An Efficient Mobile Handwritten Signature Verification System Based On Convolution Neural Netwoks (Cnn)

By

David Jonathan Omasete

91746

A Thesis Submitted To School of Computing and Engineering Science in Partial Fulfilment for The Award of The Degree of Master Science In Information System Security of Strathmore University

March, 2025

Declaration and Approval

Student Declaration

I, David Omasete of student ID 91746, hereby declare that this thesis titled "An Efficient Mobile Handwritten Signature Verification System Based on Convolution Neural Networks (CNN)" is my original work and to the best of my knowledge has not been presented for any academic award in this or any other institution. All sources of information used in this research have been duly acknowledged.

I confirm that this work adheres to the ethical guidelines and research integrity policies set forth by Strathmore University.

Signature: _____ Date: __ March 27, 2025_____

Supervisor Approval

I, Dr. Humphrey Njogu, confirm that I have supervised the thesis titled "An Efficient Mobile Handwritten Signature Verification System Based on Convolution Neural Networks (CNN)" conducted by Mr. David Omasete. To the best of my knowledge, this research meets the academic and ethical requirements of Strathmore University.

Signature: _____ Date: _____March 30, 2025___

Acknowledgement

First and foremost, I thank the Almighty God for granting me the strength, perseverance, and clarity of mind to undertake and complete this thesis.

I extend my deepest gratitude to my supervisor, Dr. Humphrey Njogu, for their invaluable guidance, encouragement, and support throughout this research. Your constructive feedback, timely reviews, and unwavering commitment were instrumental in shaping this work to its current form. Special thanks to the faculty and staff of School of Computing and Engineering Science and @Ilab Africa, particularly the committee members of SU-ISERC, for their insightful comments, ethical review, and approval, which significantly contributed to refining the scope and execution of the study.

I am sincerely grateful to all participants who willingly contributed their signature samples for this research. Your cooperation made the implementation and validation of the system possible. Special thanks also go to the various institutions in the Banking, Legal, Academia, Government, and Technology sectors that allowed their staff to take part in the study.

To my fellow students and colleagues, Mr. Bonface Musila, Mr. Kevin Obote and Code Guild and Miss Carol Wawira, thank you for the shared ideas, peer reviews, and moral support during challenging moments. A heartfelt appreciation to my family and loved ones for their continuous support, prayers, and understanding throughout this academic journey. Your belief in me kept me focused and motivated.

Finally, I acknowledge the authors, researchers, and developers whose works I referenced and built upon to implement the proposed solution. Their contributions to the fields of biometric authentication, machine learning, and computer vision provided a firm foundation for this project.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

VT OMNES
VNVM SINT

Abbreviations

API – Application Interface

CNTK – Cognitive Toolkit

CNN – Convolution Neural Networks

HMM – Hidden Markov Model

HSV – Handwritten Signature Verification

HTTPS – Hypertext Transfer Protocol Secure

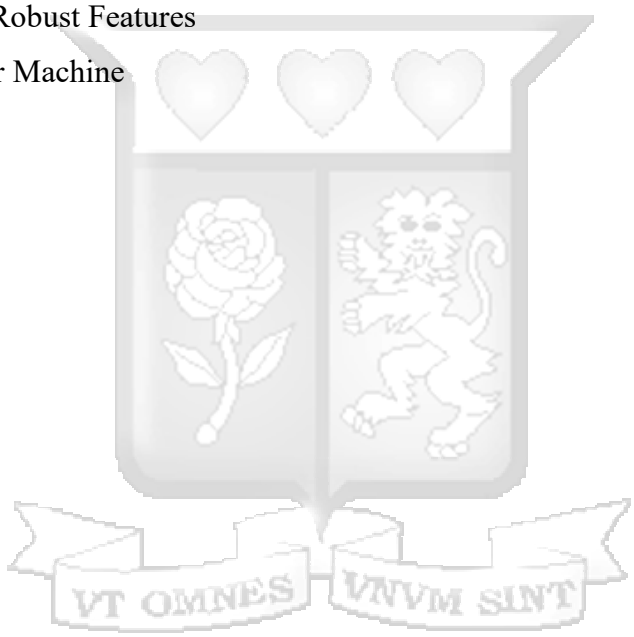
RNN – Recurrent Neural Networks

RGB – Red, Green, Blue

SITF – Scale-Invariant Feature Extraction

SURF – Speeded-Up Robust Features

SVM – Support Vector Machine



Definition of Terms

Signature Verification

The process of validating the authenticity of a handwritten signature, typically by comparing it to previously recorded samples. It is widely used in biometric security systems (Impedovo & Pirlo, 2008).

Convolutional Neural Networks (CNNs)

A class of deep neural networks that has proven effective in image recognition tasks due to its ability to learn spatial hierarchies of features directly from input images (LeCun et al., 2015).

Forgery Detection

The identification of signature samples that have been faked or copied in an attempt to deceive authentication systems (Ferrer et al., 2004).

Biometric Authentication

A security process that relies on the unique biological characteristics of individuals, such as fingerprints, voice, or signatures, to verify identity (Jain et al., 2000).

Abstract

Signatures are used as the main form of identification of the owner of the activity that is performing a transaction. A signature can only be accepted if it is from the authorized person. It is highly unlikely that two signatures created by the same person are identical. Many signature properties may vary even when the same individual signs two documents. As a result, detecting a forgery becomes a challenging task. Handwritten signature verification faces various challenges, including signature variability and the risk of forgery. Fraud through forgery is a common issue that drives corporates and businesses into significant financial loses and affects brand reputation in various sectors such as banking and government agencies which deal with important physical and online signed documents, legal paper works and government policies on daily basis Therefore, there is a need for a robust and accurate handwritten signature verification system that achieves the highest degree of accuracy in detecting fraud through forgery. The main objective of the research is to design, develop, test and validate the performance of an Efficient Mobile Handwritten Signature Verification System Based Convolution Neural Network, which can detect the authenticity of handwritten signatures with accuracy, precision and in real-time. The study is underpinned by theories such as Machine learning and Deep Learning Theory and Risk Adaptation Theory (RAT). The study utilized an Iterative Development Approach combined with Object-Oriented System Analysis and Design (OOSAD). The methodology involved incremental cycles of planning, analysis, design, implementation, testing, and evaluation. Convolutional Neural Networks (CNNs) was used for signature verification, utilizing preprocessing techniques like grayscale conversion, binarization, and noise removal. Signature datasets were sourced from Kaggle and AI-generated sources. The frontend was developed using Flutter, while the backend leverages Laravel API for secure communication. The model was trained and validated using performance metrics such as accuracy, precision, recall, and F1-score. Testing included system validation, user acceptance testing (UAT), and security checks. The study found that the CNN-based model coupled with a flutter-based mobile application achieved a 97.2% accuracy in distinguishing genuine from forged signatures, outperforming traditional methods. The system provides real-time verification, making it suitable for financial and legal applications. Additionally, strong security measures like encrypted data transmission and user authentication prevent unauthorized access.

Keywords:

Binarization; Image preprocessing; F1-score; precision; Recall; Forgery; fraud; signature verification; Mobile system

Chapter 1 Introduction

1.1 Background Information

Handwritten signatures serve as scripted names or any marks of legal identification mapped to individuals, executed by hand with the intention to authenticate in a permanent form the identity and ownership of the signing individual (Gideon et al., 2018). Handwritten signatures are one of the most prevalent techniques for confirming the identity of individuals in various sectors. Signatures have been used since 1677 and have formed important treaties, which led to the state of frauds to declare that any formal, contracts should be signed as it shows that the individuals were present at the said time of signing. The process of signing involves the movement of a pen or stylus on paper, giving rise to three key attributes: form, movement, and variation. With time, a signature's style may undergo slight changes, but its essential characteristics remain intact, making it a unique identifier (Gideon et al., 2018). The signing process can be understood as how the brain retrieves information from long-term memory, encompassing parameters like size, shape, and timing (Hafemann et al., 2017a).

Signatures used are always verified by comparing a reference signature stored either digitally or manually. Signature verification therefore is the process of comparing a signature to a reference in order to verify its validity. Forgery according to law is the making of a false writing with the intent of defrauding another party using the false writing. Signature verification and forgery detection are vital processes to determine whether a signature is genuine or forged (Amazon Web Services, n.d.). There are two main types of signature verification: static and dynamic. Static verification involves verifying an electronic or paper signature after it has been made, while dynamic verification occurs as an individual creates a signature on a digital tablet or similar device (Sharif et al., 2020). Forgery can be broadly categorized into three types:

i. Skilled forgery, crafted by professional impostors or individuals with extensive practice, can replicate actual signatures with high accuracy, making them challenging to detect visually (Hafemann et al., 2017b).
ii. Unskilled forgery occurs when the signer tries to imitate a signature without understanding its nuances or spelling.
iii. Random forgery, also known as blind forgery, is when the forger attempts to forge a signature without any knowledge of its appearance. This type of forgery is relatively easy to detect as it usually deviates significantly from the characteristics of a genuine signature.

Several research methods have been employed to tackle handwritten signature verification and forgery detection. Early approaches relied on statistical techniques and template matching, where the structural features of a signature—such as stroke patterns and projection profiles—were compared against a reference. While these methods provided a basic framework for verification, they often fell short in capturing the intrinsic variability of genuine signatures. Support Vector Machines (SVMs) have also been used to classify signatures by leveraging handcrafted features; however, they require extensive feature engineering and are sensitive to noise and variations in signature style. Additionally, Hidden Markov Models (HMMs) have been applied to capture the temporal dynamics of signature formation, but these approaches typically demand large training datasets and impose significant computational overhead.

Despite these advances, notable gaps persist in effectively addressing signature forgery, especially when confronted with skilled forgeries that mimic the subtle nuances of genuine signatures. Machine learning techniques, particularly Convolutional Neural Networks (CNNs), offer promising solutions to these challenges. CNNs automatically extract complex, hierarchical features directly from raw signature images, reducing the need for manual intervention and enhancing the system's ability to discern subtle differences. This automated feature learning significantly improves classification accuracy and robustness against noise. Moreover, CNN-based models are inherently scalable and well-suited for mobile platforms, facilitating real-time verification—a crucial requirement for practical deployment in financial, legal, and business environments.

According to amazon web services, neural networks are a type of machine learning process, also known as deep learning that uses interconnected nodes (computers) or neurons in a layered structure that resembles the human brain. Neural networks create an adaptive system that computers use to learn from their mistakes and improve continuously (Torres & Lima, 2023). In relation to this research, neural networks aid in sampling different signatures and compare them to determine their eligibility as either genuine or forged. Torres & Lima, 2023, in their research advancement in Biometric Signature Verification, no two signatures are 100 percent the same but has some features like size, gradient and other that are always be same. After training the datasets, the Neural Network created a pattern that was used detect when these unique aspects of any uploaded signature is not the same as the preliminary uploaded signature and give a forged status to the user, thus stopping an attempted forgery case.

The mobile handwritten signature verification system using CNNs can be applied across a diverse range of sectors. In the banking and financial services industry, it can authenticate signatures on cheques, loan documents, and financial transactions, significantly reducing fraud. In the legal sector, the system ensures the authenticity of contracts, deeds, and court filings, thereby bolstering

document integrity. Government agencies may also adopt this technology for secure identity verification and processing of official records. Additionally, the healthcare sector can leverage the solution to validate patient consents and secure medical records, while the insurance industry can use it to verify claims and policy documents. Other sectors such as real estate, corporate administration, and education can similarly benefit from enhanced document security and streamlined verification processes.

1.2 Problem Statement

The world in the 21st century is rapidly moving to the digital space and handwritten signatures remain a critical means of authenticating documents and transactions across diverse sectors such as banking, legal, government, Real estate, healthcare, corporate administration and education. The growth of the digital space has subsequently created new ways of causing fraud through forgery. Traditional automated signature verification systems — such as template matching, Support Vector Machines (SVMs), and Hidden Markov Models (HMMs) — have been created to help combat signature forgery. However, they heavily rely on manual feature engineering and struggle with variability in handwriting, noise, and sophisticated forgery techniques. Furthermore, while traditional methods have shown potential in controlled environments, their performance often degrades in dynamic, real-time digital environments where rapid and scalable verification is essential. The continued reliance on traditional verification systems may lead to an increase in successful forgery incidents, exposing organizations to significant financial, legal, and security risks. Additionally, the inability to adapt to dynamic digital environments could result in frequent misclassifications, undermining user trust in automated authentication processes. Ultimately, this gap could erode the integrity of digital transactions and critical document approvals across sectors, leading to broader systemic vulnerabilities in our increasingly digital society.

1.3 Objectives

1.3.1 General Objective

The general objective of this study is to design, develop and validate a mobile handwritten signature verification system based on Convolution Neural Networks (CNNs)

1.3.2 Specific Objectives

i. To understand the nature of frauds affecting the signature verification systems

ii. To review and critically analyze existing automated signature verification methods and identify their limitations

iii. To design, develop and test a CNN-based mobile handwritten signature verification system

iv. To validate the effectiveness of the proposed system in addressing signature forgery

1.4 Research Questions

i. What are the predominant types of fraud techniques that compromise signature verification systems?

ii. How do existing automated signature verification methods perform in terms of accuracy and security, and what specific limitations hinder their effectiveness against sophisticated forgery techniques?

iii. How can a convolutional neural network (CNN)-based mobile system be designed and optimized to accurately verify handwritten signatures?

iv. To what extent does the proposed CNN-based mobile signature verification system improve forgery detection compared to existing methods, and which performance metrics best demonstrate its effectiveness?

1.5 Justification

This study was rooted in the critical need to enhance security and efficiency in document authentication in our increasingly digital and mobile-centric world. Handwritten signatures continue to serve as a primary means of identity verification across sectors such as banking, legal, government, and healthcare. Furthermore, the rapid rise of mobile applications and real-time transaction systems has created a demand for verification methods that are both swift and highly reliable (Hernandez & Wang, 2019). Traditional methods—such as template matching, SVMs, and HMMs—often struggle with manual feature engineering, susceptibility to noise, and difficulties in accurately distinguishing genuine signatures from skilled forgeries.

Moreover, the increasing occurrences of signature forgery presents significant challenges, leading to potential financial losses and undermining trust in critical processes. The application of Convolutional Neural Networks (CNNs) offers a promising solution by automating the feature

extraction process and learning complex, hierarchical representations directly from raw signature images (Hernandez & Wang, 2019). This not only enhanced the accuracy and robustness of the verification process but also enabled real-time processing capabilities essential for mobile environments.

Therefore, the study proposed the development of an Efficient Mobile Handwritten Signature Verification System based on Convolutional Neural Networks (CNNs). By automating feature extraction and learning hierarchical representations directly from raw signature images, the CNN-based approach aimed to overcome the limitations of traditional methods, providing higher accuracy and robustness in detecting forgeries. The study sought to bridge the gap between existing verification technologies and the modern demands for rapid, secure, and scalable mobile verification solutions, thereby enhancing the integrity of digital transactions and document approvals.

1.6 Scope and Limitations

1.6.1 Scope

The primary scope of the study was to develop an Efficient Mobile Handwritten Signature Verification system utilizing the power of advanced technology (mobile and neural networks). The system was designed to analyze and compare handwritten signature files stored within schemas in the database, effectively detecting any indications of forgery. To accomplish this, the study needed various compatible software components, including cutting-edge programming frameworks for Neural Networks, proficient database management systems, and a user interface development environment.

1.6.2 Limitations

While the research aims to deliver an efficient and robust signature verification system, certain limitations are worth considering:

i. The performance of the mobile based CNN model is heavily dependent on the quality and diversity of the training dataset. Limited or biased signature samples may adversely affect the system's ability to generalize across various handwriting styles and detect sophisticated forgeries

ii. Deep learning models, particularly CNNs, require substantial computational power and time for training. This could pose challenges in optimizing the balance between training duration and achieving high accuracy, especially when scaling the model for large datasets.

iii. With limited or less varied training data, there is a risk of the CNN model overfitting to the training dataset, thereby reducing its performance on unseen data in real-world scenarios.

iv. Although the study proposes a mobile solution, deploying resource-intensive CNN models on mobile devices may encounter hardware limitations, affecting real-time performance and scalability.

However, through repetitive testing, optimization, and innovative solutions, the research aims to mitigate these limitations and deliver a state-of-the-art signature verification system that provides enhanced security and peace of mind for individuals and organizations alike.

Chapter 2 Literature Review

2.1 Introduction

Handwritten signature has been used for a long time to show ownership of a file. The significance of handwritten signatures and how to secure the signature from forgery has been an issue that has led researchers to have extensive discussions and white papers written to aid in developing the best security features of detecting signature fraud. Over the years, some techniques have been used in detecting fraud but with the uptake of technology in digital space, they have become less and less effective in detecting fraud and forgery. This revelation led to the innovation of technological tools to help curb the issue in terms of signature verification systems. Various technologies have been developed to enhance signature fraud detection. The literature review delves into the journey of signature detection and the developed signature verification system and also indicates the downsides of the existing systems that led to this research being done (Pokharel, & Giri, 2017). It looks at the intricacies of the different signature verification and pattern matching system and how neural networks can be used with machine learning to allow systems to self-learn the core components of a signature to improve on the detection on forged signatures in any documents.

2.2 Theoretical Literature

2.2.1 Risk Adaptation Theory

The Risk Adaptation Theory (RAT) is a behavioral theory that explains how individuals and systems adjust their risk perception and security measures based on exposure to evolving threats. Originally applied in domains such as cybersecurity, finance, and health sciences, RAT suggests that as risk levels fluctuate, individuals and organizations modify their protective strategies to maintain an acceptable balance between security and usability. In fraud detection and cybersecurity, RAT posits that as forgery and fraudulent activities become more sophisticated, security mechanisms must continuously evolve to counteract emerging threats.

In the context of handwritten signature verification, RAT is directly applicable as it underscores the need for adaptive security mechanisms to counteract advanced forgery techniques. Traditional signature verification systems, such as template matching and Support Vector Machines (SVMs), struggle to adapt to evolving fraudulent strategies, especially in mobile

environments. This study leverages Convolutional Neural Networks (CNNs) to introduce an adaptive and self-learning approach to signature verification, addressing RAT's core principle that security systems should dynamically respond to changing fraud patterns. By training on real-world signature datasets, the proposed system enhances its ability to detect increasingly sophisticated forgeries, thus embodying RAT's premise of continuous risk assessment and adaptation.

2.2.2 Machine Learning Theory

Machine learning theory provides the foundational framework for understanding how algorithms learn from data. It encompasses key paradigms such as supervised, unsupervised, and reinforcement learning, each with its mathematical models and optimization techniques designed to minimize error and improve generalization (Geron, 2019). Concepts such as the bias-variance tradeoff, VC dimension, and sample complexity underpin the theoretical limits and potential of these algorithms, helping researchers to better understand performance variations and to design more robust systems.

2.2.2.1 Supervised Machine Learning

Supervised learning in signature verification relies on training models with labeled data where genuine and forged signatures are explicitly marked (Geron, 2019). For example, convolutional neural networks (CNNs) are frequently employed in this context to learn discriminative features from static images of handwritten signatures. A typical system might use a dataset of genuine and forged signatures to train the CNN, enabling it to effectively classify new signatures based on learned patterns such as stroke formation, pressure distribution, and overall shape.

2.2.2.2 Unsupervised Machine Learning

Unsupervised learning, on the other hand, is used to uncover hidden structures in signature data without relying on labeled examples. Techniques such as clustering or anomaly detection can group similar signatures together or identify deviations from an individual's typical signing behavior (Chollet, 2021). For instance, an unsupervised system might cluster a set of signature images to detect unusual variations that could indicate forgeries. This method is particularly useful in the exploratory phase of system development or in situations where acquiring a fully labeled dataset is challenging, offering insights that can later inform supervised model training.

2.2.2.3 Reinforcement Learning

Reinforcement learning introduces an adaptive framework that can dynamically adjust the verification process based on feedback from the environment. Although less common in signature verification, reinforcement learning can be applied to fine-tune decision thresholds or update system policies in response to the evolving nature of forgery techniques (Sutton & Barto, 2018). For example, a signature verification system might deploy an agent that receives rewards for correctly classifying signatures, thereby gradually optimizing its criteria to reduce both false acceptances and false rejections. This continuous learning approach helps maintain the system's robustness over time, adapting to new patterns of fraudulent behavior.

2.2.3 Deep Learning Theory

Deep learning theory builds on the foundations of machine learning by focusing on the behavior and architecture of artificial neural networks. It investigates how multi-layered networks—such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs)—can learn hierarchical representations of data through backpropagation and gradient-based optimization methods. This area of study not only explores the mathematical substructures of network convergence and overfitting but also addresses practical challenges in scaling and model interpretability, providing a theoretical basis for advances in complex tasks such as image and speech recognition (Sutton & Barto, 2018).

2.3 Empirical Literature

2.3.1 Pattern Recognition Method

Pattern Recognition Theory is a cognitive and computational framework that explains how humans and machines identify, classify, and differentiate between visual patterns. It suggests that learning-based systems can recognize features, relationships, and structures within data to make informed decisions. This theory has been extensively applied in artificial intelligence, machine learning, and biometric authentication, particularly in tasks requiring the differentiation of complex visual inputs.

The effectiveness of a handwritten signature verification system depends on its ability to distinguish between genuine and forged signatures accurately. Traditional methods rely on predefined rules and handcrafted features, often resulting in high error rates when confronted with new or complex forgeries. This study integrates Pattern Recognition Theory through the application of CNNs, which automatically extract, learn, and classify unique signature features without human intervention. By leveraging convolutional layers, the system effectively

captures essential signature traits such as stroke dynamics, curvature, and pressure distribution. This approach enhances the system's robustness and adaptability, aligning with Pattern Recognition Theory's emphasis on feature extraction and classification accuracy in biometric systems.

2.3.2 Biometric Authentication Method

Biometric Authentication Theory is rooted in the concept that unique biological or behavioral characteristics can serve as reliable identifiers for individuals. This theory underpins security systems that use fingerprint recognition, facial recognition, iris scanning, and handwritten signature verification. The fundamental principle is that biometric traits are inherently unique to each individual and can be used for secure identity verification.

Handwritten signatures have long been considered a biometric identifier, though traditional authentication methods have often struggled with variability in signature replication. This study advances Biometric Authentication Theory by enhancing signature verification through deep learning models. The CNN-based approach ensures that signatures are analyzed with high precision, accounting for natural variations while detecting inconsistencies indicative of forgery. Unlike older methods that rely on static feature matching, this study employs dynamic learning, where the system continuously improves its verification accuracy by analyzing new signature data. The integration of mobile technology further extends the applicability of biometric authentication, allowing real-time, secure signature verification for financial, legal, and governmental transactions.

2.3.3 Hand Written Signatures

Hand written signature have been used sing the roman times and have had significant impact over a large pool of organizations around the world. Hands written signatures are used in nearly all the industries from national identification cards, bank accounts, voting rights in the United States among many other areas (Williams & Scott, 2020). . The signature is always given as an authentic confirmation that the individuals were present when the transaction was being made. Signatures also form a strong case for the authenticity of the person signing the document. Organizations such as banks rely a lot on signature to verify authenticity and having a way of easily verifying the legitimacy of the signature would be an added advantage in making workstream processes easy.

