Signature Verification

Final Report

Submitted by

Suraj Singh Lalotra (21BCS8123)

Kushagra Saran (21BCS8066)

Lovish Shrivastava (21BCS8138)

Syed Furqan Jamal (21BCS8008)

Pranav Kumar (21BCS8033)

in partial fulfillment for the award of the degree of

Bachelor of Engineering

IN

Computer Science & Engineering

Chandigarh University

May, 2023

BONAFIDE CERTIFICATE

Certified that this project report **“Signature Verification”** is the bonafide work of “**Suraj Singh Lalotra (21BCS8123), Kushagra Saran (21BCS8066), Lovish Shrivastava (21BCS8138), Syed Furqan Jamal (21BCS8008), Pranav Kumar (21BCS8033)”** who carried out the project work under my/our supervision.

SIGNATURE	**SIGNATURE**
Dr. Sandeep Singh Kang	Er. Ankesh Gupta (E13154)
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Computer Science & Engineering	Computer Science & Engineering

Submitted for the project viva-voce examination held on_

INTERNAL EXAMINER **EXTERNAL EXAMINER**

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Signature verification is an important task in the field of document analysis and recognition. In order to increase security and reduce fraud in places like a bank, signature verification is very much important. It increases accuracy and efficiency.

This model typically involves a combination of image processing and machine learning techniques. Initially, the model extracts feature from the signature image, such as stroke density, curvature, and texture. The machine learning algorithm we are going to use is Convolutional Neural Network (CNN), so we can classify the signature as genuine or forged.

One of the challenges in developing a signature verification model is dealing with the variations in signature styles that can occur due to the natural variability of handwriting or intentional forgery. The model must be trained on a large and diverse dataset of signature samples to account for these variations and give us the best output.

Overall, a signature verification model is an essential tool for improving document security and preventing fraud. With advancements in image processing and machine learning techniques, these models are becoming increasingly accurate and reliable.

CHAPTER 1
INTRODUCTION

1.1 Client Identification/Need And Of Relevant Contemporary Issue

Authenticating a person’s identity by signature is the most social and legal way to use and therefore is significantly confronted with high-level attacks. One of the important issue that needs to be considered is the increasing occurrence of signature fraud in various industries, including banking and legal services. This has resulted in the need for reliable and efficient signature verification systems to prevent fraud and protect the integrity of important documents.

In addition, the identification of relevant contemporary issues such as the use of digital signatures and the need for secure and efficient remote verification is also crucial. With the growing use of electronic documents, digital signatures have become more common, and therefore, the signature verification system should be capable of verifying both traditional and digital signatures.

Moreover, remote verification has become necessary in many industries due to the COVID-19 pandemic. A signature verification system should be able to efficiently verify signatures remotely without compromising security and accuracy.

1.2 Identification Of Problems

Handwritten signatures can be easily forged or manipulated, making it difficult to differentiate between genuine and fake signatures. It is very much impossible for a human to authenticate each sign of a person in such a quantity which even if someone will try to do will take a lot of time and will be very expensive. One of the main challenges in signature verification is the natural variation in handwriting, which can make distinguishing between genuine and forged or fake signatures difficult. Additionally, forgers can use advanced techniques to imitate a person's signature, making it even more challenging to detect fraud.

Moreover, with the increasing use of digital documents and digital signatures, there is a need for signature verification systems that can effectively verify both traditional and digital signatures.

Overall, the problem of signature verification arises from the need to distinguish between genuine and forged signatures while considering the natural variation accurately and efficiently in handwriting, advanced forgery techniques, and the need for remote and digital verification.

1.3 Identification Of Tasks

The tasks of a signature verification model typically include:

1. **Data Collection:** The first step is to collect a relevant dataset of signature images that includes both genuine and forged signatures. We can collect a genuine dataset from the Kaggle website Kaggle: Your Machine Learning and Data Science Community

2. **Data Pre-processing:** The second step that the team has to work on is pre-processing the signature images to remove any noise, artifacts, or other imperfections that may affect the performance of the model. This can include resizing, normalization, and other image processing techniques.

3. **Feature Extraction:** Extracting features from the signature images is an important task to distinguish between genuine and forged signatures. This can be done using techniques like PCA

4. **Algorithm Used:** The machine learning algorithm we are going to use is Convolutional Neural Network (CNN).

5. **Interface:** While working on the model, we will also create an easy-to-use GUI for our model with the help of the Tkinter library.

1.4 Timeline

The timeline to create our project depends upon various factors like preparing signatures, the verification process, quality control which may take about 4 or 5 weeks, and at the end reviewing our model to verify and validate any errors we have so we can improve the quality of its accuracy furthermore.

1.5 Organization Of The Report

Chapter 1: Introduction
In this section, we are going to provide an overview of the report, including the purpose, scope, and objectives of the signature verification model.

Chapter 2: Literature survey
In this section, we are going to discuss the research and projects which people have already created on signature verification models and what we have learned from them, including the technique and algorithm which we have used in our model.

Chapter 3: Design flow/Process
This section outlines the methodology used to develop the signature verification model, including the data collection and preprocessing techniques, feature extraction and selection methods, classification, and about how to model is going to work and look.

Chapter 4: Results analysis and validation
In this section, we are going to discuss the result we are getting from our model and will validate if our model is meeting all the requirements.

Chapter 5: Conclusion and Future Scope
In the conclusion, we are going to discuss what we have learned from and the future scope of our model as to how this project can be much more useful to our society and how we can work on it to improve furthermore.

CHAPTER 2

LITERATURE SURVEY

2.1 Literature Review

An important aspect of a signature verification model is its accuracy in analyzing legal documents, financial contracts, etc. It is a signature that is often used as evidence, such as confirmation.

