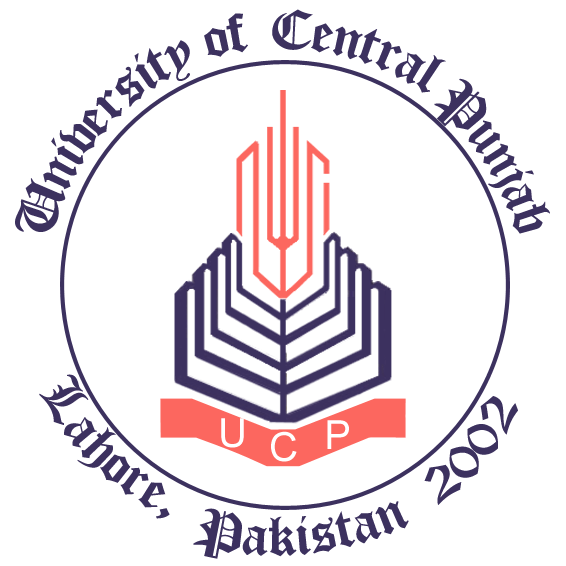
BSCS FINAL PROJECT

Software Design Specification

Offline Signature Verification System



Project Advisor

**Awais Muhammad Lodhi**

Presented by:

**Group ID: F22BS019**

Student Reg# Student Name

L1F19BSCS0570 Saleem Qureshi

L1F19BSCS0511 Abdul Rauf

L1F19BSCS0544 Zavial Sahar

**Faculty of Information Technology**

**University of Central Punjab**

Software Design Specification

SDP Phase II

Offline Signature Verification System

Advisor: Awais Muhammad Lodhi

Team # 19

|  |  |
| --- | --- |
| Member Name | Primary Responsibility |
| Saleem Qureshi | Research & Development |
| Abdul Rauf | Research & Development |
| Zavial Sahar | Research & Development |

Table of Contents

Table of Contents i

Revision History ii

Abstract iii

1. Introduction 1

1.1 Product 1

1.2 Background 1

1.3 Objective(s)/Aim(s)/Target(s) 10

1.4 Scope 11

1.5 Business Goals 11

1.6 Document Conventions 11

1.7 Miscellaneous 12

2. Overall Description 13

2.1 Product Features 13

2.2 Functional Description 13

2.3 User Classes and Characteristics 14

2.4 Design and Implementation Constraints 15

2.5 Assumptions and Dependencies 15

3. Technical Architecture 16

3.1 Application and Data Architecture 17

3.2 Component Interactions and Collaborations 21

3.3 Design Reuse and Design Patterns 26

3.4 Technology Architecture 26

4. Screenshots/Prototype 27

4.1 Workflow 27

4.2 Screens 28

4.3 Additional Information 32

5. Other Design Details 32

6. Revised Project Plan 38

7. References 39

Appendix A: Glossary 40

Appendix B: IV & V Report 42

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Abstract

The proposed project aims to develop a desktop-based application for the identification of original signatures to prevent forgery. The application will utilize advanced techniques in deep learning and image processing to train a model capable of offline signature verification. The model will be based on a modified version of the traditional twin Siamese network, which will input four genuine signature images and one test image to determine the authenticity of the test image. The system's performance will be evaluated and compared to existing models that compare a single genuine image with a test image. The goal of the project is to provide organizations with a reliable and efficient method for verifying signatures without the need for live identification.