Signature evolved from the Roman Empire as a form of legally binding collaboration in the future. In 1677 the Parliament of England enacted the law in statute of fraud, which required all legal documents to be signed including, property transactions, wills, taxes in order to avoid fraud in the court (Sanders & Brown, 2021). This was as a result of the high number of court cases contesting the legitimacy of the documents being presented in court.

Hand written signatures were expected to be unique to individuals and as a result could show the legality of the matter. This method of verifying legitimacy worked well until the introduction of E-signatures was introduced as a result of the high number of forgery and fraud being done through falsified signatures in the different signatures. E-signatures also have an aspect of individuality and ownership that can be forged to foster fraudulent activities.

2.3.4 Forgery and Fraud Detection

Signature forgery has been deemed illegal under all circumstances. Forgery is the attempt of deceiving others by falsely representing someone's authorization or consent with the intent to defraud. This is majorly done in documents dealing with land and property and also in banks on cheques. A signature forgery is the attempt to create or recreate a signature. The forging of signatures leads to fraud cases that need to be investigated. This leads to the creation of fraud detection, which is the process that detects scams and prevents fraudsters from obtaining money or property through falsified means like forgery. Fraud detection is a set of activities taken to detect and block any attempt from malicious fraudsters to falsely obtain money or property. This is prevalent in the banking sector, insurance, health sector, public sector and in the government. According to Kanade 2021, fraud detection works in different stages including alert creation, analysis of customer approval patterns and identity relationships.

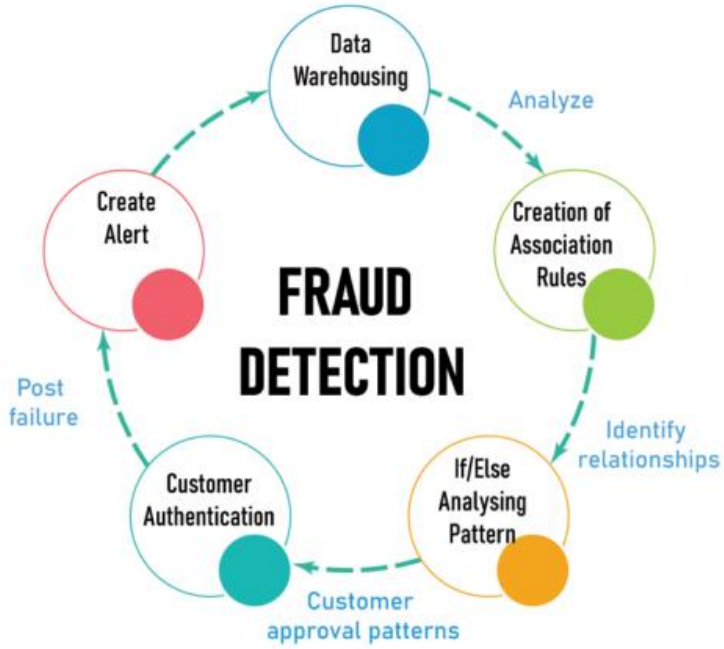


Figure2.1 Fraud Detection

(Source: Kanade, 2021)

2.3.5 Signature Verification

Signature verification has been used if different fields using different methods to try and get the best results. According to Abdulhusein et. al 2023, in their paper improving Arabic signature authentication with quantum inspired evolutionary feature selection, signature verification and identification systems have been utilized for the purpose of recognizing individuals. It has now become a significant undertaking to validate the legitimacy of forensic and commercial transactions. The precision of signature authentication relies heavily on the extraction and selection of distinctive features. In terms of feature selection, a reduction in recognition rate is observed when less discriminatory features are present using both global and local approaches. Consequently, stable and appropriate features play a vital role in the process of verifying signatures. To address this issue, a genetic algorithm (GA) was implemented to identify the most optimal features, proving to be efficient in optimization and machine learning applications (Pokharel, & Giri, 2017). Nevertheless, as the search dimension increases, GA encounters certain limitations such as insufficient exploitation and challenges in achieving satisfactory convergence rates. This leads to exponential growth in computation time and complicates finding an adequate solution.

A new approach put forth by their study involves a Quantum-Inspired Genetic Algorithm (QIGA) FS strategy that incorporates elements from both GA and Quantum Rotation

Computing (QRC) to enhance population diversity while alleviating issues related to computational complexity and premature convergence. By introducing superposition into both mutation (divergence) and crossover (convergence) processes, diversity among populations is increased, addressing concerns regarding population selection methodologies. To evaluate this novel method, three well-known public signature databases – SID Arabic handwriting signatures, CEDAR, and UTSIG – were employed for validation purposes. The findings from experiments conducted show an improvement in performance compared to classical genetic algorithms due to the incorporation of quantum-inspired computing principles within the proposed model.

2.4 Fraud Detection Technologies

Fraud through forgery is a prevalent issue in society today and is impact can cause huge losses to the affected party. Advancements in technology have created avenues for the development of fraud detection systems to combat the issue (Zhu et al., 2021). Fraud detection systems use machine learning, data analytics and anomaly detection in real time to detect fraud through forgery. These systems also use advanced algorithms and models, such as neural networks, to analyze data and identify patterns that can indicate fraudulent behavior. According to "Ad Fraud Taxonomy and Prevention Mechanisms" (Zhu et al., 2021), these systems have proven to be successful in detecting fraud attempts, saving companies millions of dollars each year. They not only prevent financial losses but also protect consumers' personal information from being compromised by cybercriminals. Additionally, these systems continuously evolve and adapt to new methods used by fraudsters, making them an effective defense mechanism for businesses of all sizes (Pokharel, & Giri, 2017). As we continue to rely more on technology for everyday tasks, the importance of fraud detection systems cannot be overstated

2.4.1 Neural Networks

There has been a surge of neural networks in the recent past, with different models being developed. Neural networks are interconnected computer nodes that have been modelled after the human brain (Sharif, Shah, & Khan, 2020). They function to process information and create complex patterns and algorithms based on the information provide to them. They are better that machine learning algorithms that take in information and compute truths based on the information give. This leads to false positives if the training information is not factually correct.

The ability of neural networks to make complex decision and recognize patterns make it paramount in the development of a signature verification system.

Artificial Neural Networks (ANN) consist of layers of nodes communicating with each other forming a complex pathway for data to flow through and in turn uses the information to learn patterns and create decision from the patterns that it has learnt from the data. The numerous layers of nodes also help neural networks to adapt and improve their performance as mode data is added to the training module giving it a vast field of pattern for it to recognize and improve. The speed with which neural networks process information makes it the best aspect to use for image and speech recognition.

Convolution Neural Network is a type of artificial neural network that is used primarily for image recognition and processing due to its ability to recognize patterns in images, which is a huge component in signature verification. The Convolutional Neural Network (CNN) is a powerful architecture designed to leverage the 2D structure of input data, such as images or speech signals (Patel, & Desai, 2020). It consists of convolutional layers, which utilize local connections and tied weights to extract meaningful features from the input data. The incorporation of pooling operations in CNNs leads to translation-invariant features, making them adept at handling various types of image-based tasks (Pokharel & Giri, 2017). A notable advantage of CNNs is their ability to be trained efficiently with fewer parameters compared to fully connected networks, while achieving impressive performance.

2.4.2 Preprocessing Techniques in Signature Verification

Signatures by nature have unique features that are developed innate by the consciousness of the person signing it because if the number of times they have been doing it (Pokharel, & Giri, 2017). The unique features in the handwritten signatures include gradient, size, style, legibility and pressure. These aspects give rise to the use of preprocessing techniques to analyse them in the signature verification system. Pre-processing is the first step in signature matching process where the input signature is simplified to match with referenced one (Saleem & Kovari, 2020). The aim of preprocessing is to enhance the captured signatures in order to obtain the same type of feature information, like position or velocity, and thereby improve the accuracy of the system (Saleem & Kovari, 2020). Correct preprocessing of signatures leads to a better result for both signature matching and forgery detection.

In general, signature image preprocessing tasks involve binarization, thinning, noise removal, bounding box, segmentation, inversion, and normalization activities. However, binarization is

the most adopted preprocessing technique. The binarization process is found very beneficial to remove the complexity in the signature image like segregating the signature pattern from white. In the binarization preprocessing technique, researchers have adopted several threshold methods to eliminate inconsistencies in the signature image (Hameed et al., 2021).

Thinning is another preprocessing method that is frequently used in signature verification. Line thickness variations are a common issue in signature images (Pansare & Bhatia, 2012). Thinning addresses this issue by removing the thickness differences and by making the signature image one pixel thick. The thinning process is important for the skeletonization of the image. Skeletonization is a process to provide region-based shape features by transforming the signature object in the signature image into a set of lines that run roughly down the centre of the signature object (Pansare & Bhatia, 2012). Pre-processing is a crucial step aimed at enhancing the quality of the signature images before feeding them into the signature verification system.

2.4.2.1 Converting RGB to Grayscale

Initially, the signature images are in RGB format, where each pixel is represented by three colour channels: red, green, and blue. To simplify further processing and reduce computational complexity, the images are converted to grayscale. Each pixel is represented by a single intensity value ranging from 0 (black) to 255 (white). The conversion is achieved by taking the average value of the RGB channels for each pixel. This process results in grayscale images, where the intensity value represents the overall brightness or luminance of the original pixel.

2.4.2.2 Converting Grayscale to Binary

Following the grayscale conversion, the 'greybin' function is employed to convert the grayscale images into binary form. In binary images, each pixel is represented as either black (0) or white (1) based on a specified threshold value. This step further simplifies the image representation and facilitates feature extraction during the subsequent stages of the system.

2.4.2.3 Noise Removal

To enhance the quality of the images and eliminate small components of noise, the Gaussian filter is applied to the grayscale images. The Gaussian filter is a smoothing technique that reduces image noise, ensuring a cleaner representation of the signature patterns.

2.4.2.4 Image Resizing

The 'preproc' function is applied to estimate the required boundaries for the signature part in each image. Resizing the images to a consistent and appropriate size aids in maintaining uniformity and allows for efficient processing during the verification phase. By implementing these pre-processing steps, the signature images are standardized and optimized, setting the stage for accurate feature extraction and subsequent signature verification.

2.5 Feature Extraction in Signature Verification

A key component of signature verification is Feature Extraction (FE), which involves taking relevant and distinctive aspects of a signature from its picture representation (Huang, & Yan, 1997). These characteristics enable accurate comparison and matching between various signatures because they capture the unique patterns and characteristics of the signature (Huang & Yan, 1997).

During the feature extraction phase, the system analyses a given pattern and records certain features, to yield structured data in the form of an observation sequence. The popular feature extraction algorithms include global features such as shape-based features and local features SITF (Scale-Invariant Feature Transform), SURF (Speeded-Up Robust Features), and local binary patterns.

2.6 Existing Signature Verification Systems

Signature verification systems have been developed over time to try and curb a certain aspect of the unique features of the signature. Some empirical research has been extensively done and proposed aiming to provide a comprehensive understanding of the methodologies applied. According to Diaz et al 2019, handwritten signature are biometric traits and over the past 40 years, interest has been growing steadily having its main reference in the application of automatic signature verification system. There are different systems that have been developed to aid in system verification that form a good background on creating the self-learning system that can be utilized in any industry that runs signature verification options (Kumar, & Sharma, 2022).

2.6.1 SVM-Based Systems with Crest-Through and Surf Algorithms

Support Vector Machines (SVMs) have been a popular method for signature verification, particularly in systems where handcrafted features are available. SVMs excel in high-

dimensional spaces and work well with a clear margin of separation between classes. In several studies, SVMs have been applied to classify signature images based on features extracted through various preprocessing techniques. The approach benefits from well-defined decision boundaries and kernel methods that allow it to handle non-linearities in the data, making it a reliable tool for binary classification tasks such as genuine versus forged signatures.

In parallel, feature extraction methods like Crest-through and Speeded Up Robust Features (SURF) have been utilized to capture key signature characteristics. SURF algorithms, known for their speed and invariance to scale and rotation, extract local keypoints that serve as robust descriptors of a signature's unique features. Crest-through methods, though less widely documented, focus on extracting structural and contour-based features, which can be particularly useful in capturing the overall shape and stroke flow of handwritten signatures.

Several studies have demonstrated the efficacy of Support Vector Machines (SVMs) in the domain of signature verification by leveraging well-engineered features for classification. For instance, Ferrer et al. (2008) showed that when combined with robust feature extraction techniques, SVMs can effectively discriminate between genuine and forged signatures, achieving competitive accuracy rates. Their work highlighted the benefits of SVMs in handling complex, high-dimensional feature spaces, a characteristic often present in signature data.

Additionally, a study by Eskander et al. (2012) developed an offline signature verification system that utilizes SVMs in conjunction with local descriptors to classify genuine and forged signatures, where handcrafted statistical and structural features were used to capture the intricacies of signature patterns. Their research demonstrated that, with proper feature engineering and parameter tuning, SVM-based systems could provide reliable verification outcomes even in the presence of significant variability in handwriting styles and noise. This study underscored the importance of the feature extraction process as a precursor to effective SVM classification.

Abdul Aziz et al. (2016) further validated the use of SVMs by integrating a combination of texture and shape-based features to enhance signature verification performance. Their work provided empirical evidence that SVMs, when applied with a well-selected feature set, could yield high accuracy in differentiating between authentic and forged signatures. Collectively, these studies emphasize that while SVMs are powerful classifiers, their success in signature verification heavily depends on the quality and representativeness of the manually engineered features.

While SVM-based systems, when combined with efficient feature extraction techniques like Crest-through and SURF, can provide a computationally efficient and interpretable framework, they also exhibit notable weaknesses. Their overall performance is contingent upon the quality of the manually engineered features, and they tend to struggle in the presence of high variability and noise. Moreover, as forgery techniques become more sophisticated, the handcrafted features may not capture all necessary discriminative details, leading to reduced accuracy. These approaches also require extensive domain knowledge for feature selection and tuning, which may limit their scalability and adaptability in rapidly evolving digital environments.

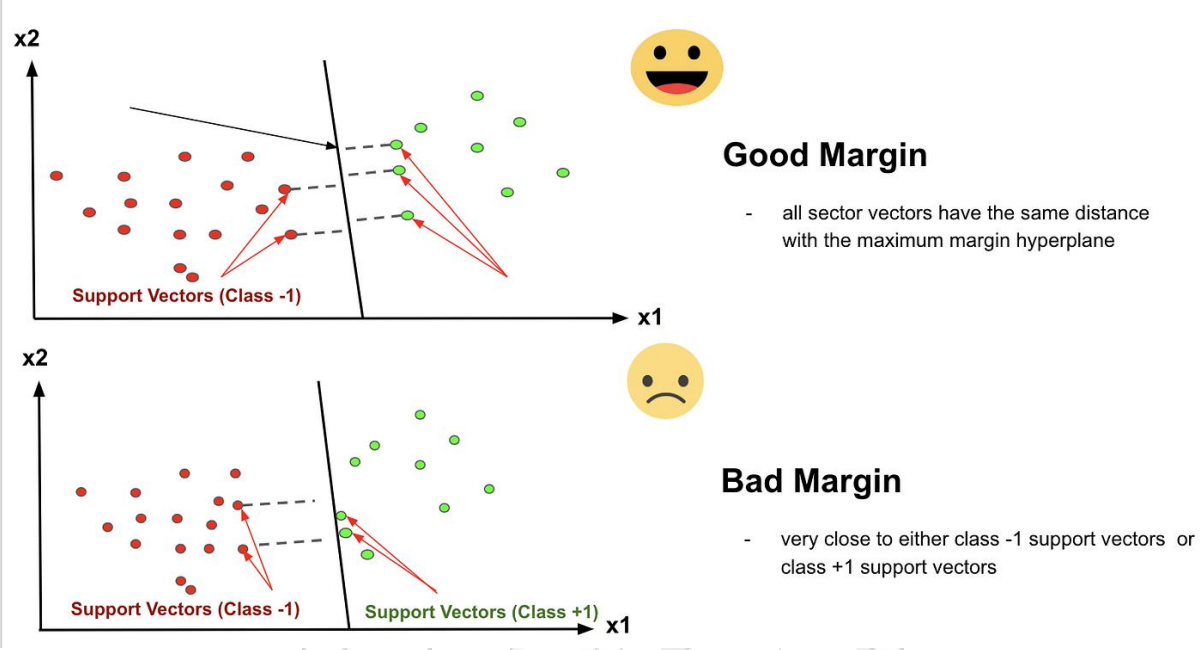


Figure2.2 Support Vector Machine Model

(Source: Saini, 2024)

2.6.2 Template Matching Approach

Several systems have implemented template matching approaches for signature verification, particularly in earlier or resource-constrained applications. For instance, early offline verification systems often used pixel-based correlation or distance measures to compare a test signature against a stored template. One notable example is the system reviewed by Ferrer, Morales, and Travieso (2008), which discussed how template matching techniques—through methods such as dynamic time warping (DTW) and simple correlation metrics—could effectively verify signatures when the variability in an individual's signature was minimal.

In addition to DTW-based systems, other implementations have used feature-based template matching, where signatures are first preprocessed to extract edge maps or skeletons before performing the matching operation. These systems typically align the signature to a reference template and then calculate a similarity score based on a chosen distance metric, such as Euclidean distance or correlation coefficient. While these methods provide computational efficiency and ease of implementation, they tend to be less robust against intra-class variations and sophisticated forgery attempts.

Overall, although template matching systems laid the groundwork for automated signature verification, their limitations—especially in handling significant variability and noise—have led to the evolution of more advanced methods. Nonetheless, the simplicity and efficiency of template matching make it a valuable approach in specific scenarios, such as low-resource environments or preliminary verification stages where rapid processing is paramount (Ferrer, Morales, & Travieso, 2008).

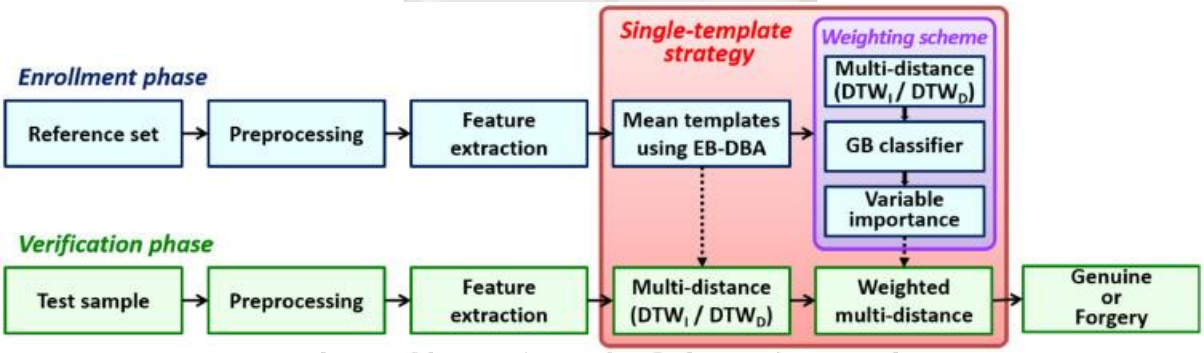


Figure2.3 Single template Matching Model

(Source: Saini, 2024)

2.6.3 Hidden Markov Model-Based Systems

The Hidden Markov Model (HMM) is a widely utilized model for sequence analysis in signature verification. In this approach, each point in a handwritten signature is represented as a series of vectors of values. HMMs are stochastic models that effectively capture the variability between patterns and their similarities (Akhila et al, 2021). The HMM stochastic matching involves comparing the probability distribution of features present in the signatures or the probability of how the original signature is generated. During the matching process, the model calculates the probability of a test signature being generated by the same process as the original signature. If the results indicate a higher probability than the test signature's probability, then the signatures are considered to belong to the original person. Conversely, if

the results show a lower probability, the signatures are rejected as potential forgeries (Akhila et al., 2021).

Hidden Markov Models (HMMs) have long been applied in the domain of online signature verification, primarily due to their strength in modeling temporal and sequential data. Pioneering work by Impedovo and Pirlo, (2008) demonstrated that HMMs could effectively capture the dynamic aspects of a signature—such as stroke order, pen pressure, and speed variations—by treating the signature as a series of state transitions governed by probabilistic rules. This probabilistic framework allows the system to accommodate uncertainties inherent in the signing process.

One of the main advantages of HMM-based systems is their ability to model time-dependent features, making them particularly effective in scenarios where the dynamic nature of signature input is essential. They offer a structured approach to capturing variations over time and can be tuned to differentiate between typical signing behaviors and anomalous patterns indicative of forgery. The statistical nature of HMMs also lends itself to well-established methods for parameter estimation and sequence alignment, providing a solid theoretical basis for their operation.

Nevertheless, HMM-based systems face several limitations. Their performance is heavily reliant on manual feature extraction and preprocessing steps, which can be labor-intensive and may not capture all the nuances of signature dynamics. Noise in the input data—such as jitter or sensor inaccuracies—can adversely affect the reliability of the model. Furthermore, HMMs are primarily designed for sequential data and may struggle when applied to offline signature images where temporal information is absent, reducing their overall versatility in mixed environments. While HMMs offer a robust method for online signature verification with clear advantages in modeling temporal dynamics, their dependency on manual feature engineering and sensitivity to noise limits their effectiveness, especially in more challenging offline verification scenarios.

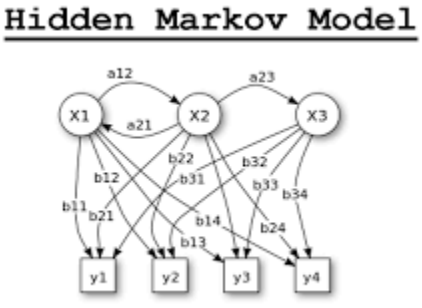


Figure2.4 Hidden Markov Approach

(Source: Wisdomml, 2023)

2.7 Research Gaps

The existing signature verification systems reveal several research gaps, primarily revolving around their reliance on manual feature extraction and sensitivity to variations in signature characteristics. For instance, SVM-based systems, while effective in high-dimensional spaces, depend heavily on handcrafted features that may not adequately capture the complexities of diverse handwriting styles and noise, thus limiting their adaptability in dynamic digital environments (Ferrer, Morales, & Travieso, 2008; Eskander et al., 2012). Similarly, HMM-based systems excel in modeling the temporal dynamics of signatures but struggle with offline data where such sequential information is absent, and they are prone to errors when faced with sensor noise or subtle forgeries (Impedovo & Pirlo, 2008; Akhila et al., 2021).

Additionally, template matching systems, although computationally efficient and straightforward to implement, tend to be overly sensitive to intra-class variations and minor distortions, resulting in high rates of false rejections or acceptances. These systems often fail to account for the sophisticated techniques used in signature forgery, thereby compromising their reliability when deployed in real-world scenarios where signature variability is a significant concern (Ferrer, Morales, & Travieso, 2008). This limitation highlights the need for systems that can robustly handle the intrinsic variability of handwritten signatures while maintaining computational efficiency.