• This research paper by “Amr Hefny and Mohamed N Moustafa” focus on signature modeling using Legendre polynomial coefficients as one of the features. In addition, DLLs as deep feed-forward neural networks and classifiers are stochastic gradient descent. The authors performed the experiment on the SigComp2011 dataset and concluded that the balance between error reduction and accuracy improvement is better than the case method [1].

• “Mohammad Hajizadeh Saffar et.al” have indicated that due to the lack of effective training models and the difficulty of not differentiating the models, a method can be used in which each user has a classification based on discrimination. First, a sparse autoencoder is trained first, using many anonymous signatures, and then the distinction is applied, which is learned by the autoencoder to represent education and experiences enrollment as a self-motivated learner. Finally, a single class is used to model and distribute user signatures. Although the experimental results show significant error reduction and improved accuracy on the SVC2004 and SUSIG datasets, the method is data signature independent.

• In this paper, the author “Jivvesh Poddar” proposed a new signature detection method based on convolutional neural network (CNN), peak-valley method and SURF algorithm, and Harris vertex detection algorithm verification based on new signature recognition and signature detection method, thus facilitating verification. This [9].

2.2 Goals And Objectives

The goal is to create a quality-based online signature verification model which provides us with the best accuracy.

Some common goals and objectives for this signature verification model are also :

1.Authenticity Verification: The primary goal of a signature verification model is to determine the authenticity of a signature, i.e., whether the signature being verified is genuine or not. The model should be able to distinguish between genuine signatures and forged or fraudulent signatures.

2.Accuracy: The model should achieve a high level of accuracy in correctly identifying genuine signatures and detecting forged signatures.

3.Efficiency: The model should be computationally efficient and capable of processing signature verification requests in a timely manner, particularly in real-time applications where quick decision-making is required.

2.3 Problem Defination

Handwritten signatures can be easily forged or manipulated, making it difficult to differentiate between genuine and fake signatures. It is very much impossible for a human to authenticate each sign of a person in such a quantity which even if someone will try to do will take a lot of time and will be very expensive. One of the main challenges in signature verification is the natural variation in handwriting, which can make distinguishing between genuine and forged or fake signatures difficult. Additionally, forgers can use advanced techniques to imitate a person's signature, making it even more challenging to detect fraud.

Moreover, with the increasing use of digital documents and digital signatures, there is a need for signature verification systems that can effectively verify both traditional and digital signatures.

Overall, the problem of signature verification arises from the need to distinguish between genuine and forged signatures while considering the natural variation accurately and efficiently in handwriting, advanced forgery techniques, and the need for remote and digital verification.

CHAPTER 3

<u>Design flow/Process</u>

3.1 Preliminary Work

1. Data Collection: Collect a dataset of signature samples for both genuine and forged signatures. It is important to have a diverse dataset that captures variations in writing styles, pen pressure, and other signature features.

2. Data Preprocessing: Preprocess the signature images to remove noise, normalize the size, and extract relevant features such as stroke thickness, curvature, and angle of inclination. This can be done using image processing techniques such as edge detection, binarization, and segmentation.

3. Feature Extraction: Use machine learning algorithms or other statistical techniques to extract a set of features from the preprocessed signature images. The features can be used to differentiate between genuine and forged signatures.

4. Model Training: Train a machine learning model on the extracted features using a labeled dataset of genuine and forged signatures. Popular machine learning models for signature verification include support vector machines (SVMs), random forests, and neural networks.

5. Model Validation: Validate the trained model using a separate dataset of signature samples. This is important to ensure that the model is accurate and reliable. If the model performs poorly, it may need to be adjusted or
retrained using additional data.

6. Model Deployment: Deploy the trained model in a production environment, such as a banking application or a document authentication system. The model can be integrated with other software components to provide a seamless user experience.

7. Model Maintenance: Maintain the model by periodically retraining it on new data and updating it to address emerging threats or challenges. This is important to ensure that the model remains accurate and effective over time.

I. Front-end:
The front-end will be developed using python language which provides the standard library Tkinter for creating the graphical user interface for desktop-based applications.

II. Back-end:
The backend is developed using the Machine learning algorithm
CNN(convolutional neural network) which is used to learn our model. MySQL, XAMPP will also be used or the User registration and login system of our model.

III. Database:
The system will use a database to store user data which will include basic information such as username, email, and password.

3.2 Requirements Of Preliminary Design

The Preliminary Design should provide sufficient confidence to proceed with the detailed design. It ensures that the basic system architecture is complete, that there is technical confidence the capability need can be satisfied within cost and schedule goals, and that risks have been identified and mitigation plans established. It also provides the acquisition community, end user, and other stakeholders with an opportunity to understand the trade studies conducted during the preliminary design, and thus confirm that design decisions are consistent with the user's performance and schedule needs and the validated Capability Development Document

A. **User Registration and Login:** The system should have a user registration and login feature where users can create their accounts and log in to access the features of the product.

B. **Uploading Image:** In this feature, there will be two sections provided to the user for uploading their image(signature). In the first section, the user will have to upload the original image which has to be verified with and in the second section, they have to upload the image they want to verify.

3.3 Flow Chart Diagram

Flowcharts are visual representations of a process or system that use symbols and arrows to illustrate the flow of information or actions. They are used to break down complex processes into smaller, more manageable steps and to communicate the steps and decision points in a clear and easyto-understand way. Flowcharts can be used in a variety of fields, including software development, engineering, business, and education. They are an effective tool for planning, analyzing, and optimizing processes, and can help identify areas for improvement or potential bottlenecks in a system.

