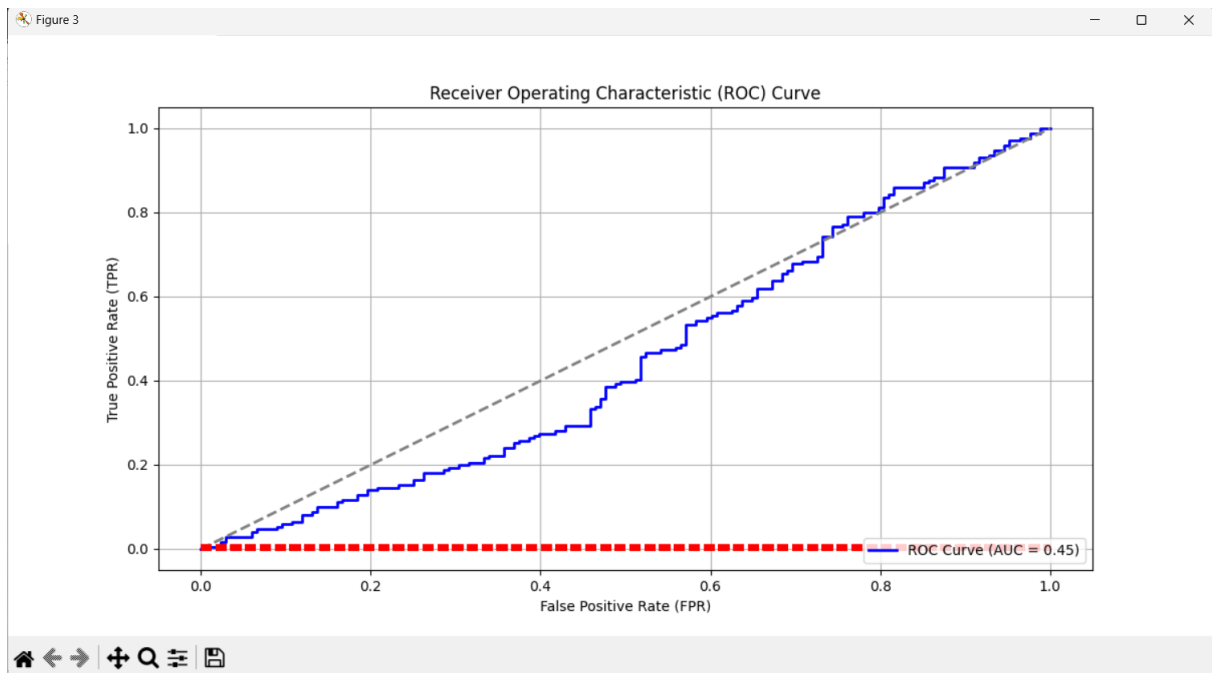


 Recognized Values ×



Chi-Squared Distance: 749.6413322859207
Pressure 1: 2.9428720746283648
Pressure 2: 1.547053954546236
precision: 0.99, recall: 0.50, f1_score: 0.66
('Stroke for Image 1:', 54, 'Stroke for Image 2:', 599)

OK



APPENDIX 3
PUBLICATION

UNLOCKING SECURITY: ENHANCED SIGNATURE VERIFICATION SYSTEM USING IMAGE PROCESSING

Karthikeyan P¹, Muralidharan U², Mrs. R. Pratheeba, M.E³,

^{1,2} Student, ³ Assistant Professor,

^{1,2,3} Department of CSE.

^{1,2,3} Anand Institute Of Higher Technology, Chennai,

Mail Id: Karthik4606k@gmail.com, umurali733@gmail.com, pratheebabar.@aiht.ac.in

ABSTRACT

The "Unlocking Security: Enhanced Signature Verification System using Image Processing" project is a transformative initiative aimed at overcoming the limitations inherent in traditional manual signature verification methods. By harnessing the power of cutting-edge technologies, including advanced image processing techniques, our project seeks to revolutionize the way signatures are authenticated. The key components of the system involve the process of feature extraction and pattern recognition, coupled with state-of-the-art image processing methods to enhance the clarity and quality of signature images.

With a versatile scope, this system is designed to find applications across various industries, offering a secure and efficient solution for document authentication. The primary objective is to develop a sophisticated and highly accurate Signature Verification System, complemented by secondary objectives that focus on streamlining the verification process, enhancing the visualization of the signature verifying and ensuring seamless integration.

Our aim is to establish a more secure and user-friendly document authentication process, shaping the future of signature verification through innovation and adaptability. By design an algorithm for extracting features from signature images and develop a method for comparing and verifying signatures based on extracted features and implement a user-friendly interface for signature input and verification.