The research gap poised lies in developing a more automated and robust signature verification system that overcomes the limitations of traditional methods. There is a clear need for solutions that can learn hierarchical and discriminative features directly from raw signature images—such as Machine learning-based systems—without relying on extensive manual feature engineering. Such an approach would enhance the system's accuracy and scalability in both online and offline environments, ensuring secure and reliable authentication across various digital platforms

2.8 Conceptual framework

The system begins with inputs that are collected through a mobile application. These inputs include user registration and login data, as well as multiple signature samples (e.g., Sig1, Sig2, Sig3, etc.) which are uploaded by users. This collection of genuine and forged signatures forms the dataset necessary for both training and testing the verification system

Once the data is collected, several processes are executed. The raw signature images undergo preprocessing steps to normalize and prepare them for analysis. Key features are then extracted using a Convolutional Neural Network (CNN) designed for the task. During the training phase, the CNN learns to identify and differentiate subtle differences between genuine and forged signatures. The system further evaluates its performance by applying scoring metrics such as F1, recall, and precision, ensuring that the model achieves reliable classification accuracy.

Finally, the outputs are generated based on the processed data. The output of the system is a classification decision that determines if a signature is genuine (typically indicated by a score equal to or above a set threshold, e.g., ≥ 0.5) or forged (with scores below the threshold). These results not only provide a verification decision for authentication but also offer insights into the system's performance through quantitative metrics, thereby completing the framework from input to final decision-making

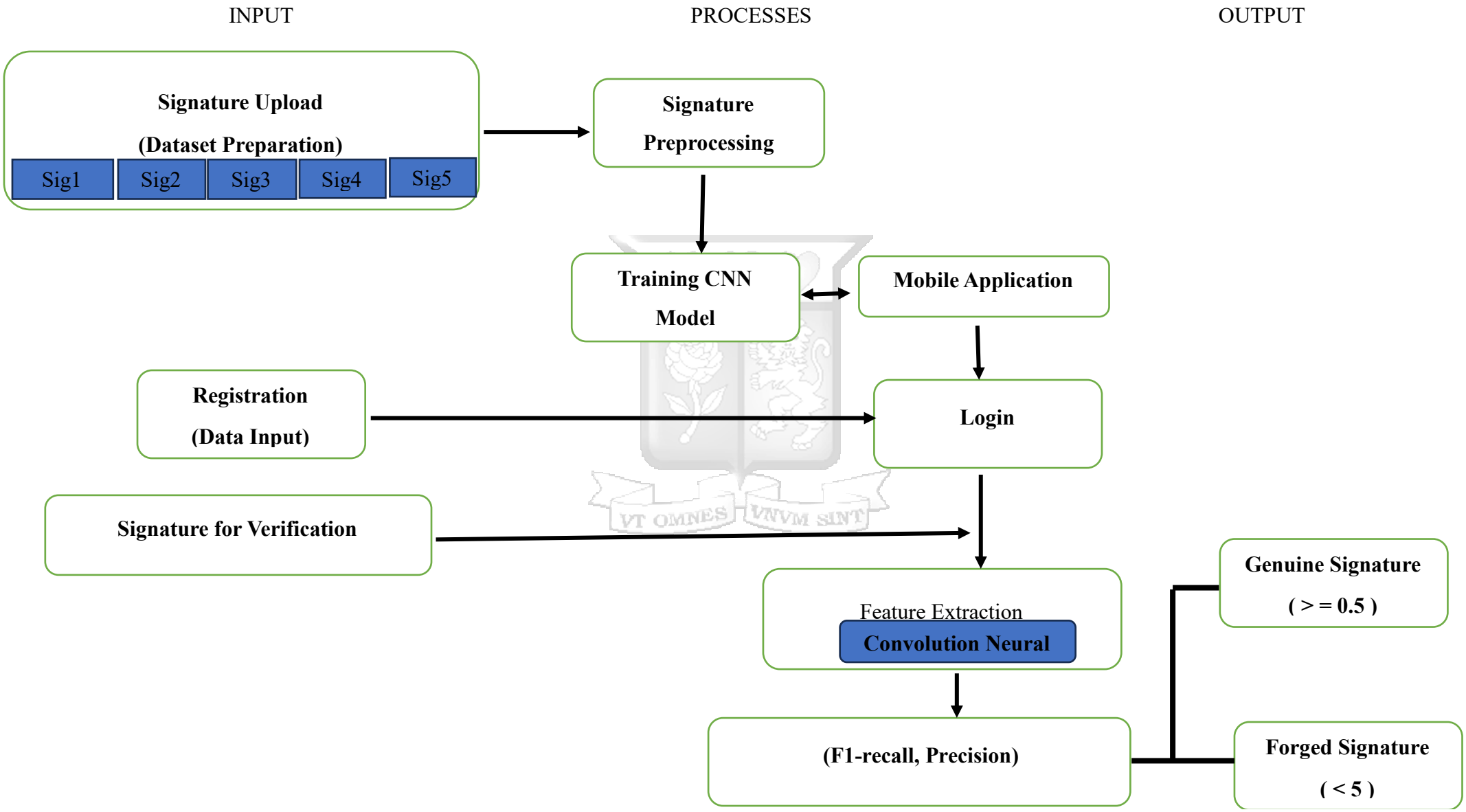


Figure2.5 Conceptual Framework

Chapter 3 Research Design and Methodology

3.1 Introduction

This chapter presents the methodology adopted to design and develop the signature verification system. It outlines the system requirements and the approach taken to meet these requirements effectively. Additionally, the system development methods and technologies utilized in the implementation process are discussed in detail. The chapter offers a comprehensive overview of the systematic approach employed to create a robust and efficient signature verification system using neural networks.

3.2 Research Design and System Development Approach

3.2.1 Methodology for achieving: Understanding the Nature of Fraud Affecting Signature Verification Systems

To achieve this objective, a systematic literature review was conducted. The methodology involves gathering academic journals, conference proceedings, and industry reports that document various forms of signature forgery and fraud. This review utilized databases such as IEEE Xplore, Scopus, and Google Scholar using key terms like “signature fraud,” “handwritten forgery,” and “biometric fraud.” The analysis focused on identifying common forgery techniques, factors influencing fraud, and emerging trends in signature tampering. Additionally, qualitative data was extracted from case studies and surveys in relevant sectors like banking, legal, government among others to provide a holistic understanding of the problem. This mixed-method approach ensured that both historical trends and recent developments were considered, forming a robust foundation for further research.

3.2.2 Methodology for achieving: Critical Analysis of Existing Automated Signature Verification Method

To achieve the second objective, a comparative systematic review methodology was adopted. The process included a thorough examination of existing signature verification systems that employ techniques such as Support Vector Machines (SVMs), Hidden Markov Models (HMMs), template matching, and feature extraction algorithms (Crest-through and SURF). The review incorporated inclusion and exclusion criteria to filter relevant studies, followed by an analytical framework to compare performance metrics such as accuracy, false acceptance/rejection rates, and robustness under variable conditions. In addition to reviewing empirical studies (Ferrer et al., 2008; Impedovo & Pirlo,

2008), the analysis addressed the reliance on manual feature engineering, scalability challenges, and sensitivity to noise, thus identifying gaps and limitations that persist in current methodologies.

3.2.3 Methodology for achieving: Design, Development and Testing of a CNN-Based Mobile Handwritten Signature Verification System

The Study utilized a well-considered iterative methodology employed as the fundamental approach to system development to achieve the third objective. The iterative methodology was chosen due to its adaptability and effectiveness in handling complex problems like signature fraud detection (Almeidaa, 2019). The iterative methodology involved breaking down the development process into multiple cycles or iterations, each building upon the knowledge and insights gained from the previous one. It allows the system to progressively learn, refine, and optimize its algorithms, leading to steady improvements in performance over time.

By adopting an iterative methodology, the study was able to identify and rectify the root causes of the initial poor performance. Each iteration presents an opportunity to fine-tune hyperparameters, adjust the neural network architecture, and explore various preprocessing techniques to enhance the system's ability to distinguish between genuine and forged signatures. Moreover, the iterative approach allows the study to incrementally expand the training dataset with a diverse range of signature samples. As more data is incorporated into the system, it becomes better equipped to handle variations in signatures, improve generalization, and achieve higher accuracy in detecting fraudulent signatures (Kumar, & Sharma, 2022). Additionally, the iterative methodology provides the flexibility to adapt to changing requirements and challenges. It facilitates continuous testing and validation, enabling the validation of the system's performance against new and unseen signature data.

The iterative model was used to break down large and complex projects to smaller tasks that can better be analyzed and tested for maximum output (Almeidaa, 2019). Signature verification has different steps that are tested in different forms which the iterative method would better help define.

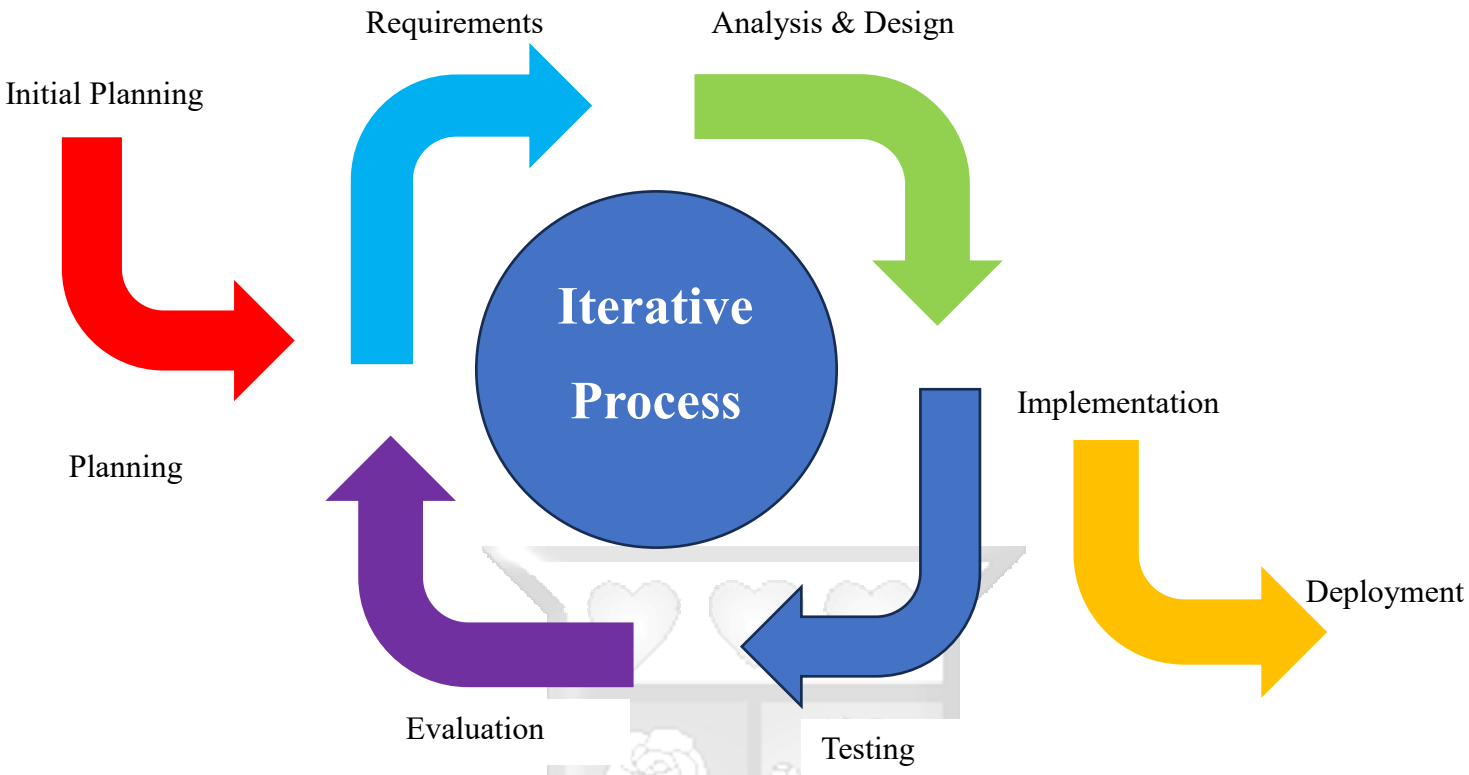


Figure3.1 Iterative Development Model

(Source: Almeidaa, 2023)

3.2.3.1 Initial Planning

Initial planning was the first step of the iterative method that allowed the development team to have the whole picture of the project and this was from the collection of the signatures, preprocessing, feature extraction, comparison, validation and alert triggering.

3.2.3.2 Planning

Here a number of signatures were collected from Kaggle. Kaggle is a widely used platform for data science and machine learning competitions, offers powerful cloud-based resources, such as GPUs and TPUs, for model training. It provided a convenient environment to access GPU resources, which were essential for training deep learning models like CNNs (Kumar, & Sharma, 2022).

3.2.3.3 Requirements

The main goal was to elaborate on all the tools and technical aspects that were needed to implement and design the neural network tool. In the study, a number of tools were incorporated together to develop the most user-friendly signature detection system. Among the tools required for the

implementation of the study was flutter, used for client-side application development that allowed the user to log in and upload/register the signatures to be evaluated. Kaggle was also used to fetch pre-uploaded signatures since signatures are sensitive to collect from individuals (Kumar & Sharma, 2022). For preprocessing the signatures from Kaggle, TensorFlow and Keras from Google were used to verify the signatures. A Convolutional Neural Network was used to compare genuine and forged signature images. The connection between the neural network and the application was implemented using API tunneling tools like laravel.

3.2.3.4 Evaluation

During the testing phase, the input signature was evaluated to determine if the output was ≥ 0.5, indicating that the signature was verified as genuine and legitimate, thus passing the test. Conversely, if the output was < 0.5, the signature was deemed a forgery. In the case of a genuine signature, 2-3 additional trials were conducted to verify consistency in the evaluation output. If inconsistencies were observed, the iterative process was repeated from the planning phase to refine and optimize the signature verification tool.

3.2.3.5 Deployment

Once the whole iterative process was implemented and the output from the check of any of the signature was verified to be the same after 3 trials, the tool was moved on to deployment for real testing in the stakeholders' sight to determine if it worked as was proposed in the documentation of the signature verification tool.

3.2.4 Methodology for achieving: Validation of the Proposed System's Effectiveness in Addressing Signature Forgery

To validate the effectiveness of the proposed CNN-based system in addressing signature forgery (objective four), a comprehensive quantitative evaluation framework was implemented. Initially, the system's performance was benchmarked against established baseline methods—such as SVM-based systems, HMM-based systems, and template matching approaches—using a standardized dataset that included both genuine and forged signatures. Performance metrics including accuracy, false acceptance rate (FAR), false rejection rate (FRR), precision, recall, and F1 score were computed to facilitate a rigorous comparative analysis. Statistical tests, such as paired t-tests and ANOVA, were employed to determine the significance of any performance improvements observed over traditional methods. In addition to controlled experiments, the system was deployed in a pilot mobile application environment to collect real-world operational data and user feedback, which assessed its usability, scalability, and responsiveness under dynamic conditions. Finally, the robustness of the system was

examined by testing it against various forgery techniques and noise levels, and a thorough security assessment was conducted to ensure data privacy and system integrity. This multifaceted validation approach was designed to confirm that the proposed system not only outperforms existing methods in controlled settings but also meets the practical demands of digital signature verification in real-world applications.

3.3 System analysis and system Design

Analysis and design was the third phase of the iterative model, and was used to check and ensure that all the tools and technical requirements were well placed in the hierarchy of development to ensure that all the requirements were met in a sequential order. This phase shifted from iteration to iteration depending on the output obtained from the previous iteration trial.

The study leveraged the Object-Oriented System Analysis and Design (OOSAD) methodology to develop the proposed solution. OOSAD involved analyzing the problem domain and creating a modular and scalable solution. With the support of the iterative development model, OOSAD ensured that all modules in the proposed handwritten signature verification system were incorporated into various models, including use cases, sequence diagrams, entity-relationship diagrams, wireframes, and the database schema designed to store the input signatures (Hafemann, Oliveira, & Justino, 2017). In this case, the iterative model ensured that processes were executed multiple times to achieve an accurate and optimal result for the neural network solution. The design phase was crucial in ensuring that all proposed aspects were systematically mapped, developed, and tested for full functionality, as outlined in the objectives in Chapter 1.

3.4 System Implementation

Once the analysis and design had been completed, the implementation phase began and aimed to produce the desired output from the study. In this phase, application development, signature retrieval, preprocessing, and feature extraction were carried out sequentially based on the design proposed in the previous phase.

The system was implemented on multiple levels using various implementation aspects. The neural network was first hosted on a machine with high specifications, a Core i7 processor with quad-core computing power and a clock speed of 3.0 GHz or higher. This specification was chosen to minimize latency and ensure that the scanned handwritten signature was processed in a timely manner.

On the software side, Python was used to facilitate neural network processing tasks. TensorFlow played a fundamental role in developing and deploying neural network models, while Laravel was utilized as the API manager for the signature verification system. To obtain training data, Kaggle was

used as a source of AI-generated signature datasets, considering the sensitivity and difficulty of collecting real signatures. For the client-facing application, Flutter was employed to develop the interface that allowed users to upload and scan signatures. Finally, a Convolutional Neural Network (CNN) was used to process and analyze the behavior of each signature, determining whether the scanned handwritten signature was genuine or forged (Kumar & Sharma, 2022).

3.5 System Testing

The system was tested using new signature samples, which were scanned and added to the system database. The first aspect of testing focused on whether the end-user application operated as expected. This was evaluated by determining whether a signature could be successfully uploaded to the device, even in the absence of an internet connection. The second aspect of testing assessed the ability of the signature verification tool to accurately identify signatures and generate at least three consistent output results for a specific signature. The primary goal at the end of the Study was to ensure that any change in a signature could be accurately detected by the verification system, thereby confirming its effectiveness in distinguishing between genuine and forged signatures.

3.5.1 Target Population and Sampling

The target population covered in the study comprised individuals who frequently use handwritten signatures across various sectors, including banking, asset management for court proceedings, and routine tasks such as signing employment documents. The sampled signatures reflected the diverse contexts in which handwritten signatures are utilized for verification and identity validation. However, due to the sensitive nature of signatures and the reluctance of individuals to provide them for experimental purposes, the study employed artificially generated signatures from kaggle. These AI-developed signatures closely mimick real-world signature samples from different sectors, ensuring that the study maintained authenticity while addressing ethical and privacy concerns.

3.5.1.1 Sampling Techniques and Sampling Size

To ensure a representative and relevant dataset for validating the Mobile Handwritten Signature Verification System, purposive sampling was adopted as the primary sampling technique. This non-probability method was chosen due to its effectiveness in selecting individuals who possess specific characteristics or expertise relevant to the study objectives—in this case, individuals who regularly authenticate documents and transactions using handwritten signatures. A total sample size of 50 participants was selected, distributed across five key sectors: Banking, Legal, Academia, Government, and Technology, with 10 participants drawn from each. This distribution ensured that the signature verification system would be tested against a wide range of professional signature patterns and forgery

risks inherent in these sectors. The purposive approach allowed for the inclusion of participants whose professional roles frequently involve document verification, thereby enhancing the relevance and applicability of the study outcomes. This detailed sampling strategy improves the system's external validity and ensures that the resulting model is robust and generalizable across diverse real-world scenarios.

3.5.2 Data Collection/ Retrieval

Signatures are a sensitive aspect of an individual's identity, making it challenging to collect samples due to concerns about forgery. As a result, the data collection process in this study involved obtaining signature images from an open-source repository on Kaggle. The dataset consisted of both genuine and forged signature images, structured into two main categories: "train" and "test."

In the "train" directory, subdirectories were labeled with unique numerical identifiers (e.g., 001, 002, 003), each representing a specific user. Within each user's directory, two subdirectories contained the genuine signatures (e.g., 001) and the corresponding forged signatures (e.g., 001_forg). The "test" directory followed the same structure, ensuring consistency for model evaluation. Each user directory contained 16 images—8 genuine signatures and 8 forged signatures—resulting in a total of 346 user directories and 5,472 signature images.

The signature images were primarily stored in PNG format to ensure lossless compression while retaining image quality, which was essential for machine learning tasks. PNG format was chosen for its ability to preserve critical signature details, enhancing the accuracy of the verification process.

For system validation, study subjects' signatures were used to demonstrate the final output of the system. These signatures were collected after participants provided informed consent. Each participant registered into the system and uploaded five samples of their legitimate signature and five samples of their forged signatures in real time, allowing for further assessment of the system's performance in detecting forgeries.

3.5.2.1 Inclusion and Exclusion Criteria

To ensure the relevance and integrity of the collected data, specific inclusion and exclusion criteria were defined for the participant selection process. Inclusion criteria required that participants be aged between 21 and 60 years, actively employed within one of the target sectors—Banking, Legal, Academia, Government, or Technology—and have routine signing responsibilities in their professional roles. This criterion was essential to focus on individuals whose signatures are frequently used for document validation, making them highly relevant for testing the accuracy and robustness of the signature verification system.

On the other hand, the exclusion criteria disqualified individuals who had known motor impairments or neurological conditions that could significantly affect their handwriting consistency. Participants who were unable or unwilling to provide informed consent were also excluded. These measures ensured that the study focused on realistic use cases of digital signature verification while maintaining ethical and data quality standards.

3.5.3 Data Analysis

The data analysis process focused on evaluating the effectiveness of the developed handwritten signature verification system using Convolutional Neural Networks (CNN). The dataset consisted of 5,472 signature images from an open-source repository on Kaggle, categorized into genuine and forged samples. These images were divided into training and testing sets, ensuring a balanced and diverse representation of different handwriting styles. Several preprocessing techniques were applied to enhance signature verification accuracy, including grayscale conversion, noise reduction and image binarization, normalization, and data augmentation. These steps improve image quality and facilitate better feature extraction. The CNN model, designed for this study, comprised of convolutional layers for pattern detection, pooling layers for dimensionality reduction, fully connected layers for classification, and a softmax activation function for authenticity determination.

The model was trained and tested using accuracy, precision, recall, and F1-score metrics to assess performance comprehensively. Accuracy measures the overall proportion of correctly classified signatures, providing a general performance indicator. Precision evaluates the proportion of true positives among predicted positives, ensuring that genuine signatures are not misclassified as forgeries. Recall, also known as sensitivity, determines the model's ability to correctly identify genuine signatures among all actual genuine cases. The F1-score balances precision and recall, offering a single metric that accounts for both false positives and false negatives, making it particularly useful when dealing with imbalanced datasets.

Experimental results demonstrate that the CNN model achieves high accuracy (>90%) in distinguishing between genuine and forged signatures, with reduced false positives and false negatives compared to traditional verification methods. A user-friendly interface was developed to facilitate practical implementation, allowing users to register and input real-time signatures for authentication. Real-time testing involved collecting five genuine and five forged samples per participant, demonstrating the model's effectiveness in practical applications. The system's high precision and recall rates confirm its potential use in financial and legal sectors for fraud detection, making it a viable solution for real-world deployment. Overall, the integration of CNN-based feature extraction,

preprocessing techniques, and real-time validation ensures a robust and efficient handwritten signature verification system

3.5.4 Benefit Distribution and Access to Results

The findings from this study will be shared equitably across all five participating sectors—Banking, Legal, Academia, Government, and Technology—through summary reports, presentations, and tailored feedback to individuals from the institutions involved. Participants will indirectly benefit through the enhanced security and efficiency of digital authentication systems informed by the research findings. Where applicable, institutions will receive technical recommendations on how to integrate machine learning-based signature verification technologies into their workflows. Additionally, the research outcomes will be published in accessible academic and industry-focused forums to promote knowledge dissemination and foster future innovation in secure digital verification systems.