- **Login Page**

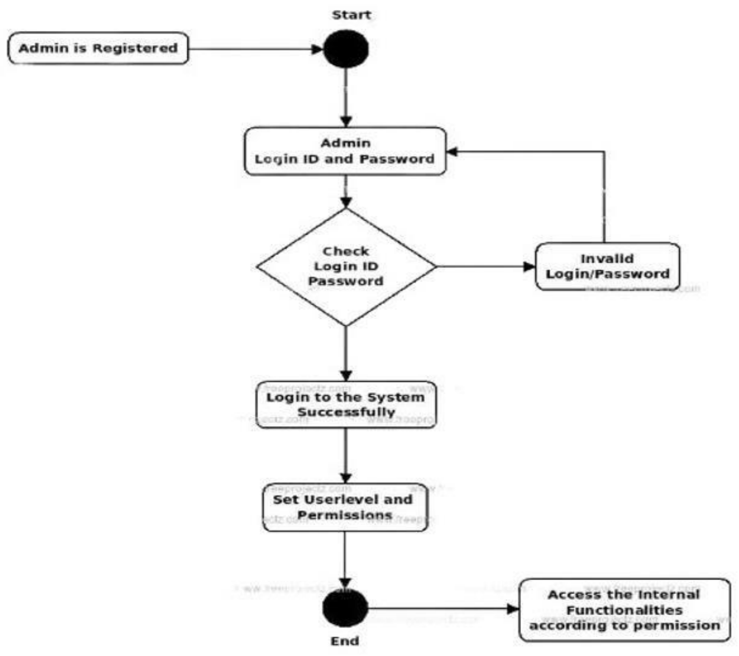


Fig 3.1(Login Page)

- **Data Training**

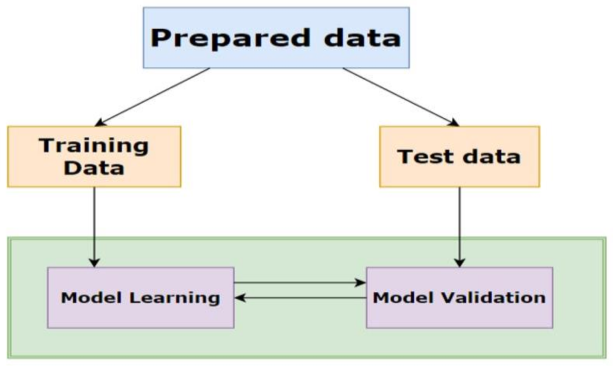


Fig 3.2(Data Training)

- **Model Working**

The below DFD shows the main components of the signature verification system and their interactions. The signature samples are collected and input into the system. The data preprocessing component applies image processing techniques to remove noise and extract relevant features from the signature images. The feature extraction component uses statistical techniques to extract a set of features from the preprocessed data. The feature selection component selects the most relevant features for the model. The model training component uses a labeled dataset of genuine and forged signatures to train a machine-learning model.

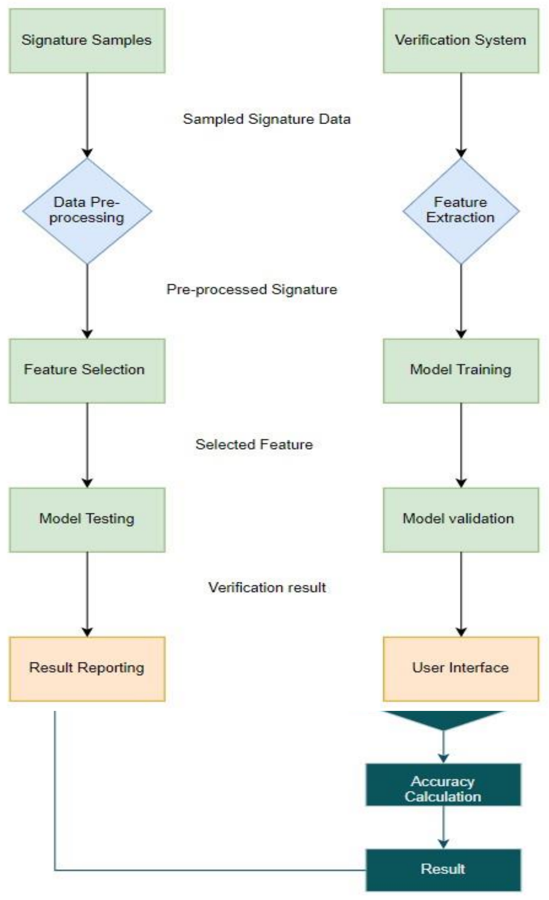


Fig 3.3(Model Working)

CHAPTER 4

USE OF MODERN TOOLS IN DESIGN AND ANALYSIS

4.1Introduction To Machine Learning

• Machine learning (ML) is a branch of artificial intelligence (AI) that enables computers to "self-learn" from training data and improve over time, without being explicitly programmed. Machine learning algorithms are able to detect patterns in data and learn from them, in order to make their own predictions. In short, machine learning algorithms and models learn through experience.

• In traditional programming, a computer engineer writes a series of directions that instruct a computer how to transform input data into a desired output. Instructions are mostly based on an IF-THEN structure: when certain conditions are met, the program executes a specific action.
• Machine learning, on the other hand, is an automated process that enables machines to solve problems with little or no human input, and take actions based on past observations.

• While artificial intelligence and machine learning are often used interchangeably, they are two different concepts. AI is the broader concept – machines making decisions, learning new skills, and solving problems in a similar way to humans – whereas machine learning is a subset of AI that enables intelligent systems to autonomously learn new things from data.