Keywords: *Image Processing, Structural Similarity Index [SSIM], Signature Verification, Feature Extraction, Handwritten Signature Verification.*

1. INTRODUCTION

In today's digital age, the verification of signatures holds immense importance across numerous sectors, including finance, legal affairs, and administrative processes. Signatures serve as a fundamental method of authentication, validating the identity of individuals and

confirming their consent or approval on various documents. However, traditional methods of manual signature verification are labor-intensive, time-consuming, and susceptible to errors.

As a result, there is a growing need for automated systems that can accurately and efficiently verify signatures, leveraging the capabilities of image processing technology. Manual signature verification typically involves visual inspection by trained professionals, who compare a submitted signature with reference samples to determine its authenticity. However, this process is subject to several limitations:

1. **Subjectivity:** Human judgment can vary, leading to inconsistencies in the evaluation of signatures.
2. **Time-Consuming:** Verifying signatures manually can be a time-consuming task, particularly when dealing with large volumes of documents.
3. **Expertise Required:** Skilled personnel are needed to accurately assess the validity of signatures, adding to the operational costs.

The Role of Image Processing:

Image processing techniques offer a promising solution to the challenges associated with manual signature verification. By analyzing digital representations of signatures, these techniques can extract and quantify various visual features, enabling automated comparison and authentication processes.

Key advantages of using image processing for signature verification include:

1. **Objective Analysis:** Image processing algorithms can objectively analyze signature images based on predefined criteria.
2. **Efficiency:** Automated signature verification systems can improving operational efficiency and reducing processing times.
3. **Accuracy:** By leveraging advanced image analysis techniques, such as pattern recognition, these systems can achieve high levels of accuracy in signature authentication.

Components of Signature Verification Using Image Processing:

1. **Image Acquisition:** Signature images are captured using digital scanners or cameras, ensuring high-quality input for the verification process.
2. **Pre-processing:** The captured images undergo preprocessing techniques, such as noise removal, binarization and normalization, to enhance their quality and suitability for analysis.
3. **Feature Extraction and Representation:** Relevant features are extracted from the pre-processed signature images, including stroke patterns, edge profiles, curvature, and texture and Extracted features are transformed into a suitable representation format, such as feature vectors or histograms, for further analysis.
4. **Comparison Algorithm:** A comparison algorithm is applied to measure the similarity between the features of the input signature and reference samples.
5. **Decision Making:** Based on the comparison results, a decision is made regarding the authenticity of the input signature, taking into account predefined threshold values or statistical measures.

Applications of Signature Verification:

- Banking and Finance
- Legal Affairs.
- Administrative Processes.
- Security Systems.

2. LITERATURE SURVEY

The proposed method for offline signature verification using image processing techniques starts with preprocessing the scanned signature images. This preprocessing step is crucial as it helps isolate the signature part from the rest of the image and removes any noise or unwanted elements that could affect the accuracy of verification. By enhancing the quality of the images and ensuring uniformity, preprocessing lays the foundation for accurate feature extraction.

Feature extraction plays a pivotal role in the verification process. In this paper, simple shape-based geometric features are utilized. These features are derived from the signature images and capture essential characteristics such as baseline slant angle, aspect ratio, normalized area, center of gravity, number of edge points, number of cross points, and the slope of the line joining the centers of gravity of two halves of a signature

image. These features are chosen for their ability to effectively distinguish between genuine signatures and forgeries.

Once the features are extracted, the system compares them with a template signature to determine the authenticity of the signature under consideration. However, the paper highlights certain challenges encountered during this process. One significant challenge is the limited availability of signature data for robust parameter estimation. This scarcity of data can affect the system's ability to accurately classify various signature styles, especially when dealing with signatures from diverse individuals with different writing habits.

Despite these challenges, the research underscores the importance of offline signature verification in ensuring security and authenticity in various applications. It emphasizes the need for further exploration and development of techniques to address the identified limitations and enhance the overall performance of signature verification systems.

In conclusion, while the proposed method shows promise in offline signature verification, there is room for improvement, particularly in addressing the performance deterioration in detecting skilled forgeries. Future research directions could involve exploring advanced feature extraction techniques, incorporating dynamic information from the signing process, and evaluating the system's performance across diverse datasets to achieve higher accuracy and reliability in signature verification.

3. PROPOSED METHODOLOGY

1. Image Acquisition:

Acquire signature images from different sources such as scanned documents, digital devices, or biometric sensors. Ensure uniformity in image resolution and quality to facilitate consistent processing.

2. Preprocessing:Image

Enhancement: Employ techniques like histogram equalization, contrast stretching, or adaptive filtering to enhance the visual quality of signature images.

Noise Reduction: Apply filters such as median filtering or Gaussian smoothing to remove noise and artifacts from the images, ensuring clean input for subsequent processing steps.