# Introduction

## Product

Our end product will be a desktop app having deep learning model implemented in backend along with a research paper in which comparison of our model i.e., N-way Siamese network will be made with simple CNN and twin Siamese network. The desktop app will have user friendly interface and have functionality for user to upload scanned signatures and see the results that either the signatures are real or fake and for admin to add/delete a user, change credentials and other administrative rights. The model implemented in backend will be a python program written using deep learning APIs and libraries like KERAS and TENSOFLOW. The model will have multiple CNNs (5 in this case) connected with each other forming a network which is Siamese network. person authentication and verification using his/her signatures will be the main problem solved by our product. Our product will primarily be used by banks and organizations where signature verification is required, as our product will be a desktop application so it will be easily accessible to everyone and less costly compared to its counterparts like online signature verification system in which a specific equipment like stylus and tablet along with dedicated system are required. Also our system will perform better compared to other offline signature verification systems because of the model used in it

Fig1: Image showing test signs matching with signs in DB

## Background

A good amount of work has been done on signature verification, specifically using those datasets which we are going to use i.e., ICDAR SIGCOMP 2011 **[1]** and university of buffalo’s dataset CEDAR **[2]** both of these datasets are standard benchmark datasets and many people have worked on it we chose these datasets because of better community support and better results after training. For signature verification many models were developed and tested almost all of them uses neural networks like CNN for feature extraction, some researchers used simple CNN and some used other techniques and architecture like Siamese network **[3]**. We are also going to use Siamese network for signature verification but the uniqueness of model will be that we will be using N-WAY Siamese network for verification of signatures rather conventional Siamese architecture which is known as twin Siamese network **[3]** and have two identical CNNs having shared weight between them and if anyone have used Siamese network for signature verification in their project the either used 2- way Siamese network alone or combined with some other techniques like one-shot learning, writer independent and dependent and some make their project unique by changing and modifying distance functions for example some used Euclidean distance at distance layer some used cosine distance and some of them followed hybrid approach in which multiple distance calculating formulas/algorithms were used. We reviewed six research papers out of which 2 papers talked about signature verification using simple CNN **[4][5]**, 3 were about Siamese architecture for signature verification using one shot learning **[6]** and writer independent approach **[7]** and 1 paper talked about other machine learning techniques for signature verification **[8].** Also some similar projects were completed or are in process of completion in university according to the list provided by project office which are project 712, project 866 handwritten sign and digit verification which uses CNN, ANN and LSTM. Contrary to that we are going to use 5-Way Siamese network as our target audience are banks and most of the banks take 4 signature samples of a person while registering and whenever he/she have any query his signatures are matched with those 4 genuine signatures, automating this process is aim of this project for this problem we need an architecture which takes 4 genuine signature images and 1 test image which can either be fake or genuine and based on extracted feature from these images signature can be validated. The architecture and approach we are going to implement makes our project different from others



Fig 2: architecture of 5 way Siamese network

We have studies several research papers to understand the architecture of Siamese Network and its real-time implementation. A few of the research papers are discussed as follows:

* **Signet Convolutional Siamese Network for Writer Independent Offline Signature [3]**

This Paper is about a writer independent signature verification task with convolutional Siamese network. Siamese network is a twin network with shared weights, training is done by exposing this network to similar and dissimilar observations, minimizing the Euclidian distance between similar pairs and maximizing the Euclidian distance between dissimilar pairs experiments are conducted on signatures in various languages and writing styles. While verifying signatures smallest and critical details need to be seen

Signatures are most commonly used and accepted biometric to identify people, signature verification is of two type online and offline in online SVS special equipment are required but in offline SVS signs are usually captured by scanners or any other imaging device like mobile camera which is a 2Dimensional signature image there can be two scenarios in offline SVS writer dependent and writer independent where writer independent scenario is preferred, for a particular signer signatures are coupled as similar (Genuine, Genuine) and dissimilar (Genuine, Forged) pairs, so signature verification can efficiently be modeled by Siamese network which consist of twin CNN accepting two distinct sign images coming from either similar or dissimilar tuples. Two CNNS are joined by cost function computing distance metric between highest level feature representation on each side of network

**Preprocessing**

Images of same size are needed for batch training a Neural Network but signature images could have different sizes ranging from **153 X 258** to **819 X 1137,** first all images need to be resized to a fixed size which is **155 X 220** in this case after resizing images are inverted using bilinear interpolation so background pixels have 0 values, further images were normalized by dividing the pixel value with standard deviation of the pixel value of image in dataset.