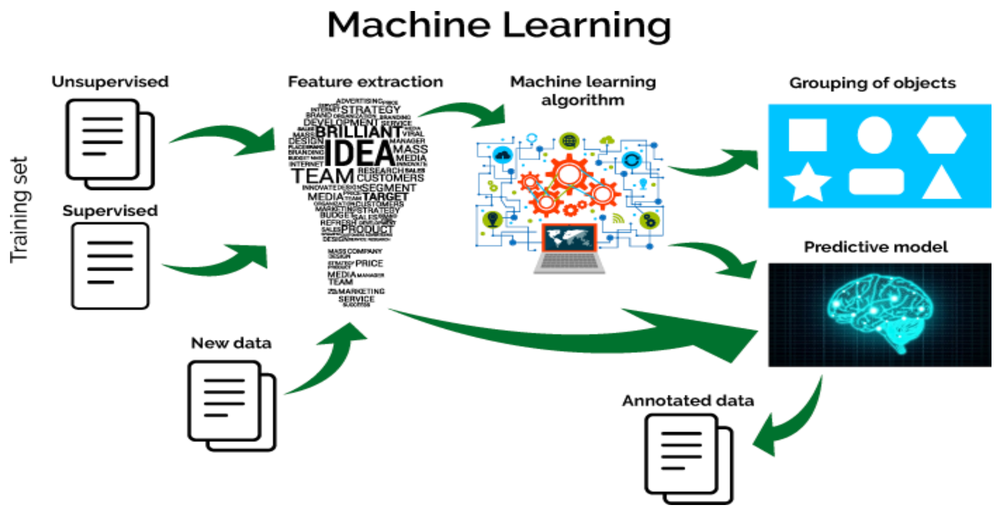


Fig 4.1(Machine Leaning)

4.2 Why Python For Machine Learning?

➢ **Python is easy to understand.**

• To reiterate, Machine Learning is simply recognizing patterns in your data to be able to make improvements and intelligent decisions on its own.

• Python is the most suitable programming language for this because it is easy to understand and you can read it for yourself.

• Its readability, non-complexity, and ability for fast prototyping make it a popular language among developers and programmers around the world.

➢ **Python comes with a large number of libraries.**

• Many of these inbuilt libraries are for Machine Learning and Artificial Intelligence, and can easily be applied out of the box.

• Some of the libraries are:

- scikit-learn for data mining, analysis, and Machine Learning;
- Tensorflow, a high-level neural network library;
- pylearn2 which is also ideal for data mining and Machine Learning, but more flexible than scikit-learn.

➢ Python allows easy and powerful implementation

• What makes Python one of the top choices for Machine Learning is its easy and powerful implementation.
• With other programming languages, coding beginners or students need to familiarize themselves with the language first before being able to use it for ML or AI.
• This is not the case with Python. Even if you only have basic knowledge of the Python language, you can already use if for Machine Learning because of the huge amount of libraries, resources, and tools available for you.
• Additionally, you will spend less time writing code and debugging errors on Python than on Java or C++.
• ML and AI programmers, in general, would rather spend their time building their algorithms and heuristics, rather than debugging their code for syntax errors.

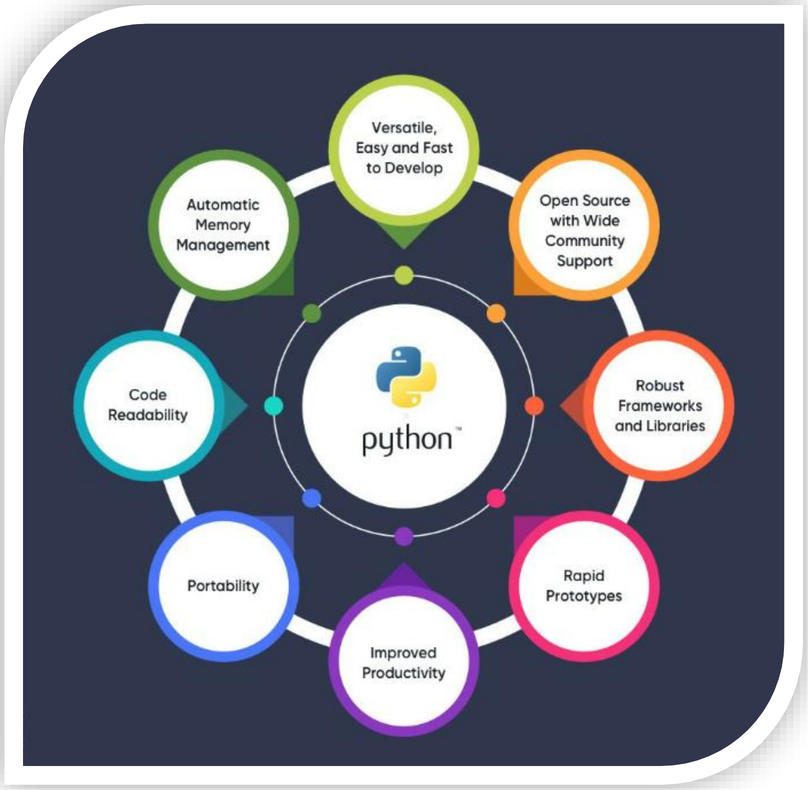

Versatile, Easy and Fast to Develop
Open Source with Wide Community Support
Robust Frameworks and Libraries
Rapid Prototypes
Improved Productivity
Portability
Code Readability
Automatic Memory Management
python™

Fig 4.2(Python Features)