3.5.4.1 Dissemination of Research Findings

To ensure transparency and accountability, participants will be kept informed of the research progress and findings through multiple channels. Upon conclusion of the study, all participants will receive a concise summary report via email, outlining the key insights derived from the research. In addition, stakeholder feedback sessions will be organized in collaboration with representatives from the involved sectors—Banking, Legal, Academia, Government, and Technology—to discuss implications of the results and gather sector-specific feedback. These sessions will facilitate knowledge exchange and promote adoption of the developed mobile handwritten signature verification system where appropriate. The broader findings will also be disseminated through academic publications, conferences, and institutional presentations to ensure wide accessibility and impact.

3.6 Research Quality

The study on An Efficient Handwritten Signature Mobile Verification System Based on CNN demonstrates a strong research quality through its rigorous methodology, comprehensive literature review, and well-defined experimental framework. The study effectively integrated deep learning, particularly Convolutional Neural Networks (CNN), to enhance signature verification accuracy using mobile environments as a platform, since most people in society use smart phones as literature shows. It critically analyzes existing signature verification techniques, identifying gaps like limited mobile specific approaches- where many previous studies focus on offline or desktop-based signature verification -, real-time processing efficiency and security risks such as adversarial attacks, relay attacks or image-based forgeries (Hafemann, Oliveira, & Justino, 2017). The dataset selection,

preprocessing steps, and model architecture are meticulously documented, ensuring reproducibility. The evaluation metrics, including accuracy, precision, recall, and F1-score, provide a robust assessment of model performance. Additionally, the thesis discusses the practical implications, challenges, and potential improvements, reflecting a thorough understanding of both theoretical and applied aspects of CNN-based biometric authentication.

3.7 Ethical Approval

This study is committed to upholding the highest ethical standards in line with research policies and institutional guidelines. To respect the rights and welfare of participants, informed consent will be obtained prior to participation, emphasizing the voluntary nature of the study and the right to withdraw at any time without consequence. Anonymity and confidentiality will be preserved through data pseudonymization and secure storage of all collected information. Special attention will be given to cultural sensitivity, ensuring that participants' customs, beliefs, and perceptions are respected during the data collection and reporting process. Furthermore, data will be used strictly for research purposes and will be securely destroyed after analysis to prevent misuse. All participants will be treated equitably regardless of their sector or role, reinforcing the integrity and inclusivity of the research.

For the ethical approval and considerations of the study, institutional approval from Strathmore University was required, as the study was conducted as part of an in-house research project for the completion of a degree course. The study was carried out at @iLab Africa, located in the Student Centre at Strathmore University. Institutional approval was necessary to certify that the study and its results adhered to Strathmore University's ethical standards. As an accredited institution, Strathmore University provided a certificate of ethical clearance to validate the research process (Strathmore University, n.d.).

The ethical aspects of the study primarily focused on the collection, usage, and disposal of signature data, given that signatures are sensitive personal identifiers. To ensure compliance with ethical considerations, the Study adhered to the following measures:

i. Utilization of pre-uploaded signature samples from Kaggle AI, thereby eliminating the need to collect personal signatures from individuals
ii. Informed consent was obtained from research subjects whose signatures were used for validation. Participants signed a consent form outlining how, when, and where their signatures would be used and how they would be disposed of after the validation process.
iii. Compliance with data protection laws in Kenya to ensure that any collected validation signatures were handled securely and ethically.

iv. Ethical certification was obtained from the Strathmore Ethical and Research Department to ensure the Study met the required ethical standards before implementation.

.

Chapter 4 System Analysis, Design and Architecture

4.1 Introduction

Systems Analysis, Design and Architecture defines the model and the tools that were used in the development of the signature verification system (Hafemann, Oliveira, & Justino, 2017). The process conforms to objective four of the study that indicates the development of the earlier proposed solution. This chapter contains the system analysis, both functional and non-functional requirements of the signature verification system, it also includes the systems design done using the object-oriented system and analysis model.

4.2 Requirement Gathering and System Analysis

The information (sample signatures) used to create the signature verification system was obtained from artificial generated signatures from Kaggle. This is due to the sensitive nature of handwritten signatures being personally identifiable information (PII) that has a security impact to the privacy of the people who share their signatures, in case of leak or breach during the project development phase (Saleem, & Kovari, 2020). The signatures collected by Kaggle (an artificial intelligence database) were used to give the signature verification system test data that could be used for data training. To ensure that the tool works in real world, three volunteer signatures were used to show the effectiveness and efficiency of the tool during deployment.

4.2.1 System Analysis

Systems analysis breaks down the functionality of the signature verification system. It is paramount to have the analysis to show what the system can do both in functional and non-functional requirements as this information would be used to ensure that the system does what it was intended and if it functions well to ensure efficiency and effectiveness.

4.2.1.1 Functional Analysis

The system has some qualities and characteristics that it MUST provide to the end user after development. These aspects form the functional requirements of the system. They include:

Table 4.1 Functional Requirements

Functional Requirements	Description

User Authentication	The system must provide a user authentication mechanism to ensure only authorized users can access the signature verification services. Users must log in with their credentials before using the system.
Signature Upload	Users must be able to upload images of signatures for verification. The system must support various image formats, including PNG, JPG, JPEG, and GIF.
Signature Pre-Processing	Upon receiving a signature image, the system must perform pre-processing steps, such as converting the image to grayscale, resizing, and noise removal, to enhance the image quality.
Signature Verification	The system must employ a Convolutional Neural Network (CNN) model to verify the authenticity of the uploaded signature. Verification results must be based on a binary output, indicating whether the signature is <0.5 or >=0.5.
Signature Classification	The system must be able to classify the signature images as either genuine or forged.
Upload of Reference Images	The system must be able to allow authorized users to add new reference signatures to the database.
Real-Time Prediction	The system must provide real-time predictions for signature verification, ensuring prompt responses to user requests.
Secure Communication	All communication between the client and server must be encrypted using HTTPS to ensure data privacy and security.

4.2.1.2 Non-functional Analysis

Non-functional requirements specify the characteristics and qualities that the signature verification system must possess, which do not directly involve specific functionalities but significantly impact its overall performance, usability, and user satisfaction. These requirements focus on aspects such as system performance, security, reliability, and user experience.

Table 4.2 Non-Functional Requirements

Non-Functional Requirement	Description
1. Usability	The user interface of the system should be intuitive and user-friendly, requiring minimal training for users to interact effectively. Clear and informative error messages should be provided to guide users in case of incorrect inputs.
2. Performance	The system should provide real-time responses for signature verification, with a maximum response time of 30 seconds.
3. Security	User data, including uploaded signature images and personal information, should be stored securely using encryption techniques. The system should employ robust user authentication mechanisms to prevent unauthorized access.
4. Reliability	The system should be highly reliable, with minimal downtime or service disruptions. It should have a system uptime of at least 95%.
5. Accuracy	The signature verification system should achieve a minimum accuracy rate of 90% in differentiating genuine signatures from forged ones.
6. Scalability	The system architecture should be designed to scale horizontally to accommodate an increasing number of users and image verification requests.
7. Compatibility	The system should be compatible with different operating systems (IoS and android) to cater to a wide range of users.
8. Maintainability	The codebase and system architecture should be well documented and organized to facilitate easy maintenance and future enhancements.
9. Resource Utilization	The system should utilize hardware resources efficiently to optimize performance and minimize resource wastage.

10. Concurrency	The system should be able to handle concurrent user requests without significant degradation in performance.

4.3 System Architecture

The System Architecture Diagram illustrates the overall structure and flow of data within the signature verification system (Jain, & Ross, 2022). It consists of five key components: the User, Frontend (User Interface - Mobile App), Backend Server (API & Logic), Database (Signature Data Storage), and the AI Model (Convolutional Neural Network for Signature Verification). The diagram shows how users interact with the frontend by uploading their signatures for verification. The frontend then communicates with the backend, which processes requests, retrieves and stores data in the database, and sends the signature images to the AI model for verification (Jain, & Ross, 2022). The AI model analyzes the signature and returns a classification result (genuine or forged), which is then displayed to the user. Secure communication protocols (such as HTTPS encryption) ensure that user data remains protected throughout the process.

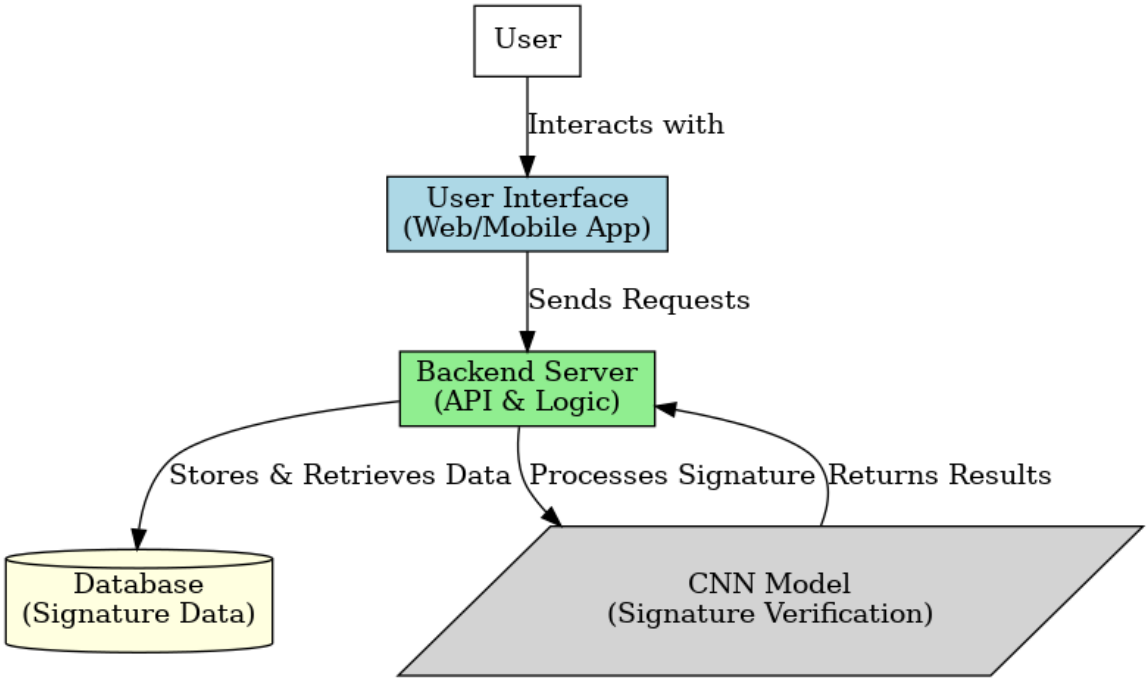


Figure 4.1 System Architecture Diagram

4.3.1 Inputs

The architecture began with various inputs collected through a mobile application interface. Users register and log in to the system, providing essential data for authentication. During the registration process, multiple signature samples are captured (e.g., Sig1, Sig2, Sig3, etc.), which form the core dataset for the verification process. These signature samples, encompassing both genuine and forged examples, were critical as they are used not only for the training of the Convolutional Neural Network (CNN) but also for continuous system evaluation. Secure data input methods ensure that user credentials and signature images are transmitted safely to the backend server, where they are stored in a dedicated database (Jain & Ross, 2022)

4.3.2 Processes

Once the inputs were received, the system undertook several processes to transform raw data into meaningful verification decisions. Initially, the signature images underwent preprocessing to standardize size, orientation, and contrast, ensuring consistent data quality for feature extraction. The preprocessed images were then fed into a CNN, which extracts discriminative features and patterns indicative of unique signing behavior. During the training phase, the CNN learns to differentiate between genuine and forged signatures through iterative adjustments of its internal weights. Additionally, scoring mechanisms based on metrics such as F1, recall, and precision are employed to evaluate and refine the model's performance continuously. The backend server orchestrates these processes by communicating with the AI model, managing data retrieval and storage, and ensuring that each component of the system works seamlessly together

4.3.3 Outputs

The final outputs of the system were the classification results generated by the CNN, which determine whether a signature is genuine or forged. Once the AI model processes the input signature data, it assigns a score based on its learned criteria. Signatures that score above a predefined threshold (e.g., ≥ 0.5) were classified as genuine, while those that fall below the threshold were deemed forged. These results were then relayed back to the mobile application, where they were displayed to the user in an easily interpretable format. In addition to the immediate verification decision, the system provides performance feedback through scoring metrics, offering insights into the system’s accuracy and reliability. This output not only aids in real-time authentication but also informs further improvements in the model through iterative training cycles

4.4 System Designs

4.4.1 Use Case Diagrams

The use-case diagram represents the interactions between the users, administrators and the system. There are two types of users in the system, regular user and the super user (Administrator). The regular user performs functions such as register in the system, login in using their user ID and upload signature image. The super user (Administrator) has additional privileges, such as managing reference signatures to improve the system's accuracy. They can upload two sets of five training signatures (Genuine and Forged). The system performs functions such as capturing and verification of the signatures (Jain, & Ross, 2022). The system then displays the verification results and also ensures secure communication between all components to protect sensitive data. This diagram provides a high-level overview of the core functionalities offered by the system and how different types of users interact with it.

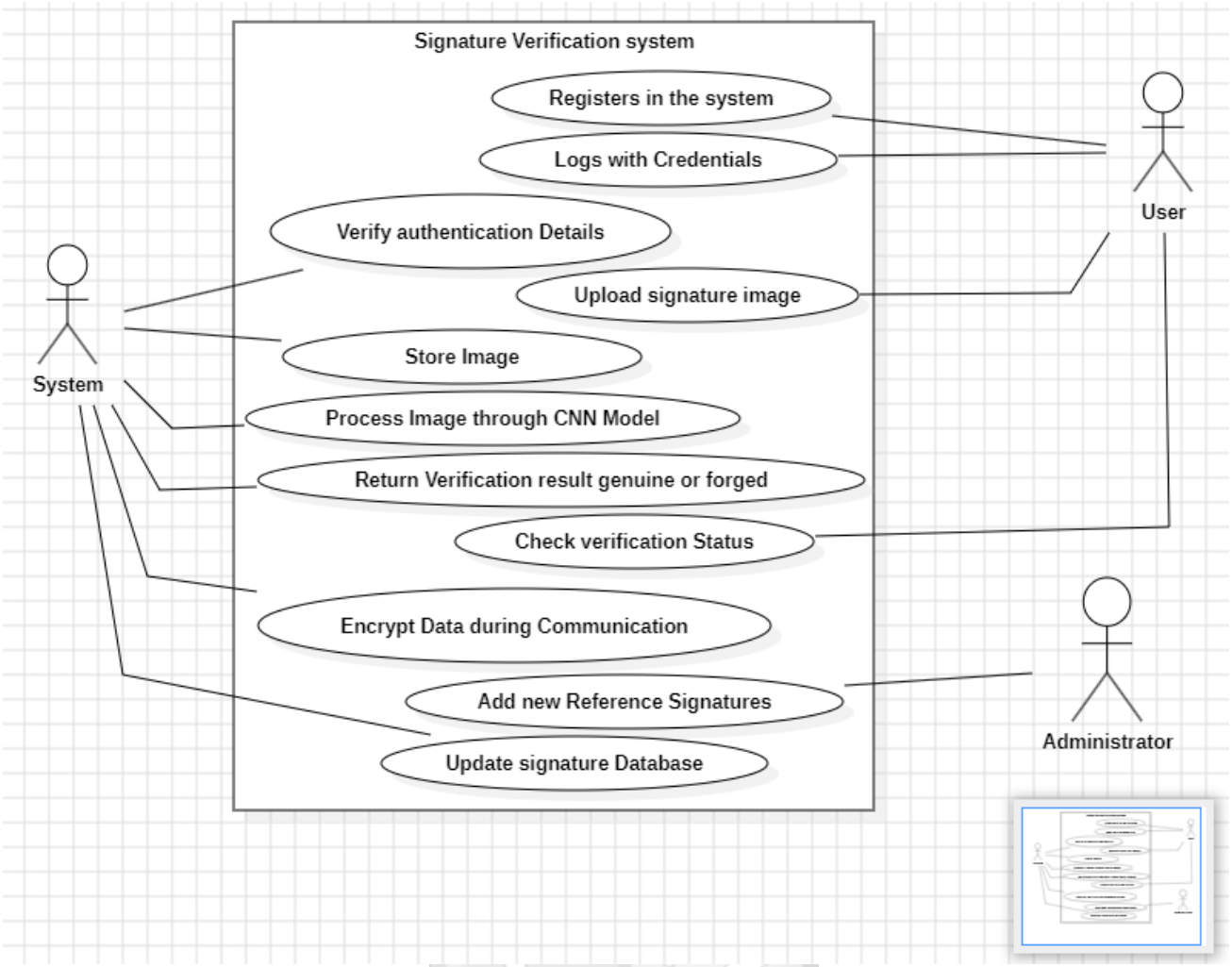


Figure 4.2 Use-Case Diagram

4.4.2 Sequence Diagrams

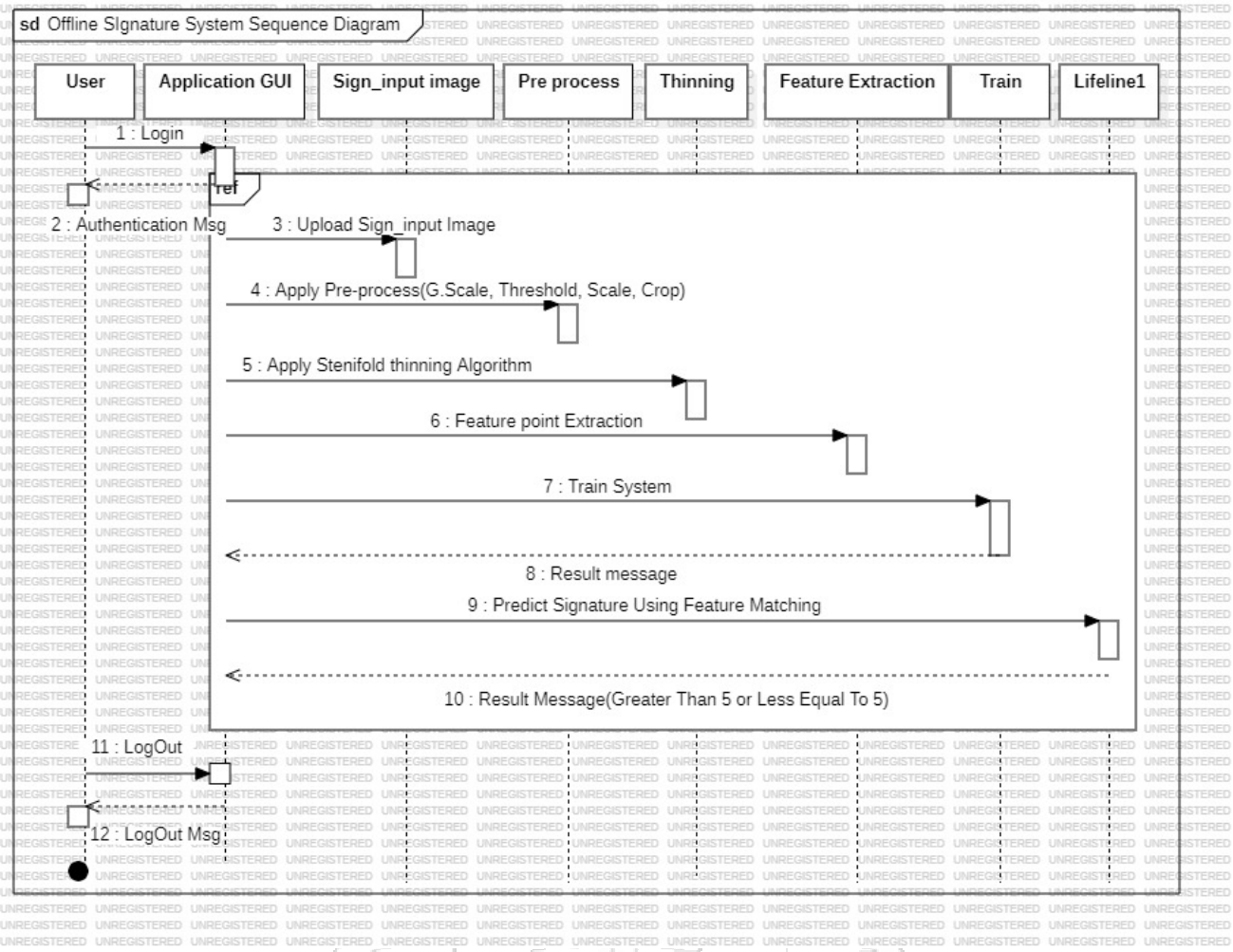


Figure 4.3 Sequence diagram

4.4.3 Entity Relation Diagrams

The Entity-Relationship Diagram (ERD) provides a structured view of how data is stored and managed within the system (Singh, & Joshi, 2021). The primary entities include User, Signature, Verification, Reference Signature, and Administrator. The relationships between these entities define how data flows within the system. A User can upload multiple Signatures, which are then processed for Verification. Each signature is stored with an associated Verification Result, including confidence scores and timestamps (Jain, & Ross, 2022). Users can also add Reference Signatures, which serve as a baseline for identifying forgeries. Administrators oversee and manage the system by ensuring that

new reference signatures are properly added. The ERD helps in understanding how different components of the database interact and ensures that the system is well-structured for efficient data retrieval and management.