This paper also tells about CNN and its architecture. Deep CNN is a multilayer Neural Network consisting of several Convolutional layers having different kernel sizes interleaved by pooling layers which summarize and down sample the output of its convolutions before feeding to next layer kernel sizes ranging from 11 X 11 to 3 X 3 were used in this project, given a differentiable loss function, weight of different layers are updated using back propagation

**Working**

The twin CNN have the same configuration with the same parameter and shared weights, parameters updating is mirrored across both subnetworks, these subnetworks are joined by loss function at the top which computes a similarity metric involving Euclidian distance between, feature representation on each side of Siamese network.

**Loss**

Contrastive loss is a type of loss function used in Siamese network once a high level feature output from both of the constituting CNNs is obtained they are joined with a loss function which is

In which L is a function of S1, S2 and y where s1 and s2 are the two samples (signature images) y is a binary indicator denoting whether two samples belong to same class or not α and β are constants and m=1 is margin

is Euclidian distance computed in the embedded feature space f is an embedding function that maps a signature image to real vector space through CNN and w1, w2 are learned weights for a particular layer, due to the loss function selected this space will have a property that images of same class will be closer to each other than the images of different classes i.e. forgeries or signature of different writers, both branches join together by a layer that computes Euclidian distance between the two points in embedded space in order to decide that two images belong to similar class or dissimilar class then threshold value on distance need to be determined. For convolutional and pooling layers CNN architecture is used the size of the filter is N X H X W where N is number of filter, H is height and W is width

**Dataset**

4 widely used benchmark datasets were considered including **CEDAR, GPDS300, GPDSsynthetic and BHSig290 signature corpus**. To evaluate signature verification algorithm. CEDAR contain signatures of 55 signers having 24 signature signed 20 minutes apart so total 1320 genuine signatures and equal number of forged signatures images are available in Gray scale mode in the dataset. Randomly M signatures were selected from K for testing and remaining K-M for testing purpose