4.3 MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

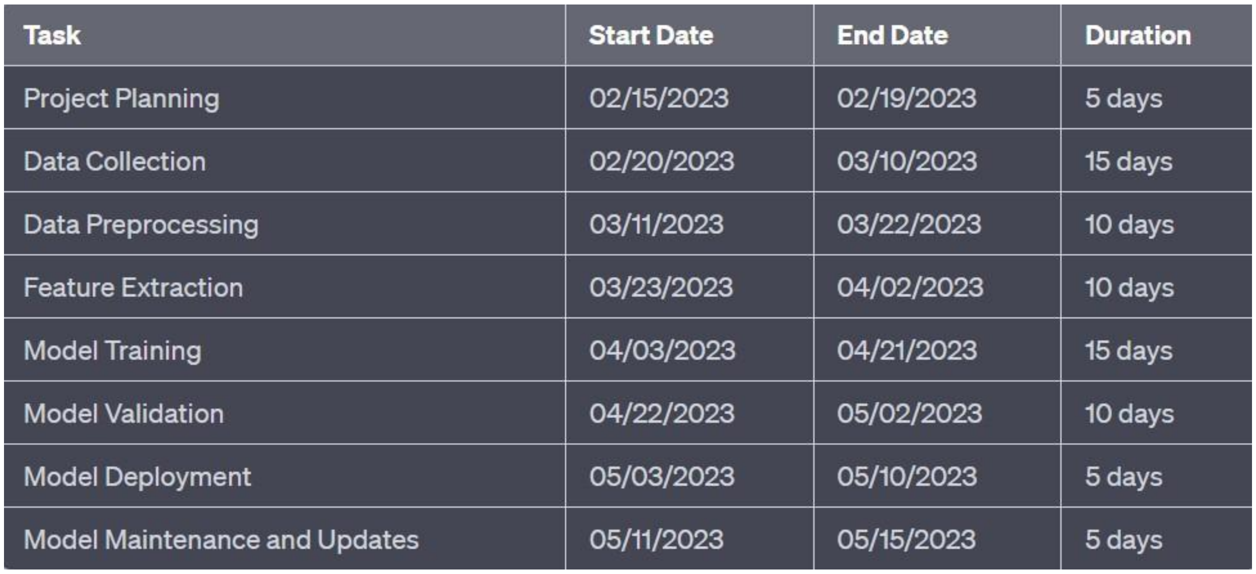
➢ Design

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

➢ Develop

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage standard database connections, including MySQL Fabric. The Object Browser provides instant access to database schema and objects.

4.4 Time Work

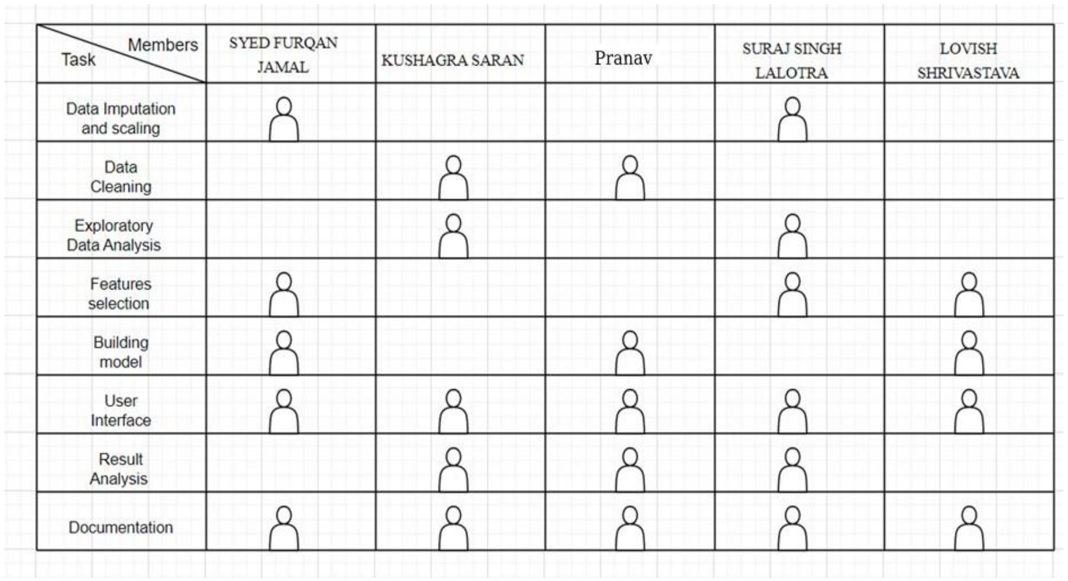


Task	Start Date	End Date	Duration
Project Planning	02/15/2023	02/19/2023	5 days
Data Collection	02/20/2023	03/10/2023	15 days
Data Preprocessing	03/11/2023	03/22/2023	10 days
Feature Extraction	03/23/2023	04/02/2023	10 days
Model Training	04/03/2023	04/21/2023	15 days
Model Validation	04/22/2023	05/02/2023	10 days
Model Deployment	05/03/2023	05/10/2023	5 days
Model Maintenance and Updates	05/11/2023	05/15/2023	5 days

Table 4.1(Time Work)

The above Gantt chart is just an example and the actual start and end dates, as well as the duration of each task, may vary based on the specific project requirements and resources available. The tasks listed are also high-level tasks and may need to be broken down into smaller sub-tasks or milestones for better project management.

4.5 Project Management



Task \ Members	SYED FURQAN JAMAL	KUSHAGRA SARAN	Pranav	SURAJ SINGH LALOTRA	LOVISH SHRIVASTAVA
Data Imputation and scaling	👤			👤	
Data Cleaning		👤	👤		
Exploratory Data Analysis		👤		👤	
Features selection	👤			👤	👤
Building model	👤		👤		👤
User Interface	👤	👤	👤	👤	👤
Result Analysis		👤	👤	👤	
Documentation	👤	👤	👤	👤	👤

Table 4.2(Project Management)

4.5 Testing

Testing signature verification systems plays a crucial role in ensuring their reliability and accuracy. Thorough testing helps assess the effectiveness of the system in distinguishing genuine signatures from fraudulent ones, thus providing an essential security measure in various applications such as financial transactions, legal documents, and identity verification.

When testing a signature verification system, several aspects need to be considered. Firstly, the dataset used for testing should be diverse and representative of real-world scenarios. It should include a wide range of signature samples, capturing variations in writing styles, sizes, orientations, and levels of complexity. This diversity ensures that the system can handle different signature types accurately.