3. Segmentation:

Foreground Extraction: Utilize thresholding or edge detection algorithms to separate the signature region from the background.

Connected Component Analysis: Identify and isolate individual components representing signature strokes or characters.

Stroke Width Transform: Detect and segment individual strokes of the signature based on variations in stroke width.

4. Feature Extraction:

Shape-Based Features: Extract geometric properties such as curvature, aspect ratio, and slant angle of signature strokes.

Texture Features: Compute texture descriptors such as Gabor filters, local binary patterns (LBP), or histogram of oriented gradients (HOG) to capture textural patterns within the signature.

Local Descriptors: Utilize keypoint-based descriptors like Scale-Invariant Feature Transform (SIFT) or Speeded Up Robust Features (SURF) to identify distinctive points and regions within the signature.

5. Template Creation:

Feature Aggregation: Combine extracted features from multiple samples of genuine signatures to create representative templates for each signer.

Normalization: Normalize feature vectors to account for variations in scale, orientation, and position across different signatures.

6. Comparison:

Similarity Metrics: Compute similarity scores between the features of the test signature and the stored templates using distance-based metrics like Euclidean distance, cosine similarity.

Thresholding: Establish decision thresholds to determine the acceptance or rejection of a test signature based on the computed similarity scores.

Multimodal Fusion: Integrate similarity scores from multiple feature modalities (e.g., shape, texture) to improve verification accuracy and robustness.

7. Decision Making:

Threshold Adjustment: Fine-tune decision thresholds based on performance evaluation metrics and application-specific requirements to achieve desired levels of false acceptance and false rejection rates.

Statistical Modelling: Employ statistical methods such as Bayesian inference or support vector machines (SVMs) to model the decision-making process and adaptively adjust decision boundaries.

8. Evaluation and Optimization:

Performance Metrics: Evaluate the performance of the signature verification system using metrics such as accuracy, false acceptance rate (FAR), false rejection rate (FRR), and receiver operating characteristic (ROC) curves.

Cross-Validation: Perform cross-validation experiments to assess the generalization ability of the system and identify potential overfitting or underfitting issues.

Parameter Tuning: Optimize system parameters, feature selection methods, and classifier configurations through iterative experimentation and validation.

9. Iterative Improvement:

Feedback Mechanisms: Incorporate feedback from end-users, domain experts, and performance evaluation results to iteratively refine and enhance the system.

Adaptive Learning: Employ machine learning techniques such as online learning or transfer learning to adapt the system to evolving signature patterns and emerging forgery techniques over time.

4. RESULT AND ANALYSIS

The signature verification system utilizing image processing compares two signature images by computing a similarity score. If the similarity score surpasses a predetermined threshold, the system categorizes the signatures as genuine; otherwise, they are deemed forged.

Upon receiving input, the system proceeds with comparing the extracted features of the signatures and calculates their resemblance using a suitable similarity measure, such as cosine similarity or Euclidean distance. The obtained similarity score is then contrasted with the predefined threshold to make the authentication decision.

The system's performance is extensively elaborated. This includes detailing the similarity score computation process, highlighting any preprocessing steps involved in standardizing the input images, and specifying the similarity measure employed. Accuracy metrics are presented, such as true positive rate, false positive rate, precision, and recall, to provide a

comprehensive evaluation of the system's effectiveness. Additionally, visual representations, like ROC curves or confusion matrices, can further elucidate the system's performance across different threshold values.

The strengths and limitations of the system are thoroughly examined. The advantages of this approach, such as its simplicity and computational efficiency, are acknowledged, particularly in scenarios where real-time authentication is crucial. However, limitations concerning variability in signature styles, susceptibility to skilled forgeries, and the impact of image quality on similarity scores are also addressed. Potential avenues for improvement are discussed, such as incorporating advanced feature extraction techniques or leveraging deep learning models to enhance the system's robustness and accuracy.

5. CONCLUSION

The signature verification system utilizes image processing to effectively distinguish between genuine and forged signatures. Through preprocessing steps like normalization and noise reduction, input image quality is enhanced, bolstering the reliability of verification. Leveraging feature extraction and comparison algorithms such as cosine similarity, the system accurately quantifies the resemblance between signatures, enabling precise authentication decisions. Its performance metrics, including accuracy and precision, underscore its efficacy across various handwriting styles and conditions. The system's real-time capability enhances its practicality for security-sensitive applications. However, challenges like signature style variability and susceptibility to skilled forgeries necessitate ongoing refinement. Future enhancements, such as integrating deep learning-based techniques, hold promise for further improving system robustness and accuracy, ultimately advancing identity authentication technology and bolstering security measures.