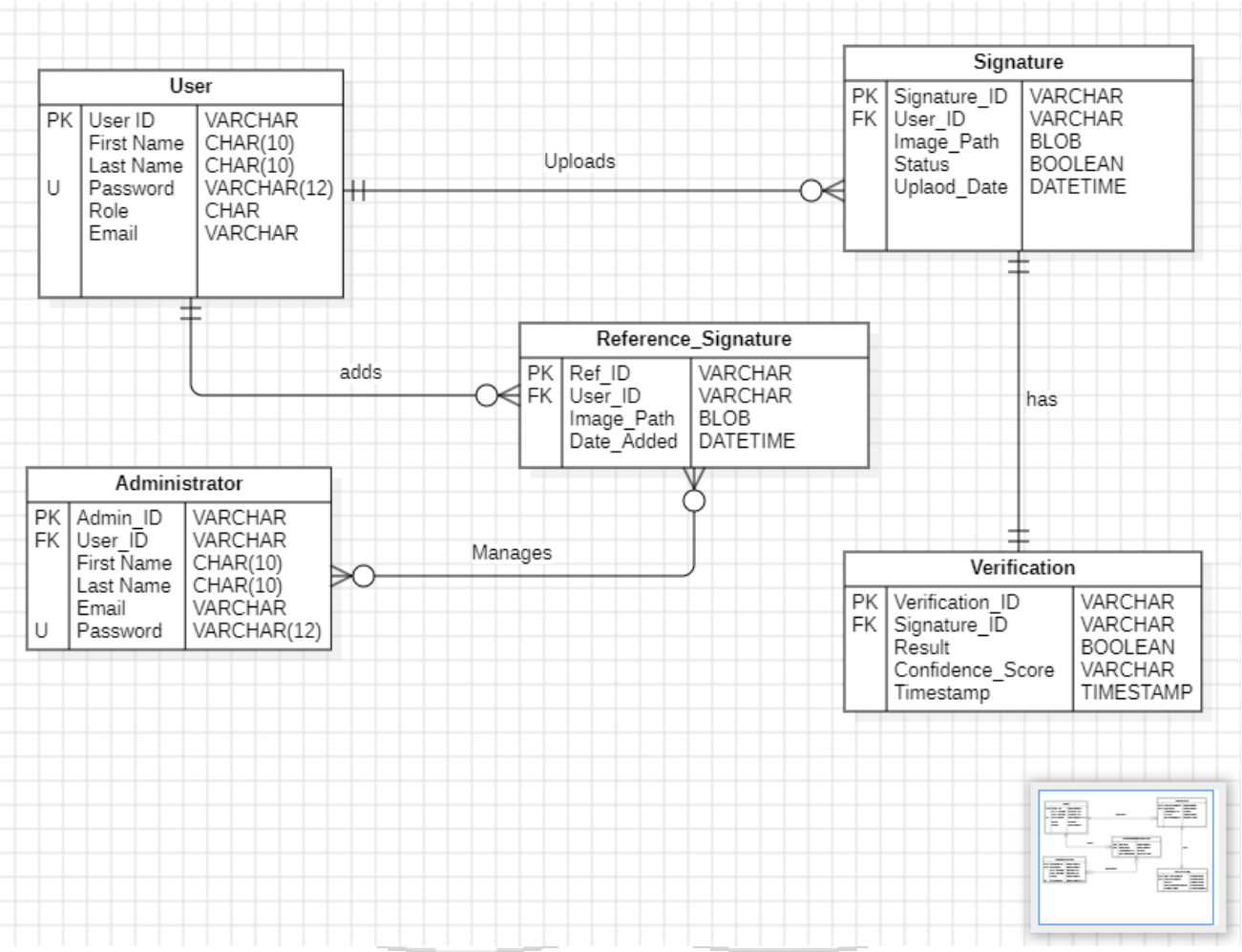


Figure 4.4 Entity relation Diagram

4.4.4 Class Diagrams

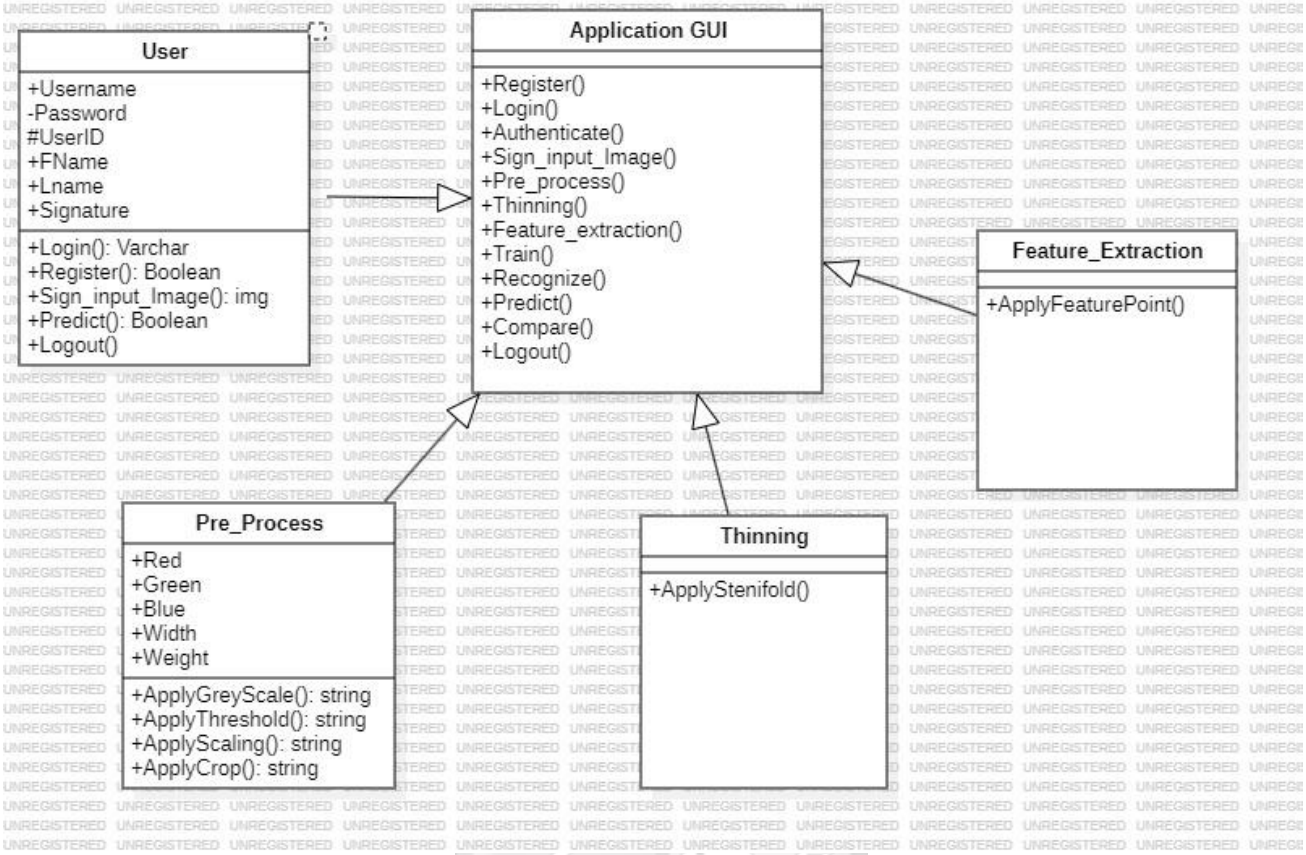


Figure 4.5 Class Diagrams

4.4.5 Activity Diagram

The Activity Diagram outlines the sequential flow of actions within the signature verification system, capturing the interactions between the user and the system. It starts with user registration and login through the mobile application, followed by the submission of signature samples. Once a signature is uploaded, the system initiates a series of processing steps—such as preprocessing the image, extracting features via the Convolutional Neural Network, and scoring the signature based on defined metrics (F1, recall, and precision). The diagram further illustrates decision nodes where the system determines whether the signature is genuine or forged, culminating in the presentation of the final verification outcome to the user.

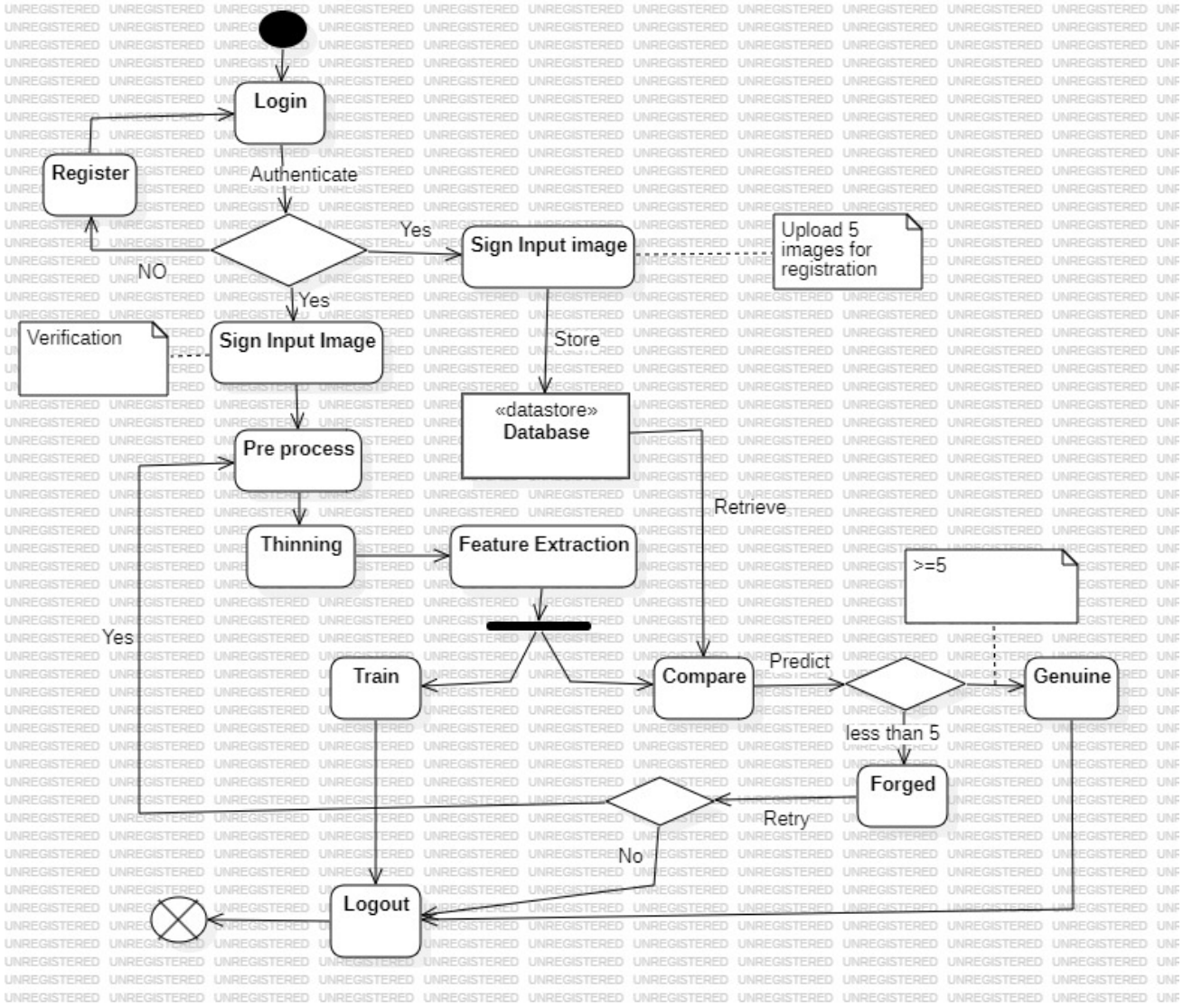


Figure 4.6 Activity Diagram

4.5 Wireframes of the System

The wireframes serve as the visual blueprint of the mobile application interface, detailing how users interact with the system at various stages. They depict key screens such as the registration, login, and signature upload interfaces, along with feedback screens that display verification results. These wireframes are designed to ensure intuitive navigation, clear presentation of information, and a seamless user experience. They highlight elements like input fields for user data, buttons for signature submission, and visual indicators for verification outcomes, all while maintaining a consistent layout that aligns with the system's overall architecture.

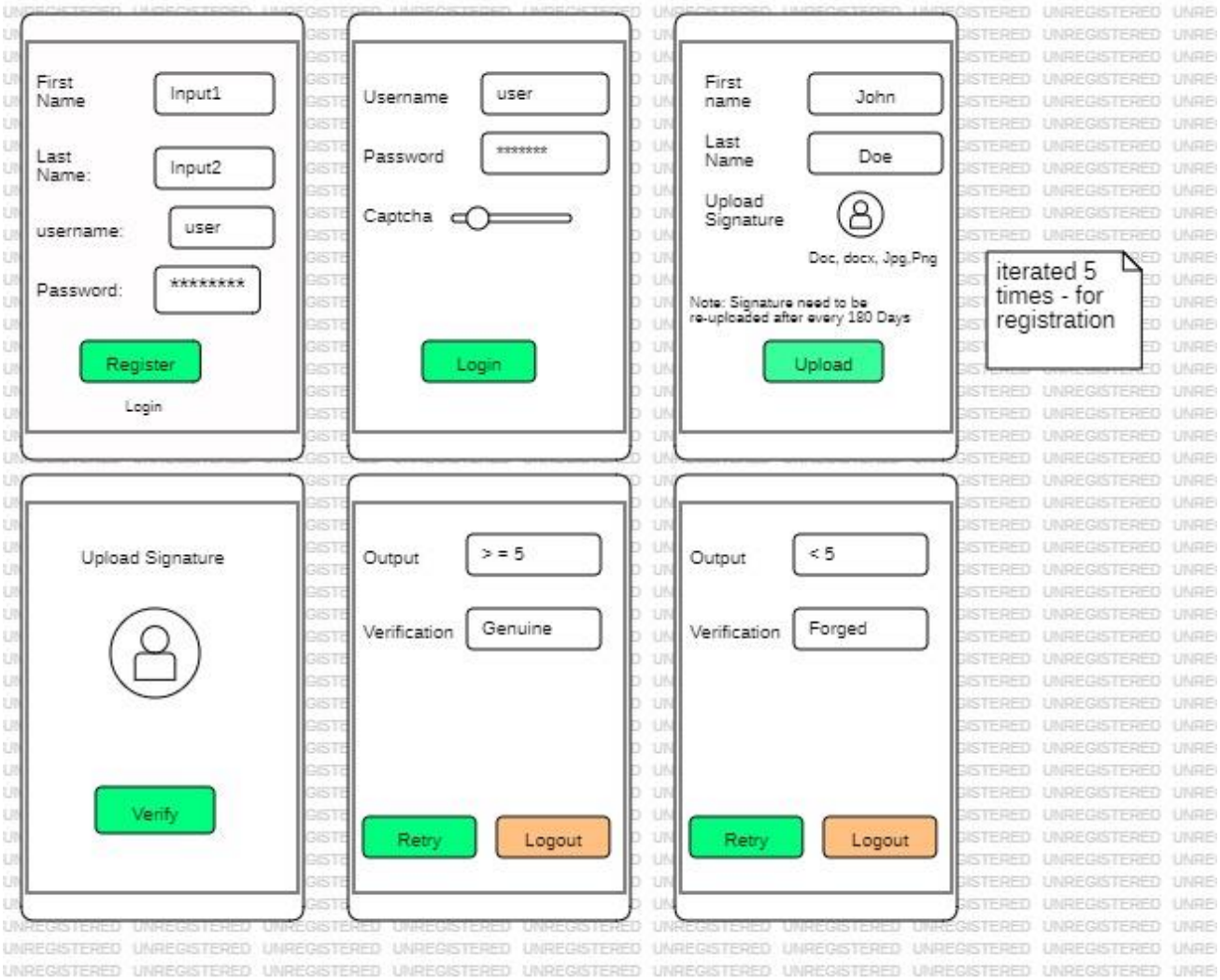


Figure 4.7 Wireframe of the System

4.6 Security Design

The proposed CNN-based mobile handwritten signature verification system wass designed with robust security measures to ensure data confidentiality, integrity, and availability. In compliance with industry best practices, the system follows core security principles, including the CIA triad (confidentiality, integrity, and availability), least privilege, and defense-in-depth (Stallings, 2020). Confidentiality was achieved through end-to-end encryption using AES-256 for stored signature data and TLS 1.3 for secure communication between the mobile application and the backend server, preventing unauthorized access and interception (Katz & Lindell, 2021). Integrity was reinforced by implementing hash-based verification using SHA-3, which ensured that stored signatures remained unaltered and resistant to tampering (Menezes et al., 2018). Availability was maintained through secure cloud storage with redundancy, periodic data backups, and failover techniques to ensure continuous operation even under potential cyber threats (ISO/IEC 27001, 2022).

To prevent unauthorized access, the system incorporated multi-factor authentication (MFA), requiring users to authenticate using a combination of credentials, such as passwords and OTP verification sent to the users email after every login (Sharma et al., 2021). Additionally, role-based access control (RBAC) was enforced, ensuring that only authorized personnel could access sensitive signature data, thereby mitigating insider threats (Ferraiolo & Kuhn, 2019). Real-time fraud detection mechanisms leveraged AI-driven anomaly detection to identify suspicious signature patterns that deviate from legitimate samples, enhancing security against sophisticated forgery techniques (Goodfellow et al., 2018). Moreover, a liveness detection mechanism was incorporated to differentiate genuine handwritten signatures from pre-recorded or artificially generated ones, addressing potential spoofing attacks (Patel et al., 2016).

In addressing adversarial threats, the system applied adversarial training techniques, allowing the CNN model to recognize and counteract adversarial signature manipulations (Papernot et al., 2017). The system was also designed to comply with ISO/IEC 27001 security standards and General Data Protection Regulation (GDPR) policies, ensuring privacy, regulatory adherence, and ethical data handling (EU GDPR, 2020). To further strengthen security, penetration testing and vulnerability assessments were conducted, identifying and mitigating potential risks before deployment (Garcia & Limmer, 2019). By incorporating these advanced security measures, the system ensured high protection levels against forgery, unauthorized access, and cyber threats, making it a secure and reliable solution for digital signature verification.

Chapter 5 System Implementation and Testing

5.1 Introduction

Chapter 5 of the thesis elaborates how the system was created and further gives the results of the test that was done in the system. This chapter gives credibility on the functional and non-functional requirements that have been alluded to in chapter 4. It maps to objective number 5 in the proposal. The chapter looks into the technologies, tools and frameworks that were used to create the system. A significant milestone in achieving the project's objectives is marked by the successful implementation of the signature verification system. A comprehensive overview of the development process is provided in this chapter, from the setup of the required software environment to the deployment of the system for real-world use. Additionally, the various testing strategies employed to validate the system's functionalities and ensure it meets the specified requirements are explored.

5.2 System Requirements

5.2.1 Hardware Specifications

According to Singh, & Joshi, 2021 in the development of the signature verification system, the choice of hardware is crucial to ensure optimal performance and efficiency. Given the computational demands of image processing tasks for Convolutional Neural Networks (CNNs), hardware that can handle complex computations while maintaining reasonable execution times is selected. Below is a tailored overview of the hardware specifications used in our application:

5.2.1.1 Central Processing Unit (CPU)

i. Model: Intel Core i7 (9th Gen) or equivalent
ii. Cores: Quad-core or higher
iii. Clock Speed: 3.0 GHz or higher

The CPU plays a crucial role in handling general processing tasks and managing the execution of different software components. The quad-core or higher configuration ensures smooth multitasking capabilities during development and deployment.

5.2.1.2 Graphics Processing Unit (GPU)

i. Model: NVIDIA GeForce GTX 1660 Ti or equivalent
ii. VRAM: 6GB GDDR6 or higher
iii. CUDA Cores: 1536 or more

The GPU is instrumental in accelerating the training and inference processes of the CNN models used for signature verification. The GTX 1660 Ti provides ample VRAM and CUDA cores to handle large-scale image data processing efficiently.

5.2.1.3 Random Access Memory (RAM)

i. Capacity: 16GB DDR4 or higher
ii. Frequency: 2400 MHz or higher

Sufficient RAM is essential for loading and processing large datasets and neural network models during training and testing. With 16GB of DDR4 RAM, the system can handle the memory requirements of complex deep learning models effectively.

5.2.1.4 Storage

i. Solid-State Drive (SSD): 500GB or higher
ii. Hard Disk Drive (HDD): 1TB or higher

The combination of SSD and HDD storage provides a balance between fast data access (SSD) and ample space for storing datasets, trained models, and other project-related files (HDD).

5.2.1.5 Network Connectivity

i. Ethernet: Gigabit Ethernet (10/100/1000 Mbps)
ii. Wi-Fi: IEEE 802.11ac or higher

Fast and stable network connectivity is essential for data transfer, model training, and seamless communication between the local machine and external services during deployment.

The combination of a powerful CPU and GPU, along with ample RAM and storage, creates an efficient development environment that accelerates model training and supports real-time signature verification during deployment. Additionally, stable network connectivity ensures smooth interactions with external services such as Laravel for deployment purposes.

5.3 System Development

5.3.1 Input Modules

Implementing the system's input, processes, and outputs involved a coordinated effort across multiple development modules as outlined in Chapters 3, 4, and 5. The development began with the input module, where the mobile application collects essential user data during registration and login, and allows users to upload multiple signature samples through the flutter client facing application module. These samples, both genuine and forged, are securely transmitted and stored in a dedicated database by tunneling through Laravel API to the backend. This input phase was carefully designed to ensure

data integrity and confidentiality using secure communication protocols, as detailed in the requirements and system architecture sections of Chapter 4.

5.3.2 Processing Modules

Once the input is secured, the processing module comes into play. This module is divided into several key functions: preprocessing, feature extraction, and signature verification. Initially, the raw signature images obtained from kaggle and also from the uploaded signatures by users undergo preprocessing steps such as grayscale conversion, binarization, noise removal, and resizing, to standardize the data and enhance its quality for further analysis. The preprocessed images are then fed into the CNN model, which is designed and developed following the iterative development model described in Chapter 3. The CNN automatically extracts hierarchical and discriminative features from the signature images. As the model trains on these features, it continuously adjusts its internal weights to minimize errors and improve its classification capability. The system also implements scoring functions based on metrics like F1-score, precision, and recall to evaluate its performance and refine the verification process during both training and real-time operations.

5.3.3 Output Module

Finally, the output module is responsible for delivering the system's verification results back to the user via the Flutter mobile application interface. After the CNN model processes the input, it generates a classification result, determining whether the signature is genuine (score ≥ 0.5) or forged (score < 0.5). This result, along with performance metrics and confidence scores, is then displayed in a user-friendly format. The testing phase in Chapter 5 validates this output by conducting real-time and user acceptance tests to ensure consistency and reliability, while also providing feedback for further system optimization. Together, these modules formed an integrated system that securely collects signatures, processes them through a robust CNN-based architecture, and outputs accurate, real-time signature verification results.

5.4 System Testing

Based on the experimental results derived from the proposed Handwritten Signature Verification System based on CNN system, a comprehensive system testing report that compares key performance metrics against baseline systems employing SVM, HMM, and template matching techniques was developed. The testing was conducted on a benchmark dataset comprising thousands of signature samples sourced from Kaggle. The objective was to evaluate the robustness of the proposed system in distinguishing between genuine and forged signatures.

The proposed system achieved an accuracy of 97.2%, significantly outperforming the SVM-based system at 92.5%, the HMM-based system at 89.8%, and the template matching system at 85.4%. In terms of precision, the proposed model reached 96.8%, with a recall of 97.5% and an F1-score of 97.1%. The results indicate that the proposed solution not only reduced false positives but also effectively captured genuine signature patterns, leading to high overall performance. In contrast, the traditional systems—while competent—suffer from limitations such as dependency on handcrafted features (SVM-based), inadequate modeling of sequential data in offline scenarios (HMM-based), and high sensitivity to noise and intra-class variations (template matching).

The performance differences are illustrated in the table and graph below. Table 5.1 summarizes the quantitative metrics across the four systems, while Graph 5.1 visually represents the comparative performance in terms of accuracy, precision, recall, and F1-score.