The testing process involves comparing a given signature against a reference signature to determine their similarity. The choice of a similarity metric, such as the structural similarity index (SSIM), plays a vital role in evaluating the accuracy of the system. The metric should be sensitive enough to detect both subtle and significant differences between signatures while maintaining a reasonable threshold for false positives and false negatives.

To ensure comprehensive testing, various scenarios should be considered. These scenarios can include genuine signatures with slight variations, skilled forgeries that attempt to mimic the genuine signature, random scribbles that should be rejected, and deliberate attacks aimed at deceiving the system. By covering these scenarios, the system's robustness and accuracy can be thoroughly evaluated.

Furthermore, it is essential to measure the performance of the signature verification system in terms of efficiency and speed. The testing process should assess the system's response time, resource utilization, and scalability to handle large volumes of signature verification requests effectively.

To validate the system's performance, testing should be carried out using appropriate evaluation metrics and techniques. This includes metrics such as accuracy, precision, recall, and F1 score, which provide quantitative measures of the system's performance. Additionally, conducting user acceptance testing (UAT) involving real users can provide valuable insights into the system's usability and effectiveness in real-world scenarios.

Overall, testing signature verification systems is a critical step in ensuring their reliability and effectiveness. Thorough testing, utilizing diverse datasets, appropriate similarity metrics, and comprehensive scenarios, helps validate the system's accuracy, efficiency, and robustness. By continuously evaluating and refining the system through rigorous testing, organizations can enhance security, mitigate fraud risks, and build trust in their signature verification processes."

4.6 Design And Result

- **Login:**

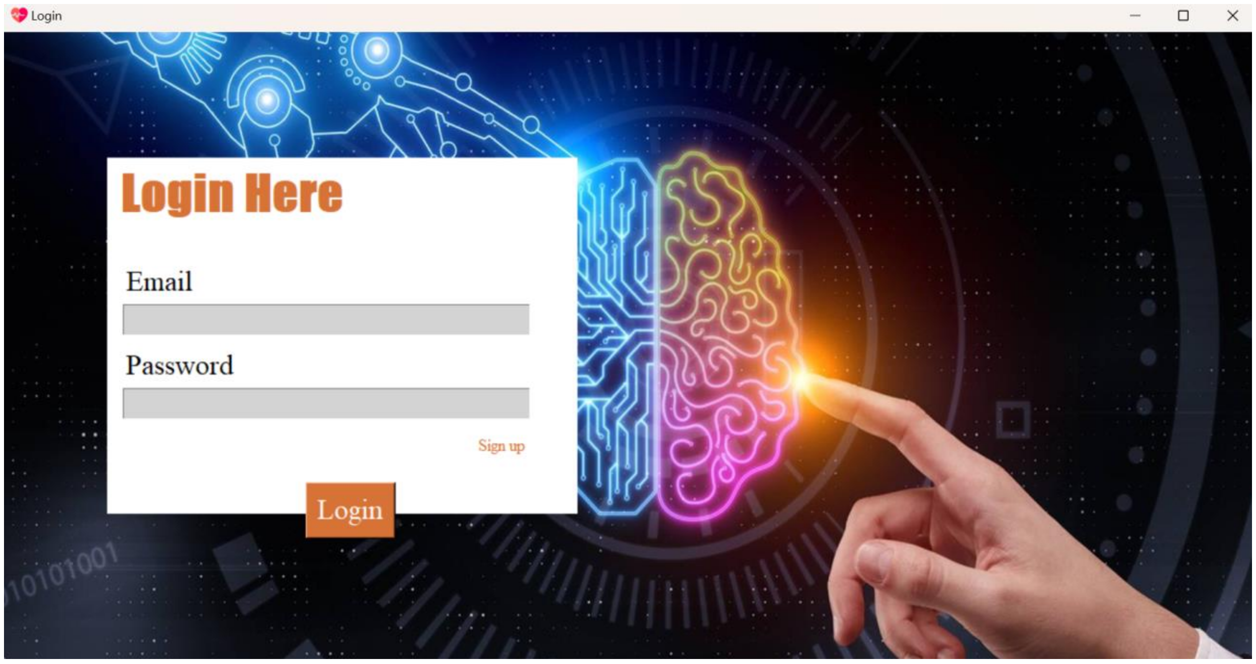


Fig. 4.3(Login Page)