6. REFERENCES

1. P.R.Shahane , A.S.Choukade , A.N.Diyewar (2015), "Online Signature Recognition Using Matlab" , International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering Vol. 3, Issue 2, February 2015.
2. Mirudu Basini K S, Gopinath R (2021), "Signature Recognition Using Image Processing", IJARIE-ISSN(O)-2395-4396 Vol-7 Issue-2 2021.
3. B.Akhila, G. Nikhila, A Lakshmi , G. Jahnavi, Mrs. J. Himabindhu (2021), "Signature Verification Using Image Processing And Neural Networks", International Journal of Creative Research Thought,(Volume 9, Issue 8 August 2021.
4. Pallavi V. Hatkar, Zareen J Tamboli (2015), "Image Processing for Signature Verification", International Journal of Innovative Research in Computer Science & Technology (IJRCST) I Volume-3, Issue-3, May- 2015.
5. P. N. Narwade, R. R. Sawant, and S. V. Bonde (2018), "Offline signature verification using shape correspondence", International Journal of Biometrics, vol. 10, no. 3, pp. 272–289, 2018.
6. Hemant A. Wani , Kantilal Rane and V. M. Deshmukh (2023), "A Comparative Study On Signature Identification And Verification System", International Journal of Applied Engineering & Technology Vol. 5 No.4, December, 2023.
7. Poddar J, Parikh V, Varti SK (2020), "Offline Signature Recognition & Forgery Detection using Deep Learning", The 3rd International Conference on Emerging Data and Industry 4.0 EDI40, Warsaw, Poland, April , 2020.
8. Sayali Gowre (2022), "Signature Verification using Image Processing & Neural Network", International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 2, Issue 2, April 2022.
9. Amina Khatra (2013), "Signature Verification Using Image Processing Techniques", International Global Research Analysis Volume -2 Issue -12,| Dec 2013.
10. [10]. Swarali Patil, Pranali Misal, Mayuri Mhaske, Prof Vilas Jadhav (2021), "Signature Verification using Image Processing & Neural Network", International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 03, Mar 2021.
11. [11]. M.SURUTHI, M.THAMIZHARASI (2021), "Signature Verification Using Matlab – Image Processing", International Conference on Intelligence Computing and Control Systems ICICCS, Paper ID: ICICCS192, May 2021.



CERTIFICATE

Dear Author(s),

This certifies that the research paper entitled '**UNLOCKING SECURITY: ENHANCED SIGNATURE VERIFICATION SYSTEM USING IMAGE PROCESSING**' authored by '**Karthikeyan P, Muralidharan U, Mrs. R. Pratheeba, M.E**' was reviewed by experts in this research area and accepted by the board of 'AG Publications' which has published in IJADST (International Journal of Advanced Development in Science and Technology), ISSN: 2582-1059 (Online), Volume: 6, Issue: 4, May 2024.

Your published paper and Souvenir are available at:

<http://ijadst.com/issue-details.php?issue=4&volume=6>

Thanks and warm regards,



Editor in Chief

Dr. San Murugesan.

REFERENCES

1. Amina Khatra (2013), “Signature Verification Using Image Processing Techniques”, International Global Research Analysis Volume -2 Issue -12,| Dec 2013.
2. Akhila, G. Nikhila, A Lakshmi , G. Jahnavi, Mrs. J. Himabindhu (2021), “Signature Verification Using Image Processing And Neural Networks”, International Journal of Creative Research Thought,\Volume 9, Issue 8 August 2021.
3. Hemant A. Wani , Kantilal Rane and V. M. Deshmukh (2023), “A Comparative Study On Signature Identification And Verification System”, International Journal of Applied Engineering & Technology Vol. 5 No.4, December, 2023.
4. Mirudu Basini K S, Gopinath R (2021), “Signature Recognition Using Image Processing”, IJARIIIE-ISSN(O)-2395-4396 Vol-7 Issue-2 2021.
5. Narwade P. N, R. R. Sawant, and S. V. Bonde (2018), “Offline signature verification using shape correspondence”, International Journal of Biometrics, vol. 10, no. 3, pp. 272–289, 2018.
6. Poddar J, Parikh V, Varti SK (2020), “Offline Signature Recognition & Forgery Detection using Deep Learning”, The 3rd International Conference on Emerging Data and Industry 4.0EDI40, Warsaw, Poland, April , 2020.

7. Sayali Gowre (2022), “Signature Verification using Image Processing & Neural Network”, International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 2, Issue 2, April 2022.
8. SURUTHI . M, M.THAMIZHARASI (2021), “Signature Verification Using Matlab - Image Processing”, International Conference on Intelligence Computing and Control Systems ICICCS, Paper ID: ICICCS192, May 2021.
9. Swarali Patil, Pranali Misal, Mayuri Mhaske, Prof Vilas Jadhav (2021), “Signature Verification using Image Processing & Neural Network”, International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 03, Mar 2021.