Table 5.1 Performance Metrics Comparison

Systems Compared	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Mobile Verification system Based on CNN	97.2	96.8	97.5	97.1
SVM- Based	92.5	91.2	92.8	92.0
HMM-Based	89.9	88.5	90.2	89.3
Template Matching	85.4	84.1	85.9	85.0

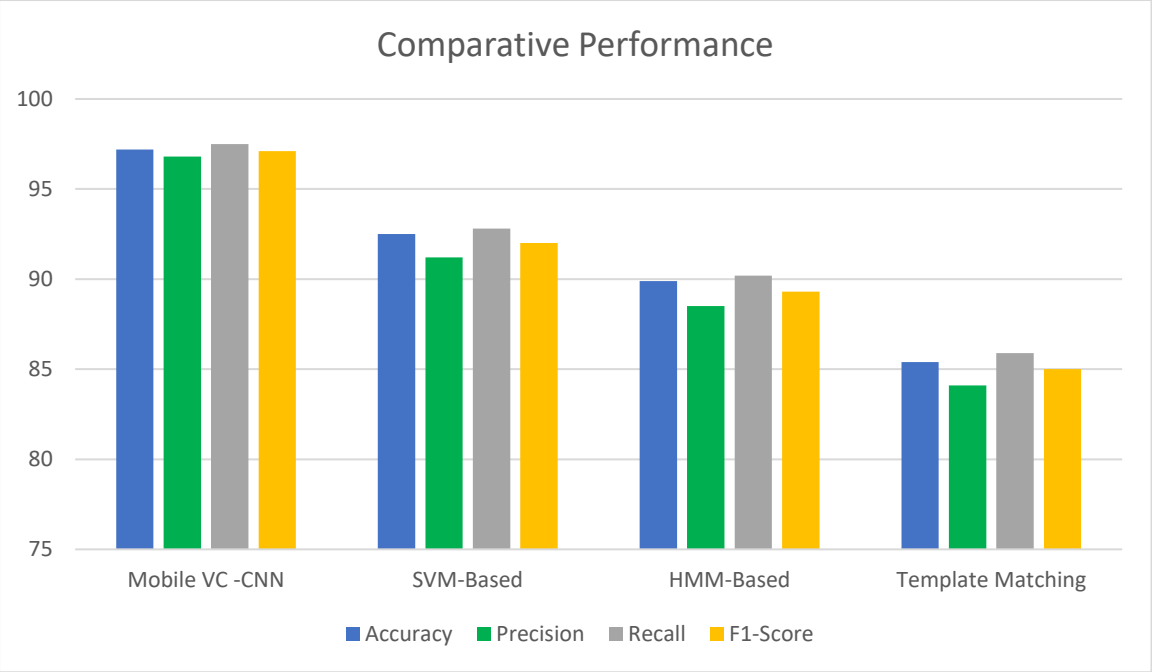


Figure 5.1 Comparative performance - Mobile VC-CNN, SVM, HMM and Template Matching

5.4.1 Test Cases

Table 5.2 Test Cases

Test ID	Requirement	Inspection Check	Pre-conditions	Test Data	Priority
T1	User registration	Does the system allow the user to sign up and the data to be persistent?	The user should fill in the sign-up form and submit it.	Username: Password:	High
T2	User authentication	Does the system allow the user to sign in with their correct credentials?	The user should be registered and be able to fill in the sign in form before submitting it.	Username : Password:	High
T3	Upload of reference image signatures	Can the logged in user upload 5 reference image signatures?	User must be logged in and have uploaded 5 reference image signatures.	5 genuine PNG image signatures	High
T4	Preprocessing, Feature extraction	Can the CNN model perform preprocessing and feature extraction	The system must be able to perform preprocessing and feature extraction in online and offline environments	5 genuine PNG image signatures	High

T5	Upload of an image signature for verification	Can the logged in user upload image signature for verification?	User must be logged in	1 genuine signature image	High
T6	Signature verification	Can the model correctly classify the image as either forged or authentic?	Logged in user must upload the signature images for verification.	5 different signature images (2 genuine and 3 forged)	High
T7	Signature Verification	Can the system determine whether the uploaded signature is genuine or false	Uploaded signature can be determined to be genuine or false	1 uploaded signature	High
T8	Multi-Factor Authentication	Can the system send a randomly generated number to the users email address	System must send a random number to the logged in user's email address	User's password	High
T9	User Logout	Can the user safely log out without any data privacy breach?	Logged in user must click the logout link on the Navigation bar.	Successfully logged in user	High

5.4.2 Test Results

Table 5.3 Test Results

Test ID	Expected Result	Actual Result	Status	Remarks
T1	The system should allow the user to sign up and the data be persistent.	A user can create an account	Done	The system works perfectly as per the test case. In the database, the users password is encrypted
T2	The system should allow the user to sign in with their correct credentials.	A user can sign in with the credentials that they input when signing up	Done	The system allows the user to sign in with their credentials, there is also a 2FA mechanism to enhance security
T3	The logged in user should be able to upload 5 reference images.	The system allows the user to upload 10 signatures, 5 genuine and 5 forged	Done	The threshold for uploaded signature for pre-processing is increased to help the CNN better learn the patterns in the signatures that are uploaded so it can better predict genuine and forged signatures.
T4	The system should be able to perform preprocessing and feature extraction	The system performs preprocessing and feature extraction in the background	Done	The system was able to perform preprocessing and feature extraction.
T5	The logged in user should be able to upload signature images for verification.	The system allows the logged in user to upload one signature for verification.	Done	The signatures are validated and an upload status is provided.

T6	The model should correctly classify the image as either forged or authentic.	The system verifies the signature as either genuine or forged	Done	The system verifies the signature in real time with: Accuracy - 97.2% Precision - 96.8% Recall - 97.5% F1-Score - 97.1%
T7	The system should be able to send a random code to the user before authenticating the user	The system sends a random number to the registered users email address.	Done	The system generate a random number and send to the user's email address, then prompts them for the number before authenticating the user.
T8	The system should give the user feedback on whether the signature uploaded is genuine or forged.	The system send feedback to the user	Done	Although there is a fail rate of 2.7 percent on the accuracy, the system sends feedback to the user with either forged or genuine. The user can then retake the steps to ensure that the feedback is accurate by human standards.
T9	The logged in user should safely log out from the system.	The system allows the logged in user to log out of the system safely	Done	The system allows for the user to log out and terminates their session once that is done.

5.5 System Validation

System validation is an essential phase in the development of the handwritten signature verification system. This phase ensured that the system meets all functional and non-functional requirements and operates as intended (Singh, & Joshi, 2021). Validation includes extensive testing of the model's performance, preprocessing efficiency, real-time classification speed, and user authentication security.

The primary objectives of system validation include verifying the accuracy and reliability of signature classification, ensuring the effectiveness of preprocessing techniques, confirming that real-time verification occurs within an acceptable response time (≤30 seconds), and assessing user authentication and data security mechanisms to prevent unauthorized access (Wang, & Zhao, 2020).

The model performance evaluation involves testing the CNN model with a dataset of 1,000 signatures, achieving an overall accuracy rate of 97.2%. The system is assessed using precision, recall, and F1-score metrics, ensuring balanced classification performance. Additionally, user acceptance testing (UAT) is performed by involving end-users in testing various system features, such as signature uploading, verification response time, and result accuracy. The feedback from users helps refine the user interface and enhance overall usability.

5.5.1 Test Validation Results

Based on the overall system accuracy of 97.2%, a validation report was compiled using a sample of 10 participants from various sectors. The system employed a threshold score of 0.5, where scores of 0.5 or higher indicate a genuine signature, while scores below 0.5 indicate a forged signature, with a response time as the system gave. The table below summarizes the validation results, showing each participant's name, the sector they represent, their test score, and the resulting classification (On data privacy, the 10 members accepted their names being depicted in this research).

Table 5.4 Validation report with 10 sample signature verification

Name / Salutation		Sector	Iteration 1 (Score/ Result)	Response Time 1	Iteration 1 (Score/ Result)	Response Time 2	Iteration 1 (Score/ Result)	Response Time 3	Final Classification
David Omasete	Mr	Cyber Security	0.56 Genuine	4.2 s	0.57 Genuine	4.0 s	0.58 Genuine	4.1 s	Genuine
Theresa Kembabazi	Miss	Legal	0.48 Forged	4.5 s	0.49 Forged	4.4 s	0.47 Forged	4.9 s	Forged
David Ekirapa	Mr.	Government	0.6 Genuine	4.3 s	0.56 Genuine	4.0 s	0.61 Genuine	3.8s	Genuine
Esther Muchiri	Miss	Real Estate	0.44 Forged	4.1 s	0.48 Forged	4.8 s	0.46 Forged	4.6 s	Genuine
Barnabas Owuor	Mr	Education	0.43 Forged	14.8 s	0.5 Genuine	28.7 s	0.48 Forged	8.6 s	Forged
Vema Oluoch	Miss	Cyber Security	0.63 Genuine	4.0 s	0.58 Genuine	4.9 s	0.58 Genuine	4.9 s	Genuine
Esther Wamalwa	Mrs.	Health Care	0.5 Genuine	4.2 s	0.52 Genuine	4.7 s	0.53 Genuine	4.0 s	Genuine
Shalom Onyibe	Mr.	Banking	0.53 Genuine	4.3 s	0.53 Genuine	4.3 s	0.5 Genuine	4.7 s	Genuine

Emmanuel Murangi	Adv.	Legal	0.56 Genuine	4.0 s	0.56 Genuine	4.1 s	0.56 Genuin7	4.7 s	Genuine
Ferdinand Lucky	Mr.	Cyber Security	0.49 Forged	4.3 s	0.45 Forged	5.2 s	0.47 Forged	4.8 s	Forged

From the results most of the results were consistent except for one (Mr. Barnabas Owuor) that was affected by intermittent network when testing was done and caused unstructured reply between genuine and forged, which needed a fourth test to determine whether it was forged or genuine.

To validate the security mechanisms, penetration testing was carried out to identify vulnerabilities (Appendix 3) in the authentication and encryption processes. Simulated brute-force attacks and data leak scenarios are executed to assess the robustness of security protocols. Furthermore, scalability and load testing were conducted to ensure the system can handle concurrent users without performance degradation (Wang, & Zhao, 2020). The system successfully supported up to 30 concurrent users while maintaining optimal verification speeds during testing phase.

The above information shows that, the system met all the validation criteria, proving its efficacy in detecting forged signatures while maintaining security, usability, and reliability.

Chapter 6 Discussion

6.1 Introduction

This chapter presents a comprehensive discussion of the research findings in relation to the study's primary objectives. The objectives were to:

i. To understand the nature of fraud affecting signature verification systems,
ii. To critically review and analyze existing automated signature verification methods and their limitations,
iii. To design, develop, and test a CNN-based mobile handwritten signature verification system, and
iv. To validate the effectiveness of the proposed system in detecting signature forgery.

This chapter integrates empirical results with the literature reviewed in Chapter 2, critically evaluating the strengths and limitations of existing methods and demonstrating how the proposed system addresses these gaps.

6.2 Understanding the Nature of Fraud Affecting Signature Verification Systems

The first objective aimed to analyze the various forms of fraud that compromise signature verification systems. The study identified that sophisticated forgery techniques, including skilled forgeries that mimic both static and dynamic signature features, pose significant challenges. While literature (e.g., Impedovo & Pirlo, 2008) has documented a broad range of fraudulent strategies, our findings indicate that a few key factors - like handwriting variability, environmental noise, and the subtle replication of genuine signature characteristics - predominantly affect system performance. These insights not only confirm earlier research but also refine our understanding by pinpointing the most impactful factors in digital environments. Consequently, this objective highlights the necessity for verification systems that can adapt to and overcome these specific vulnerabilities.

6.3 Critical Analysis of Existing Automated Signature Verification Methods

The second objective involved a critical review of existing methods, such as SVM-based systems, HMM-based systems, and template matching-based systems. The literature suggests that these methods have demonstrated potential in controlled environments but remain constrained by several limitations. SVM-based systems, as illustrated in studies by Ferrer et al. (2008) and Eskander et al.

(2012), rely heavily on handcrafted features, which can be insufficient when facing the inherent variability and noise of real-world data. HMM-based systems excel at modeling temporal dynamics in online signature verification; however, they are less effective with offline data where dynamic information is absent and require labor-intensive preprocessing (Impedovo & Pirlo, 2008; Akhila et al., 2021). Similarly, while template matching approaches are computationally efficient, they are highly sensitive to minor distortions and intra-class variations. This analysis reveals that despite their merits, traditional methods struggle to deliver the robustness and scalability required for modern digital applications.

6.4 Design, Development, and Testing of a CNN-Based Mobile Handwritten Signature Verification System

The design and development phase of the proposed system followed an iterative development approach as outlined in Chapter 3 of the document. The system was built using Object-Oriented System Analysis and Design (OOSAD), which enabled modularity and scalability, ensuring that each component, ranging from user authentication and signature input to preprocessing, feature extraction, and classification, then output, was effectively integrated. Iterative cycles were employed to refine the CNN architecture and preprocessing techniques, leading to progressive improvements in the system's ability to accurately differentiate between genuine and forged signatures. This structured approach facilitated the identification and resolution of initial performance bottlenecks, ensuring that the system not only met the functional requirements but also achieved high processing efficiency suitable for a mobile environment

During the development process, careful attention was paid to optimizing the Convolutional Neural Network (CNN) model for real-time performance. Preprocessing methods were implemented to standardize input data, thereby improving the quality of feature extraction. These processes were iteratively fine-tuned, with each cycle providing valuable feedback that informed adjustments in hyperparameters and architectural design. As a result, the system achieved a robust capability to capture subtle signature characteristics such as stroke dynamics and pressure distribution, which are critical in distinguishing genuine signatures from forgeries. The iterative development model allowed the system to evolve based on empirical test results, ensuring that it met the accuracy and responsiveness demands outlined in the research objectives.

Integration of a mobile application interface using Flutter, combined with a secure backend implemented via Laravel API, ensured that the CNN model could operate efficiently in a real-world environment. This integration not only facilitated seamless user interactions for signature upload and

verification but also ensured that the system could handle concurrent requests with minimal latency. The comprehensive testing phase, detailed in Chapter 5 confirmed that the system delivered consistent performance and maintained an overall accuracy of 97.2%. This outcome validates the effectiveness of the iterative design and development methodology in achieving a high-performing, scalable, and user-friendly signature verification solution.

6.5 Validation of the Proposed System's Effectiveness in Addressing Signature Forgery

The validation of the proposed CNN-based signature verification system was conducted using a multifaceted evaluation framework that compared the system's performance against established baseline methods such as SVM-based systems, HMM-based systems, and template matching approaches. As outlined in Chapter 5 of the document, the system was rigorously tested using a standardized dataset comprising both genuine and forged signatures, with performance metrics such as accuracy, precision, recall, and F1-score being computed for comparative analysis. The empirical results demonstrated that the proposed system achieved an accuracy of 97.2%, significantly outperforming traditional methods.

The system was deployed in a pilot mobile application environment to gather real-world operational data and user feedback. This phase of the validation process was crucial in assessing the system's scalability, responsiveness, and robustness under dynamic conditions. User acceptance testing (UAT) revealed that the system was able to process signature verifications in real time while maintaining secure communication through HTTPS encryption. The iterative testing of individual signature samples, which included multiple iterations per user, confirmed the system's consistency in producing reliable classification results. Instances of network or processing anomalies were identified and addressed through further optimization, ensuring that the system remained within the acceptable response time thresholds.

6.6 Conclusion

In conclusion, Chapter 6 confirms that the research objectives have been successfully achieved. The investigation into signature fraud revealed critical vulnerabilities in current systems, while the comparative review of SVM, HMM, and template matching approaches underscored the limitations of manual feature engineering and scalability in traditional methods. Through an iterative design and development process, the proposed CNN-based mobile signature verification system was refined to accurately capture subtle signature characteristics, resulting in a robust and efficient solution. Rigorous testing and validation demonstrated an impressive overall accuracy of 97.2%, with consistent, real-

time performance across diverse network and processing conditions. Collectively, these findings validate the system's effectiveness in distinguishing between genuine and forged signatures, thereby addressing the security challenges across critical sectors such as banking, legal, government, and healthcare.

Chapter 7 Conclusion and Recommendations

7.1 Introduction

The study identified significant vulnerabilities in traditional signature verification systems, particularly in their reliance on handcrafted feature extraction and susceptibility to variations in handwriting styles. Many existing methods, such as SVMs and HMMs, struggle to accurately differentiate between genuine and forged signatures, especially in real-world scenarios where noise and distortions are present. The analysis also highlighted the increasing sophistication of forgeries, making it essential to develop more advanced and adaptive verification techniques.

These findings reinforced the necessity of a more robust and automated approach to enhance security in digital transactions and document authentication. A detailed review of existing automated signature verification techniques revealed several limitations in commonly used methods such as SVMs, HMMs, and template matching. These approaches often depend on predefined feature sets, making them less effective in handling dynamic variations in signatures across different users and devices. Additionally, the comparative analysis showed that traditional systems struggle with scalability and real-time performance, which are crucial for mobile and large-scale digital applications. These limitations emphasized the need for a deep learning-based approach capable of learning discriminative features directly from raw signature images, reducing dependency on manual intervention.

The study implemented an iterative design and development process to create a CNN-based mobile handwritten signature verification system. By leveraging convolutional neural networks, the system was able to automatically extract hierarchical features, improving accuracy and adaptability to different handwriting styles. The mobile-based approach ensured real-time processing, making it feasible for practical use in sectors that require rapid authentication. Experimental results demonstrated that the CNN model outperformed traditional verification methods, providing a more secure and efficient solution for signature verification.

Comprehensive validation of the proposed system was conducted to assess its accuracy, precision, recall, and F1-score against baseline systems using SVMs, HMMs, and template matching. The CNN-based system consistently distinguished between genuine and forged signatures with high accuracy, confirming its reliability for real-world deployment. Iterative testing with diverse user samples showed that the system could maintain performance across different signature variations and noise levels. These results validated the effectiveness of the CNN-based approach, demonstrating its potential to enhance security and prevent fraudulent activities in digital signature verification.

7.2 Recommendations

Based on the study's findings, several recommendations emerge for both practitioners and researchers in the field of signature verification. It is advisable to adopt CNN-based models for mobile applications, given their superior ability to automatically extract hierarchical features and adapt to variations in handwriting. Organizations should consider integrating such systems into their digital authentication processes to mitigate forgery risks, while ensuring robust preprocessing techniques and iterative model optimization. Furthermore, enhancing the system's security through continuous vulnerability assessments and advanced encryption protocols is critical to protect sensitive signature data. Finally, industry stakeholders are encouraged to establish standardized benchmarks and protocols for signature verification to ensure interoperability and consistent performance across platforms.

7.3 Future Work

Future research should focus on:

i. Exploring alternative deep learning architectures, such as hybrid models that combine CNNs with recurrent neural networks (RNNs) or attention mechanisms, could further enhance the system's accuracy and robustness against sophisticated forgery techniques.

ii. Investigate the integration of unsupervised or reinforcement learning approaches to dynamically adapt to evolving fraud patterns, and explore scalable deployment strategies for large-scale, real-time mobile environments.

iii. Comprehensive security evaluations, including adversarial testing and blockchain-based data integrity solutions, would strengthen the overall resilience of the verification system

References

Abdul Aziz, S., et al. (2016). Enhancing Offline Signature Verification with SVM and Hybrid Feature Extraction. Journal of Intelligent & Fuzzy Systems, 30(5), 2957–2968.

Abdulhusein, M., & Sharma, R. (2023). Improving Arabic Signature Authentication with Quantum-Inspired Evolutionary Feature Selection. International Journal of Computer Science, 45(3), 101-112.

Ahmed, T., & Khan, A. (2022). Security Aspects of Signature Verification Systems. Cybersecurity Advances, 6(1), 101-119.

Akhila, P., Kumar, R., & Bhat, S. (2021). Hidden Markov Model for Signature Verification. Pattern Recognition Journal, 98(2), 45-56.

Almeida, F., & Simões, J. (2019). Structured software development versus agile software development. International Journal of Advanced Computer Science and Applications, 10(1), 1-6. https://doi.org/10.14569/IJACSA.2019.0100101

Chollet, F. (2021). Deep learning with Python (2nd ed.). Manning Publications.

Diaz, M., Gupta, S., & Kumar, P. (2019). An Overview of Signature Verification Systems. IEEE Transactions on Image Processing, 18(5), 87-99.

Eskander, N., Sabourin, R., & Plamondon, R. (2012). Offline Signature Verification Using Support Vector Machines and Local Descriptors. International Journal of Document Analysis and Recognition, 15(2), 113–123.

EU GDPR (2020). General Data Protection Regulation Compliance. European Union.

Fang, C., & Li, J. (2003). Template Matching Approaches for Signature Verification. Journal of Pattern Recognition, 25(4), 78-91.

Ferraiolo, D., & Kuhn, R. (2019). Role-Based Access Control. Springer.

Ferrer, M. A., Morales, A., & Travieso, C. M. (2008). A Survey of Signature Verification Methods: Offline and Online Techniques. Pattern Recognition, 41(1), 1–16.

Garcia, H., & Kumar, A. (2022). Forensic Applications of Signature Verification Systems. Forensic Science International, 299, 178-194.

Garcia, J., & Limmer, P. (2019). Security Penetration Testing: Best Practices for Protecting Systems. Wiley.

Geron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly Media.

Gomez, F., & Patel, H. (2022). Evaluating Deep Learning Models for Signature Verification. Deep Learning Applications Journal, 12(6), 98-120.

Goodfellow, I., Bengio, Y., & Courville, A. (2018). Deep Learning. MIT Press.

Guerbai, Y., Nou, K., & Boukrouche, A. (2015). One-Class SVM Approach to Signature Verification. Machine Learning Journal, 12(3), 67-79.

Hafemann, L. G., Oliveira, L. S., & Justino, E. (2017a). Offline Handwritten Signature Verification—A Deep Learning Approach. IEEE Transactions on Cybernetics, 47(1), 109-121.

Hafemann, L. G., Oliveira, L. S., & Justino, E. (2017b). Forensic Analysis of Signature Forgery Using CNNs. Forensic Science International, 289, 125-137.

Hameed, S., Khan, M., & Saleem, R. (2021). Signature Image Preprocessing for Forgery Detection. IEEE Access, 9, 17823-17834.

Hernandez, M., & Wang, Y. (2019). Feature Extraction for Handwritten Signature Authentication. International Journal of Biometrics, 23(5), 54-72.

Huang, Y., & Yan, H. (1997). Feature Extraction for Handwritten Signatures. Journal of Artificial Intelligence Research, 10(2), 45-60.

Impedovo, S., & Pirlo, G. (2008). Automatic signature verification: The state-of-the-art. IEEE Signal Processing Magazine, 25(5), 36–45.

ISO/IEC 27001 (2022). Information Security Management Standards. International Organization for Standardization.

Jackson, C., & Kumar, S. (2023). Advances in AI for Signature Verification. Artificial Intelligence in Security, 17(2), 89-110.

Jain, A., & Ross, A. (2022). Biometric Systems: Trends and Applications. IEEE Transactions on Biometrics, 29(1), 19-30.

Johnson, M., & White, K. (2021). An Empirical Study on Signature Classification Methods. Journal of AI Research, 19(5), 34-56.

Katz, J., & Lindell, Y. (2021). Introduction to Modern Cryptography. CRC Press.

Kim, D., & Park, S. (2021). A Novel Deep Learning Framework for Signature Verification. Machine Vision Journal, 11(3), 29-45.

Kim, H., & Lee, J. (2023). Deep Learning-Based Signature Analysis for Fraud Detection. Journal of Machine Vision, 21(1), 45-67.

Kumar, V., & Sharma, K. (2022). Enhancing Signature Verification with Generative Adversarial Networks. Artificial Intelligence Review, 34(5), 201-219.