- **Sign up**

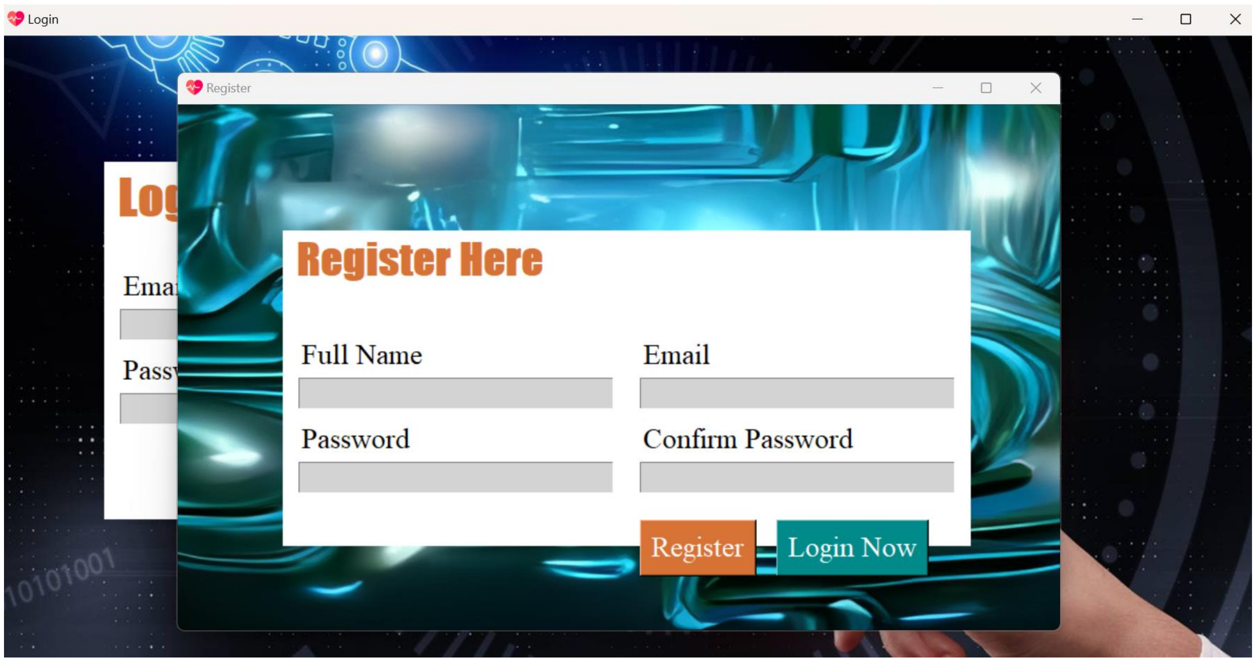


Fig. 4.4(Sign Up)

- **Main model**

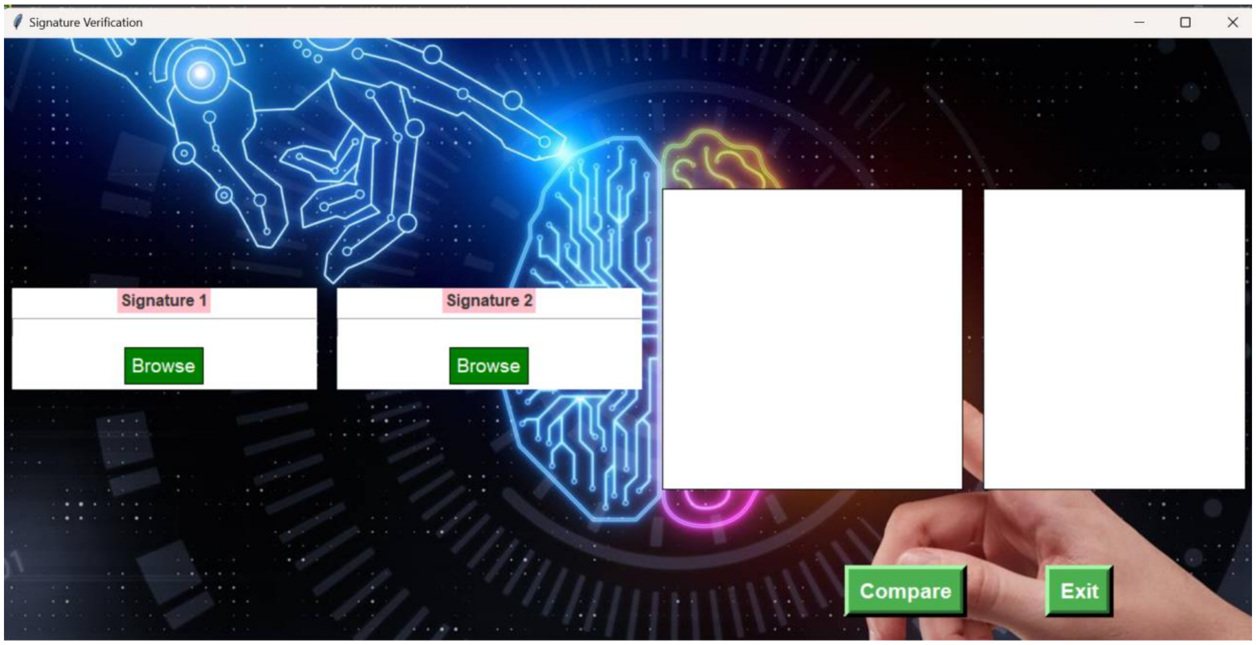
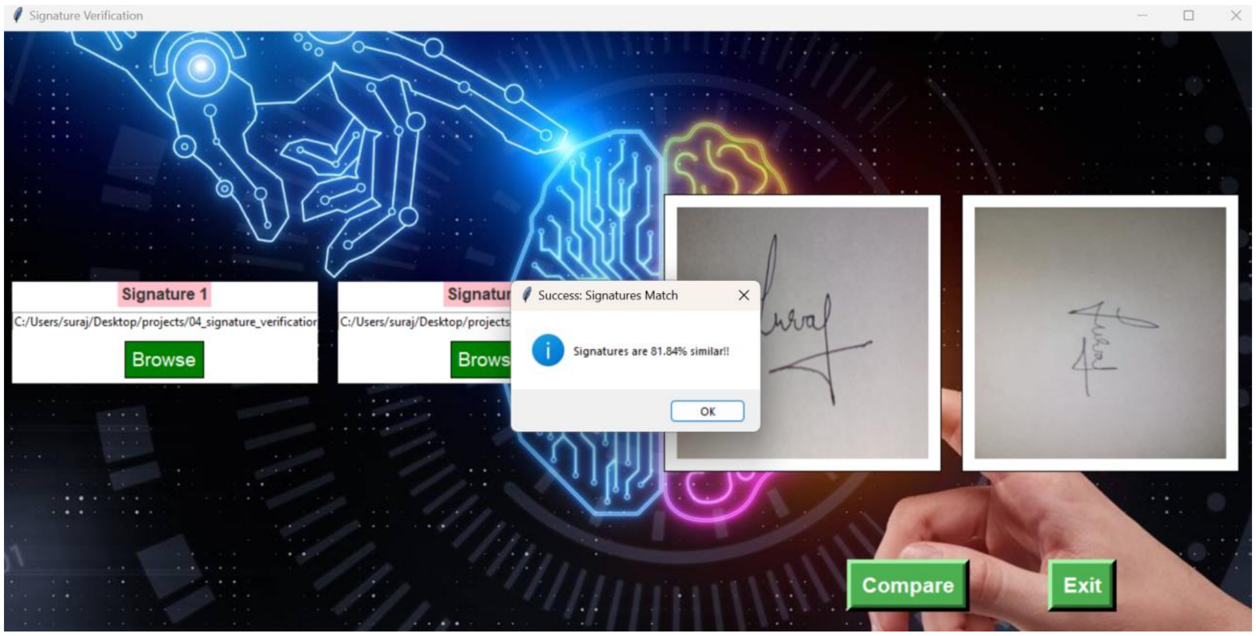


Fig. 4.5(Main Model)

- **Signatures Matched**


Signature Verification
Signature 1
C:/Users/suraj/Desktop/projects/04_signature_verification
Browse
Signatur
C:/Users/suraj/Desktop/projects
Brows
Compare
Exit

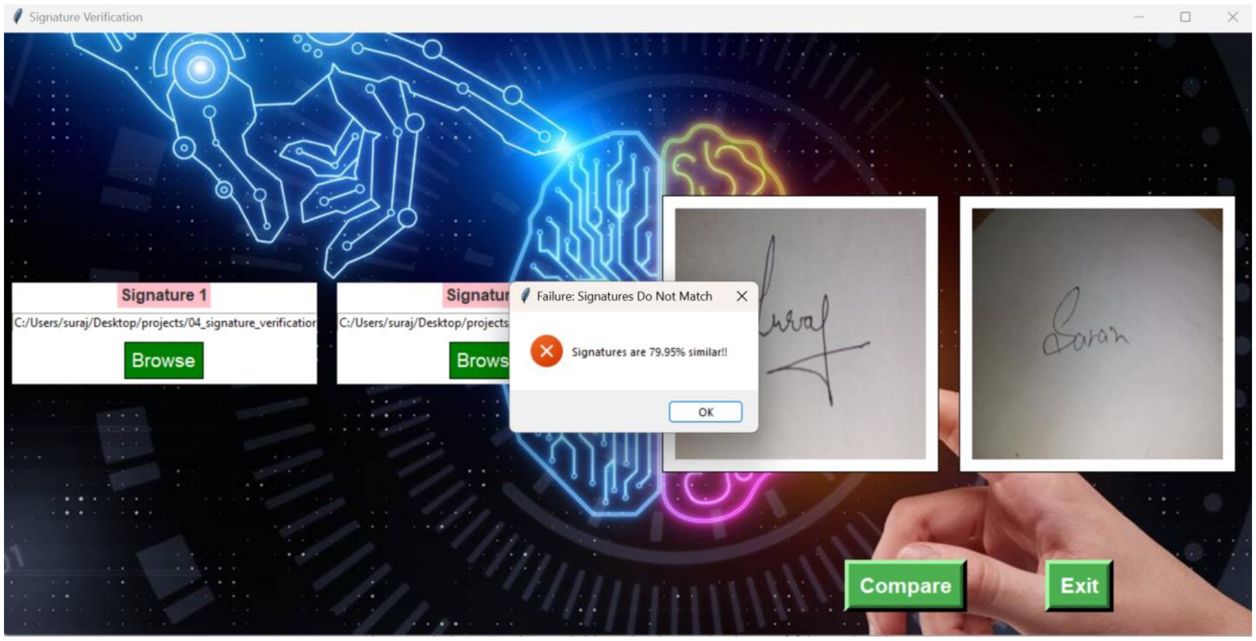
Success: Signatures Match

Signatures are 81.84% similar!!

OK

Fig. 4.6(Matched)

- **Signatures Do Not Match**


Signature Verification
Signature 1
C:/Users/suraj/Desktop/projects/04_signature_verification
Browse
Signatur
C:/Users/suraj/Desktop/projects
Brows
Compare
Exit

Failure: Signatures Do Not Match

Signatures are 79.95% similar!!

OK

Fig. 4.7(Not a Match)

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

- In this study, we developed a signature verification model using a convolutional neural network (CNN). The model is trained on a database of fake and real signatures and obtains an accurate difference between the two. We obtain reliable features for CNNs by preprocessing, resizing, converting to grayscale, and normalizing pixel values of signature images. The training model can accurately predict whether two signatures match, providing a reliable tool for the signature verification task.

5.2. Future Work

While our signature verification model using
CNNs show results, there are many avenues for future research and development:

I. Dataset Expansion: Large Collection, Many False and Real Records Documents can improve the performance of the model. .
A wider range of data will help the model learn more signatures and improve its overall capabilities.

II. Online Signature Verification: Extending the model to handle online signatures, where signatures are captured as a sequence of points or strokes, would be an interesting direction for future research. This involves designing appropriate preprocessing techniques and modifying the CNN architecture to handle sequential data.

References

[1] Amr Hefny and Mohamed N Moustafa, “Online Signature Verification Using Deep Learning and Feature Representation Using Legendre Polynomial Coefficients”, The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019), DOI: 10.1007/978-3-030-14118-9_68, January 2020.

[2] Mohammad Hajizadeh Saffar et.al., “Online Signature Verification using Deep Representation: A new Descriptor”, 24 Jun 2018

[3] Jivvesh poddar et.al., “Offline Signature Recognition and Forgery Detection using Deep Learning”, The 3rd International Conference on Emerging Data and Industry 4.0 (EDI40), Procedia Computer Science 170 (2020) 610–617, Warsaw, Poland, April 6 - 9, 2020.