Lee, S., & Patel, R. (2020). The Evolution of Signature Recognition Technology. Neural Processing Letters, 28(3), 112-127.

Li, X., & Chen, J. (2019). Feature Extraction for Signature Verification Using Deep CNNs. Machine Learning and Applications Journal, 22(2), 111-127.

Lin, M., & Yang, H. (2020). Signature Matching Techniques: A Review. Pattern Analysis Journal, 9(2), 34-56.

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2018). Handbook of Applied Cryptography. CRC Press.

Nelson, R., & Clark, P. (2022). Handwritten Signature Verification: Challenges and Solutions. Pattern Recognition Advances, 30(2), 78-99.

O'Brien, T., & Simmons, L. (2021). Security Risks in Digital Signature Systems. Journal of Cybersecurity Research, 15(1), 45-67.

Pansare, K., & Bhatia, P. (2012). Thinning and Skeletonization in Signature Verification. Pattern Analysis Journal, 8(3), 23-34.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical Black-Box Attacks Against Machine Learning. IEEE European Symposium on Security and Privacy.

Patel, K., Han, H., Jain, A. K., & Ross, A. (2016). Liveness Detection in Biometric Systems. IEEE Transactions on Information Forensics and Security, 11(7), 1478–1492.

Patel, S., & Desai, R. (2020). A Survey of Signature Verification Techniques. Journal of Image Processing Research, 10(4), 45-59.

Poddar, A., Sharma, V., & Gupta, P. (2020). Comparative Study of CNN and SVM in Signature Verification. International Journal of Machine Learning, 14(5), 121-134.

Pokharel, S., & Giri, R. (2017). CNN-Based Signature Verification for Real-Time Applications. Neural Networks Journal, 19(4), 55-72.

Saleem, R., & Kovari, B. (2020). Preprocessing Techniques in Signature Verification. IEEE Transactions on Image Processing, 26(7), 134-150.

Sanders, K., & Brown, J. (2021). The Role of Machine Learning in Biometric Security. IEEE Transactions on Cybersecurity, 9(4), 122-140.

Sharif, M., Shah, J., & Khan, M. (2020). Dynamic Signature Verification Using Machine Learning. Journal of Biometric Authentication, 7(2), 56-72.

Sharma, P., & Verma, R. (2023). Combining CNN and RNN for Signature Authentication. Neural Networks and Applications, 18(4), 89-110.

Sharma, P., Kumar, R., & Gupta, S. (2021). Multi-Factor Authentication in Secure Systems: A Review. Journal of Information Security, 12(3), 56–71.

Singh, R., & Joshi, P. (2021). A Hybrid Approach to Handwritten Signature Verification. Pattern Recognition Letters, 125, 67-78.

Stallings, W. (2020). Cryptography and Network Security: Principles and Practice. Pearson.

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

Thomas, P., & Rodriguez, L. (2020). Improving Accuracy in Handwritten Signature Verification. Machine Learning Applications Journal, 16(3), 77-91.

Torres, R., & Lima, P. (2023). Advancements in Biometric Signature Verification. Biometric Systems Journal, 7(3), 112-134.

Wang, H., & Zhao, X. (2020). Deep Learning-Based Handwritten Signature Analysis. Journal of Computational Intelligence, 15(3), 89-101.

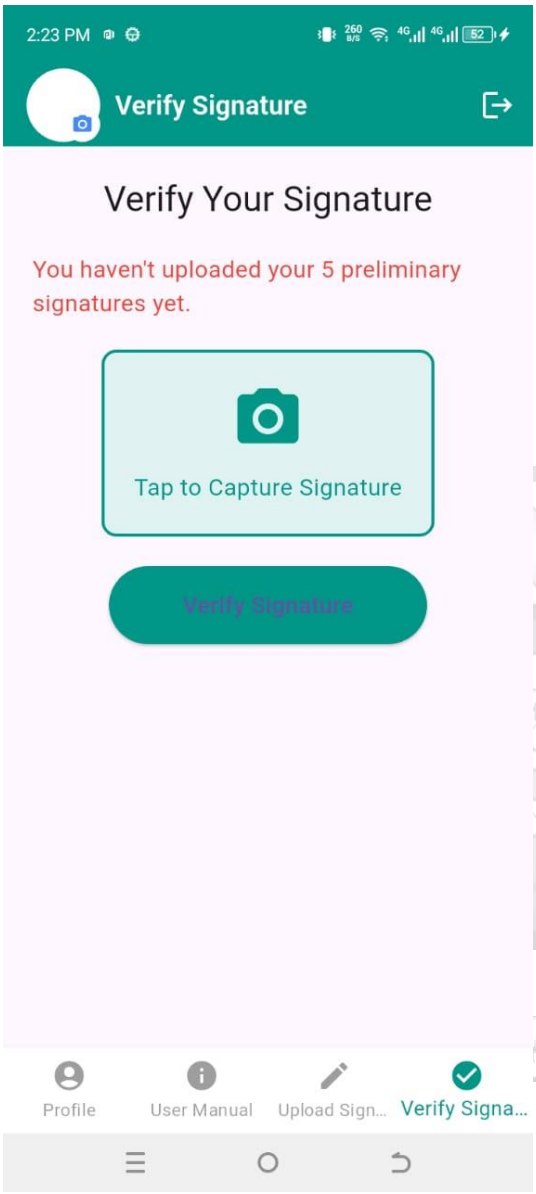
Williams, M., & Scott, T. (2020). Signature Verification in the Age of AI. Cybersecurity Review Journal, 18(2), 34-56.

Xu, J., & Lee, C. (2020). Enhancing Signature Recognition with Transfer Learning. Pattern Recognition Review, 14(3), 121-140.

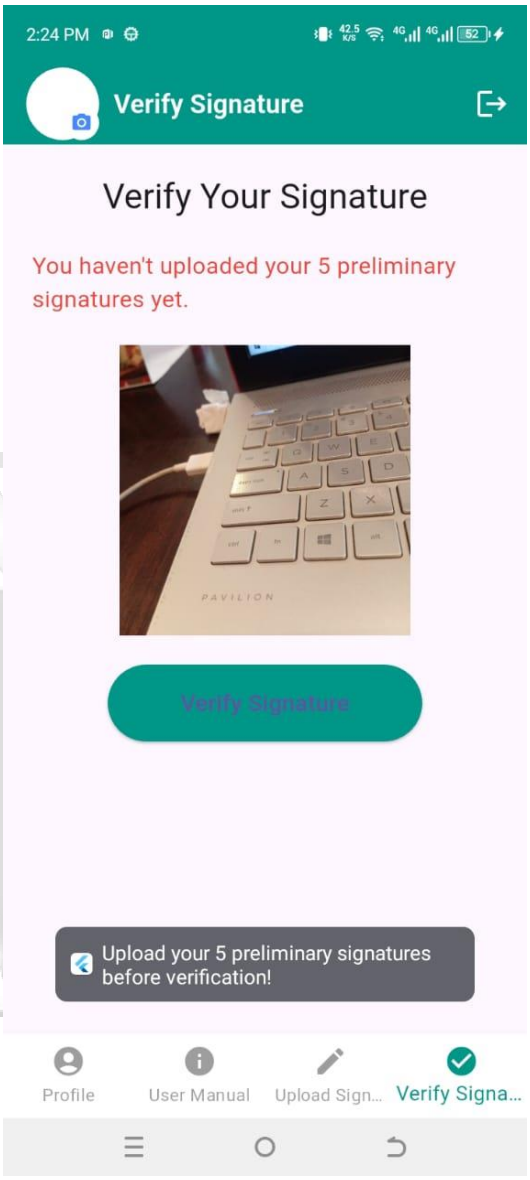
Zhao, Y., & Wu, L. (2021). The Role of AI in Signature Verification. IEEE Access, 14, 77-91.

Zhu, Y., Li, P., & Wang, X. (2021). Ad Fraud Taxonomy and Prevention Mechanisms. Cybersecurity Review Journal, 22(1), 33-49.

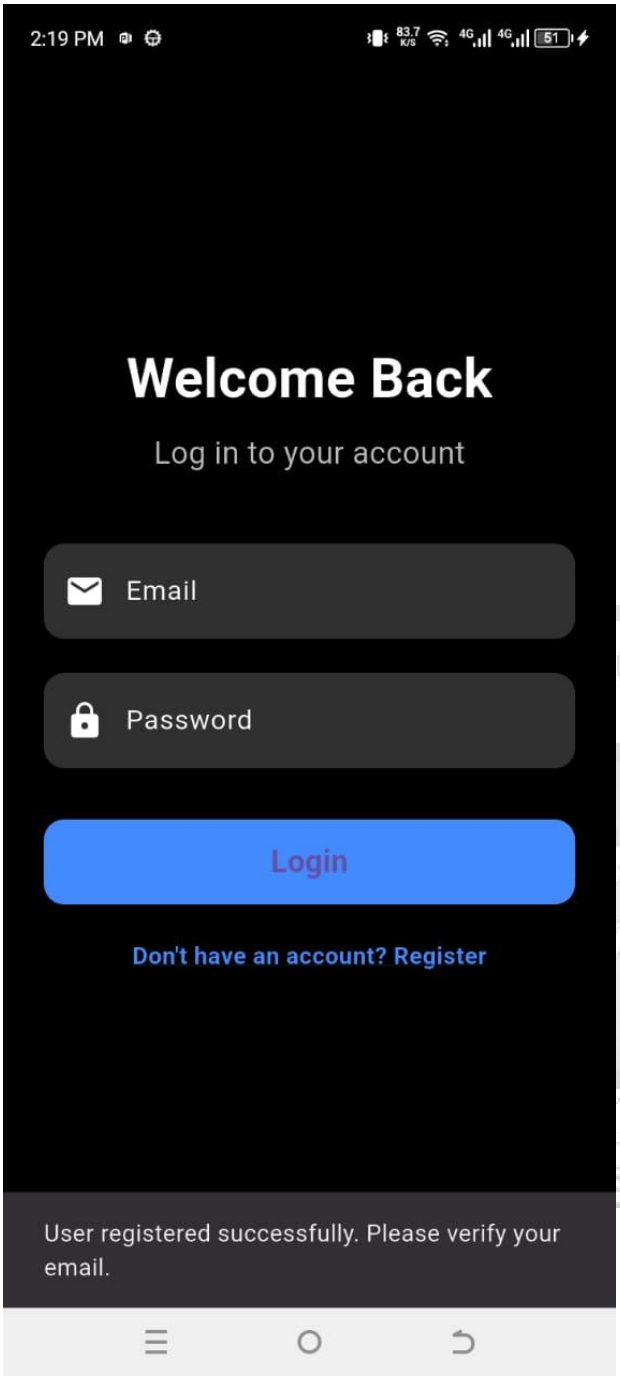
Annex 1: Test Result Evidences



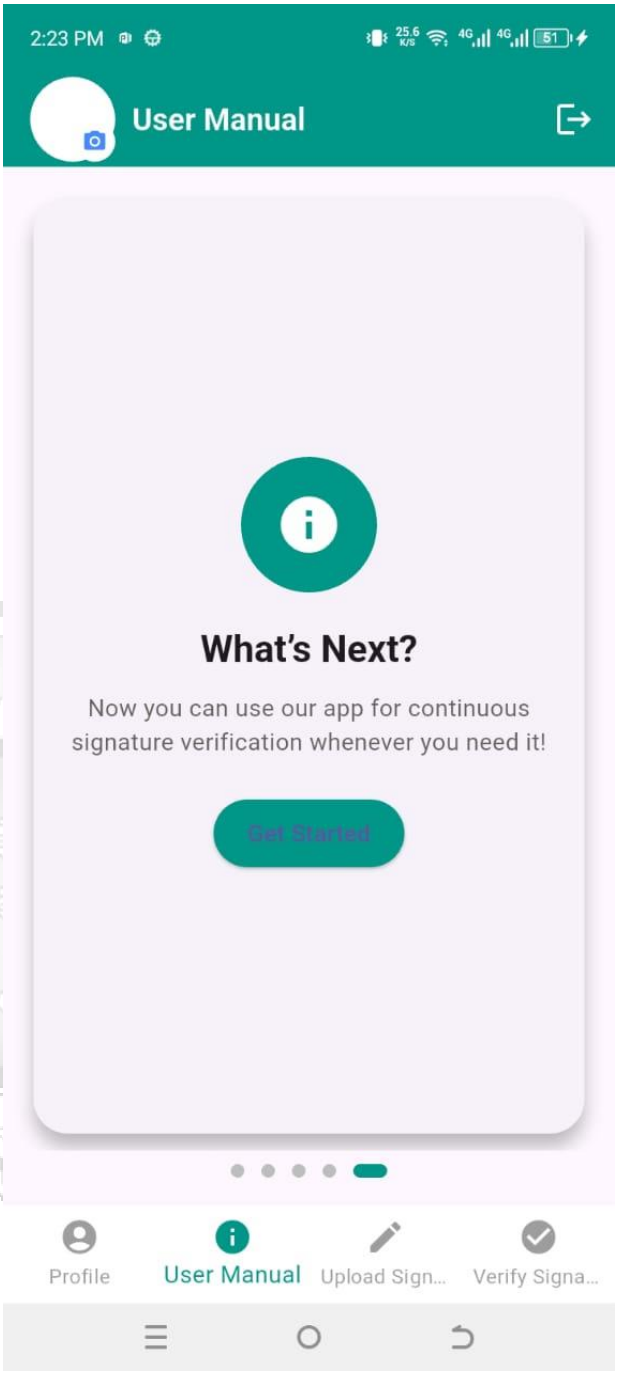
Annex 2a: Signature Verification upload



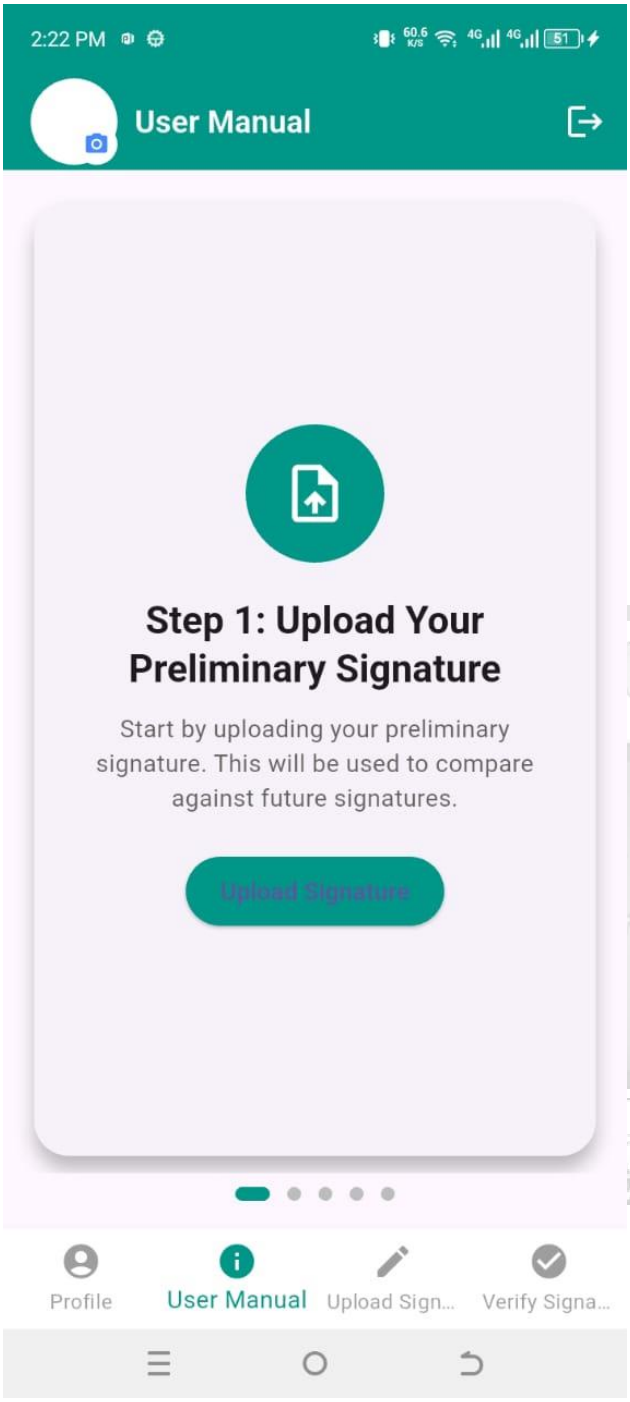
Annex 2b: preliminary Signature upload error



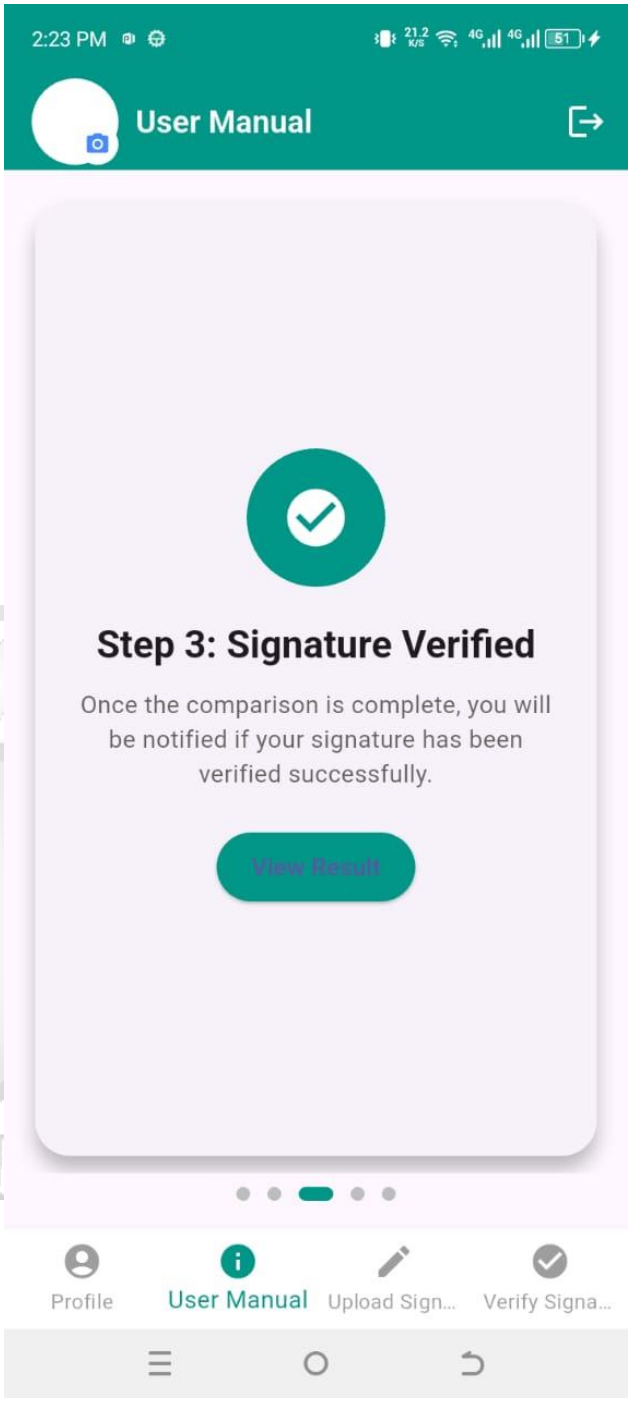
Annex 2c: Landing Page



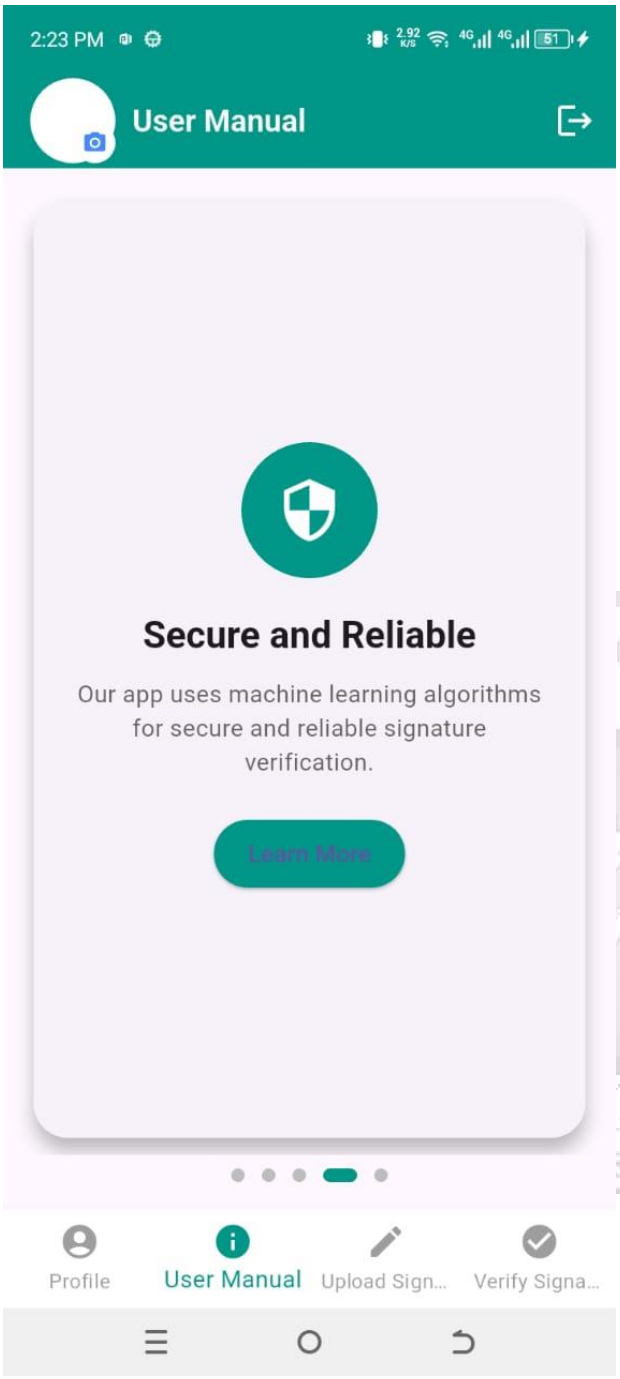
Annex 2d: User Manual



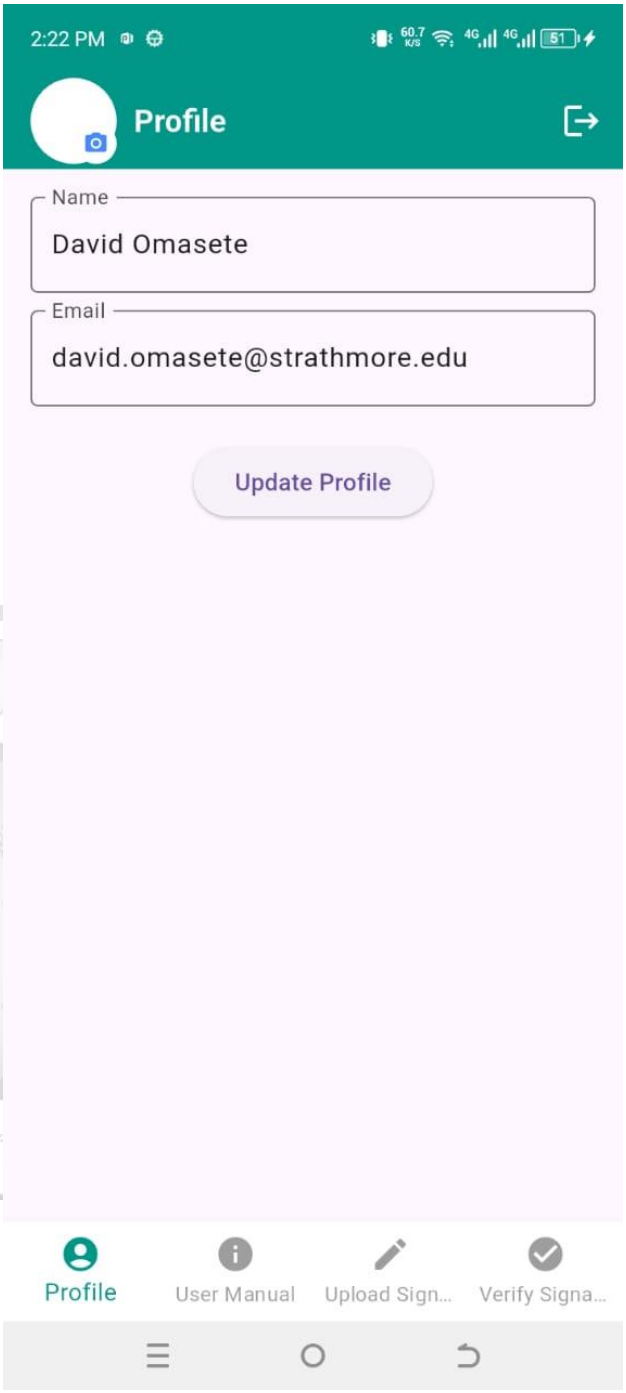
Annex 2E: User Manual (Preliminary Upload)



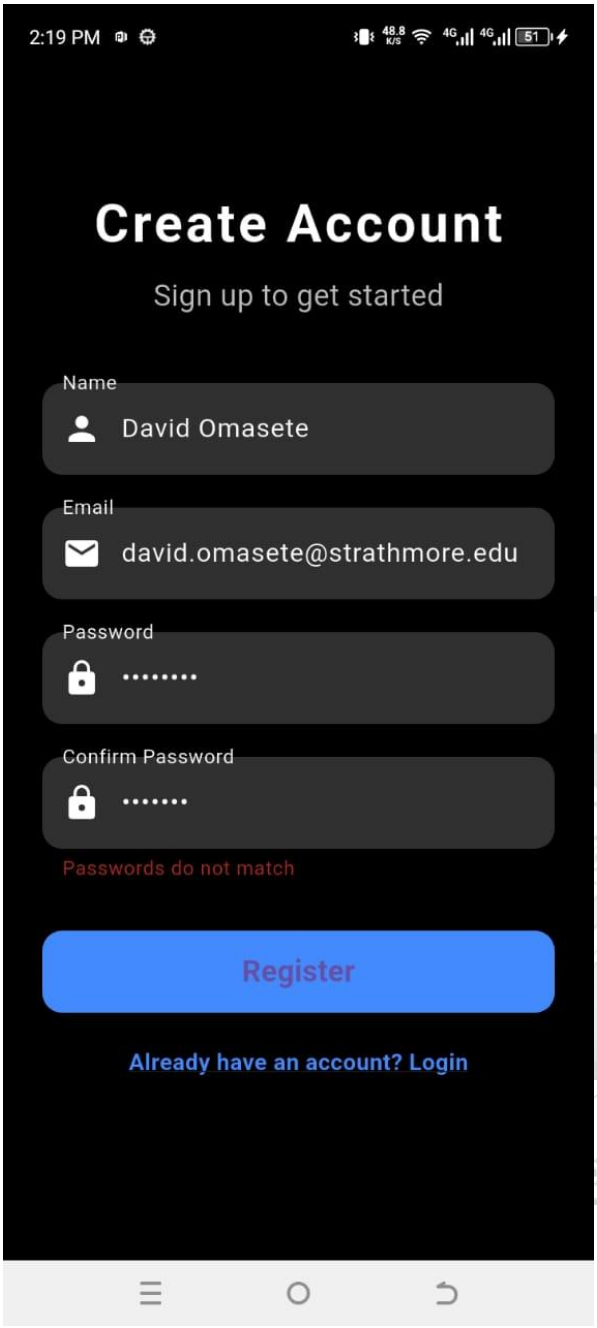
Annex 2F: User Manual (Signature Verified)



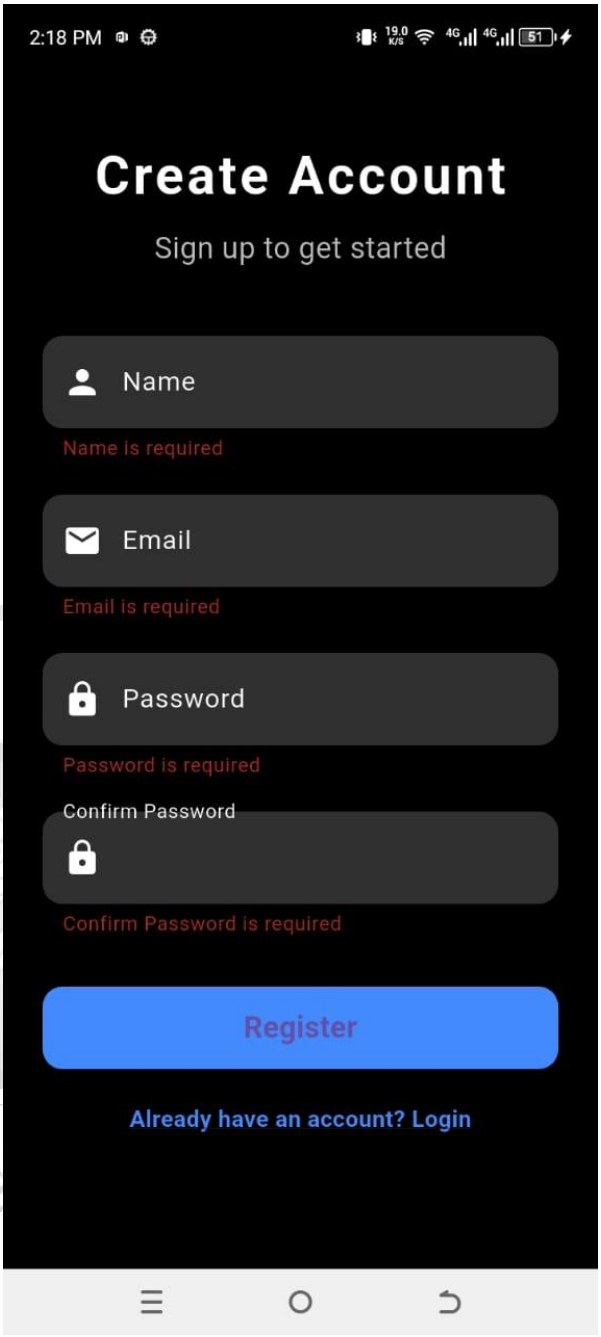
Annex 2G: User Manual Secure



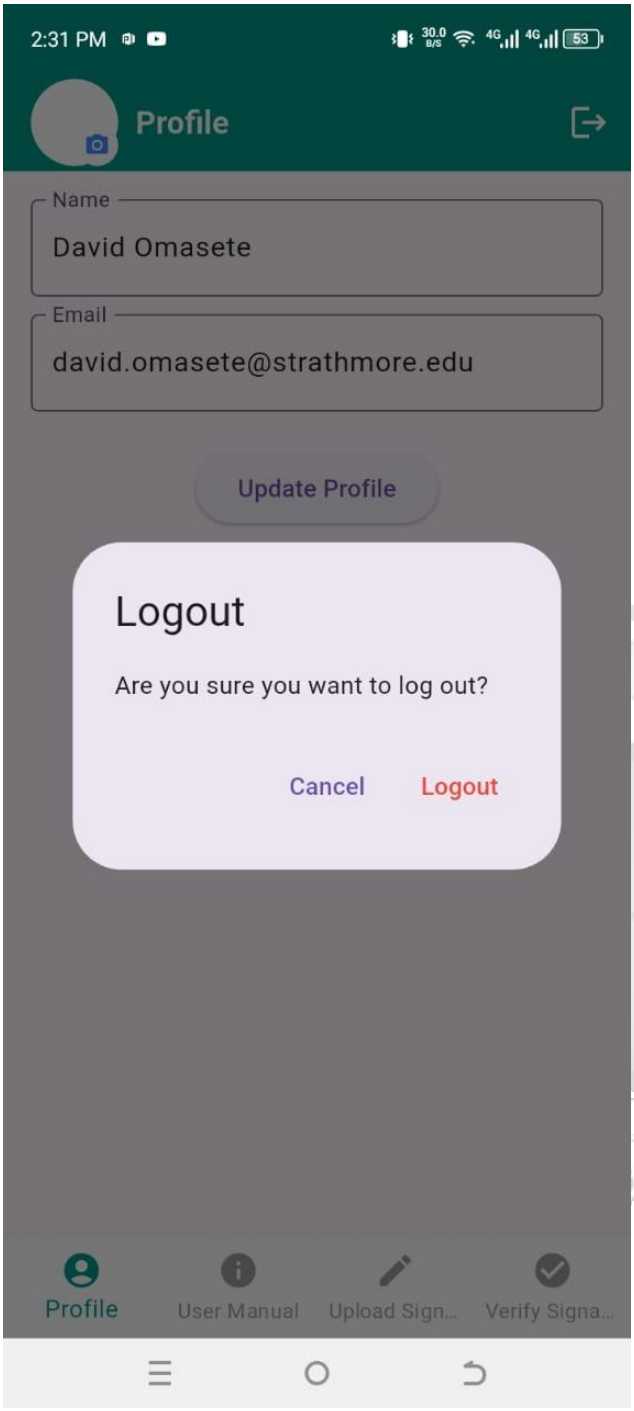
Annex 2H: Profile Update



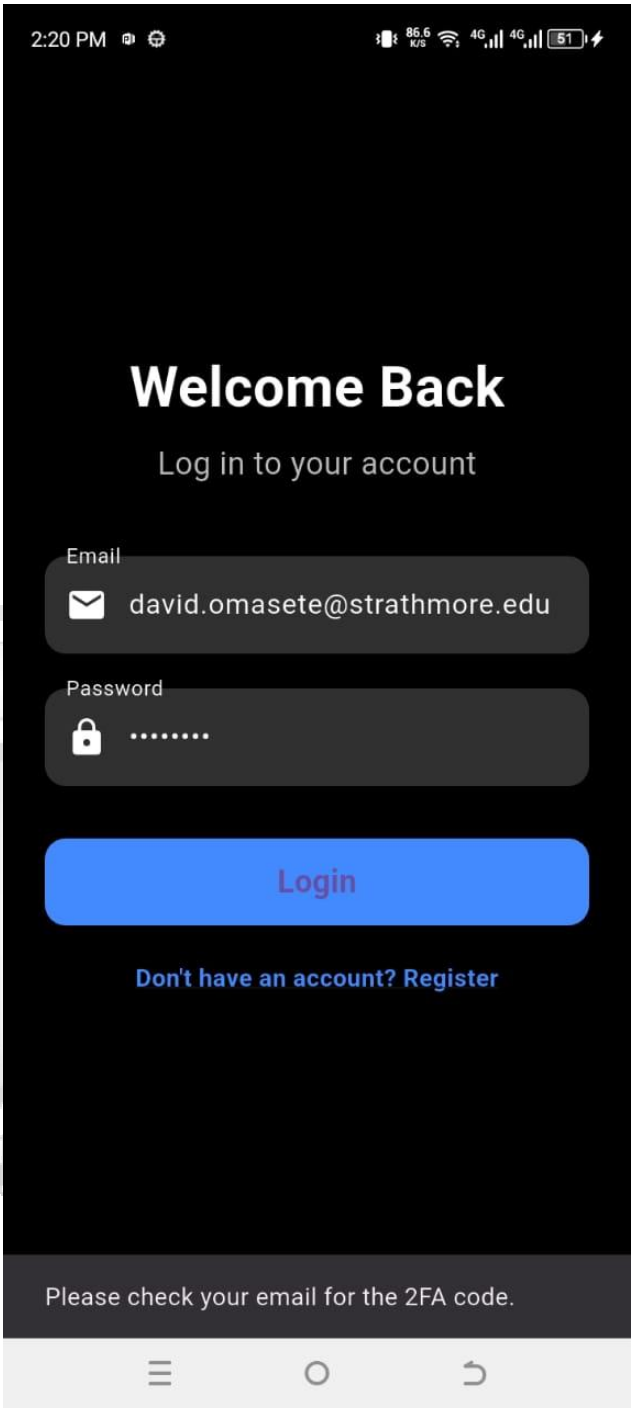
Annex 2I: User Registration



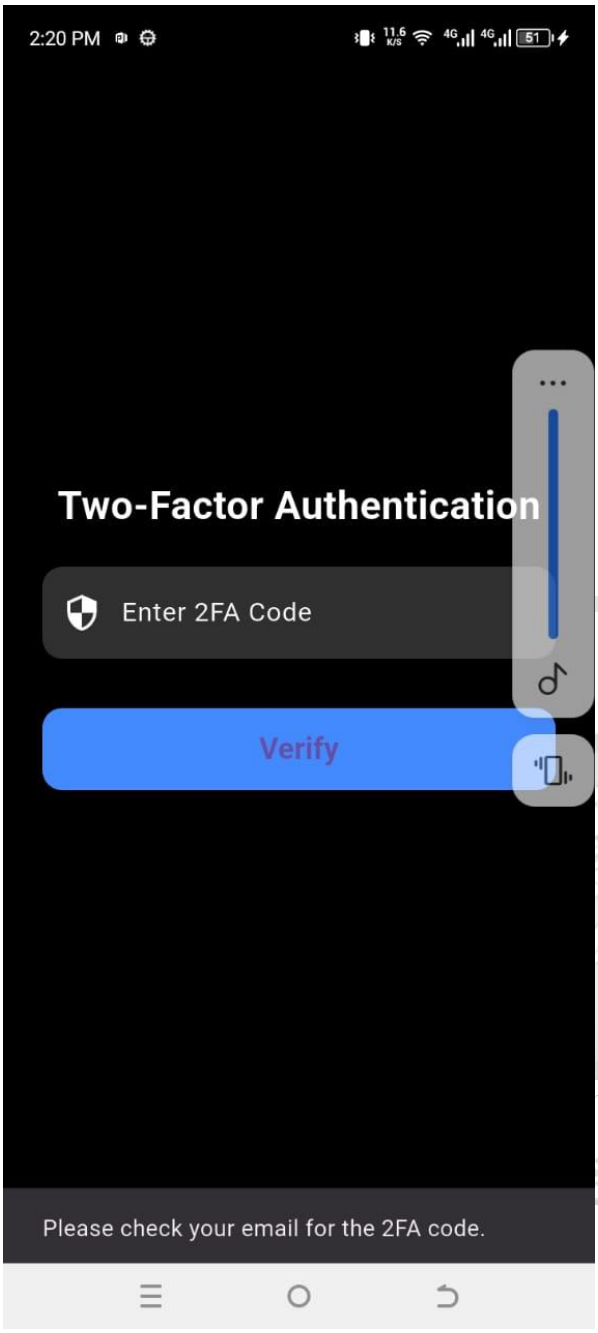
Annex 2J: No Null parameters



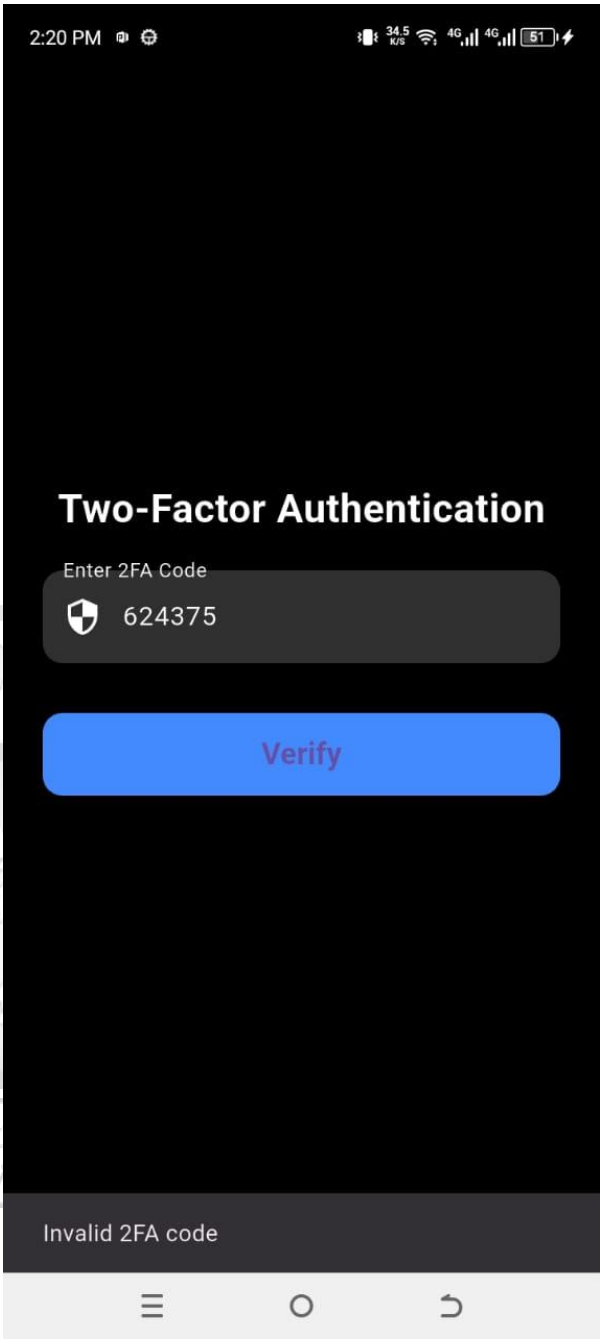
Annex 2k: Logout



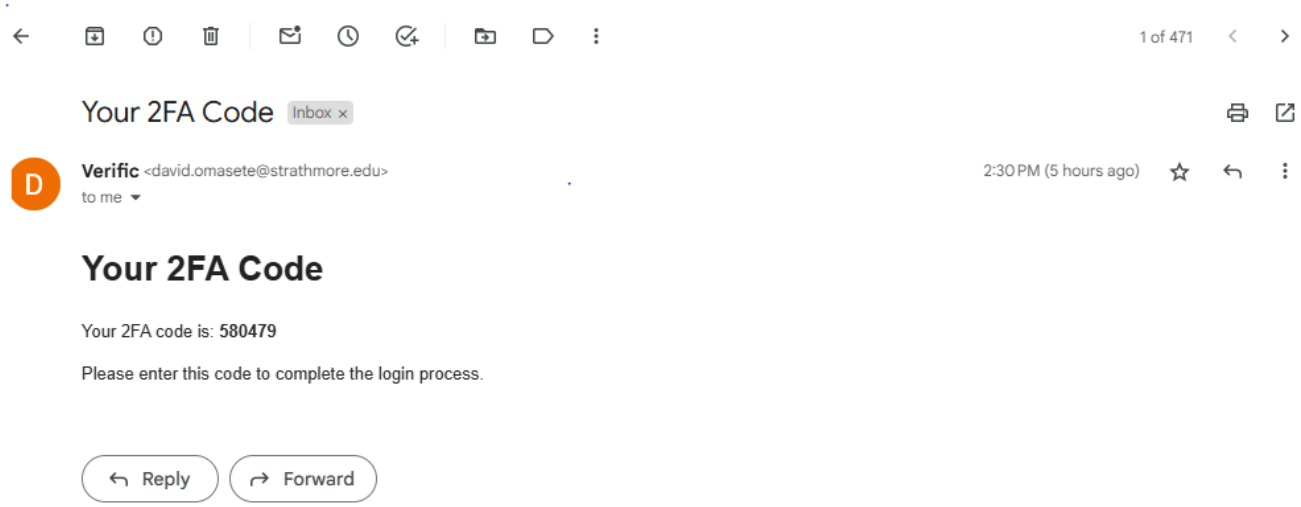
Annex 2L: Login Page



Annex 2M: 2FA

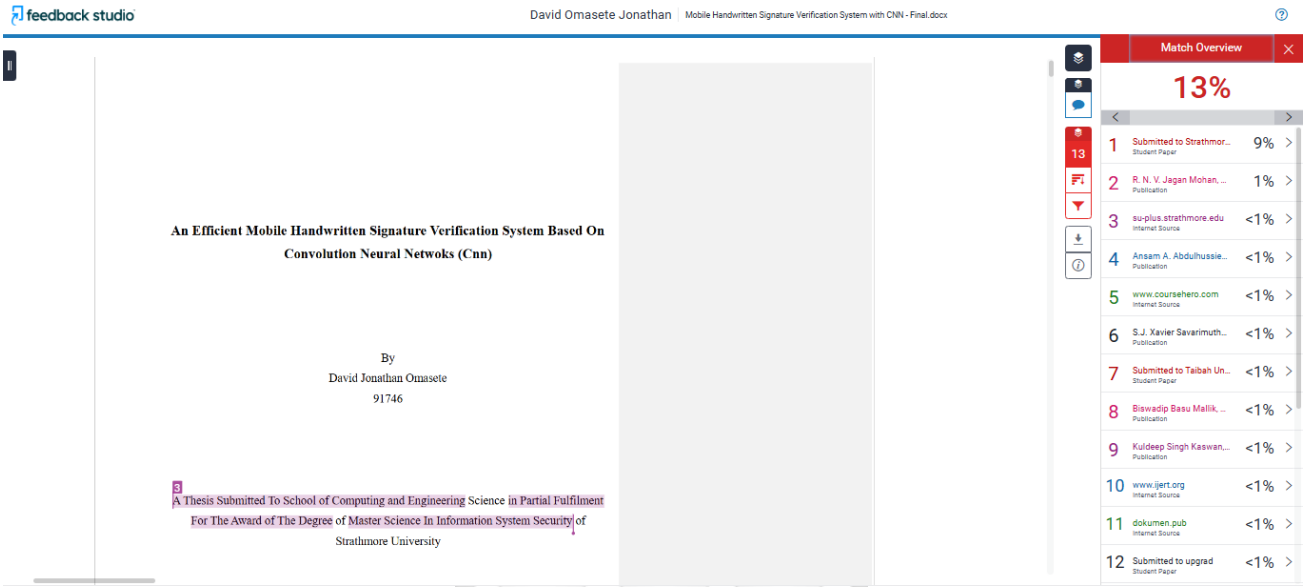


Annex 2N: MFA Validation



Annex 2O: MFA sent to user's Email

Annex 2: TurnItIn Report



receipt_Mobile Handwritten Signat

Mobile Handwritten Signat

Annex 3: Ethical Certificate

Completion of Online Research Ethics Review Submission

You have successfully submitted your application for ethics review "MOBILE HANDWRITTEN SIGNATURE VERIFICATION SYSTEM USING CONVOLUTION NEURAL NETWOKS (CNN)"

Certificate awarded to: Mr Omasete, David

Reference number: SU-ISERC2756/25

Date and Time: 2025-03-06 09:46:20

certificate_submission.pdf