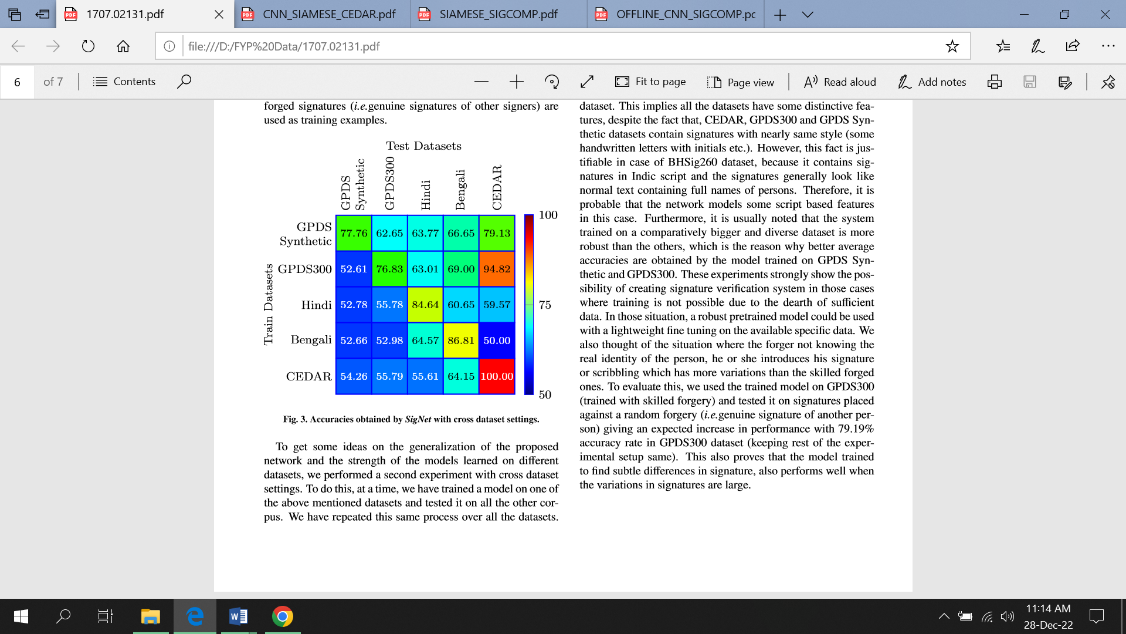
**Result**

A threshold d is used on the distance measured D(Xi, Xj) to decide whether the signature (i, j) belongs to similar or dissimilar class sign pair (i,j) having same identity are denoted by Psimilar and having different identities are denoted by Pdissimilar then set of all the true positivesand true negatives

True positive rate and True negative rate

And final accuracy

100 percent accuracy was achieved when CEDAR dataset was used ,77.76 percent accuracy when GPDS synthetic dataset was used and 63.36 percent accuracy was achieved when GPDS 300 was used



* **Offline Signature verification using Convolution Siamese network [4]**

This paper presents a method of signature verification using convolutional Siamese network, a deep learning base framework combining feature extraction and metric learning

**Introduction:**

This system verify person’s identity based on his chosen signature it classifies person’s signature sample genuine or forged. In offline signature verification signature is captured after the completeness of writing process, and represented as digital image, it is similar to other verification problem like face verification or person identification

**Working:**

Recent work explored different features like contours feature which encode directional property of signatures contours and length of regions enclosed inside letters and curvlet transformation which uses energy of curvlet coefficient computed from whole hand written signature.

In metric learning there are two approaches writer dependent and writer independent. A specialized metric model is learned for each individual, during training phase using person’s genuine or both fake and genuine signatures. Combining feature extraction and metric learning provide better performance. Siamese network is used in this problem, which assess similarity of *2 signature image in pair* for offline signature verification and combine the separate modules together i.e. feature extraction and metric learning

**System Design/System Working:**

In pattern recognition input of Neural Network is sample signature and the output is predicted label, but for signature verification, this style of Neural Network is not suitable so Siamese network is constructed including two sub CNNs which take a pair of sample as input and gives label as output, the input images in the pair are first preprocessed to same size and fed to CNN in Siamese network for feature extraction after that metric method is use to calculate distance of extracted feature.

**Dataset:**

Two publically available standard benchmark datasets are mentioned in this paper i.e., **CEDAR** and **GPDSsynthetic,** CEDAR contain signatures of 55 individual people signed **24** signatures each creating **1320 genuine** signatures having equal number of forged signatures. The images in CEDAR are available in grayscale mode, GPDSsynthetic is based on synthetic individual protocol dataset comprises of **4000** signers every individual having **24** genuine and **30** forged signatures making **96000** genuine and **120000** forged signs

**Preprocessing:**

There are differences between signature images from both datasets, the signature images in both datasets are more skewed so more steps are required to preprocess the signature images in CEDAR compared to GDPSsynthetic for CEDAR images **Otsu’s method** is used to binarize image which is a threshold selection method from gray level histogram, to clean grain noise **connected component analysis** is used, **Hough transform** is used to detect and remove lines which are not part of signature. After preprocessing CEDAR images signature region from images were extracted and it was found that signature images from both datasets have variable size but to train CNN all input needed to be of same size so images were normalized, for **CEDAR** signatures were centered in canvas of size **1060 X 450** pixels and for **GDPSsynthetic** signatures were centered in canvas of size **2800 X 1200** pixels then images were resized to fixed size of **224 X 512** pixels

**Feature extraction:**

Features are extracted by Siamese network containing two identical subnetworks with branches sharing same parameters features were extracted through CNN using automatic feature learning, the first layer in network is convolutional layer with **11 X 11** kernel then there are two parts one part comprises of **5** convolutional layer and **4** maxpooling layer second part make use of low level features, feature map size after first convolutional layer is **W X H** where **w** is width and **h** is height of feature map

**Metric learning:**

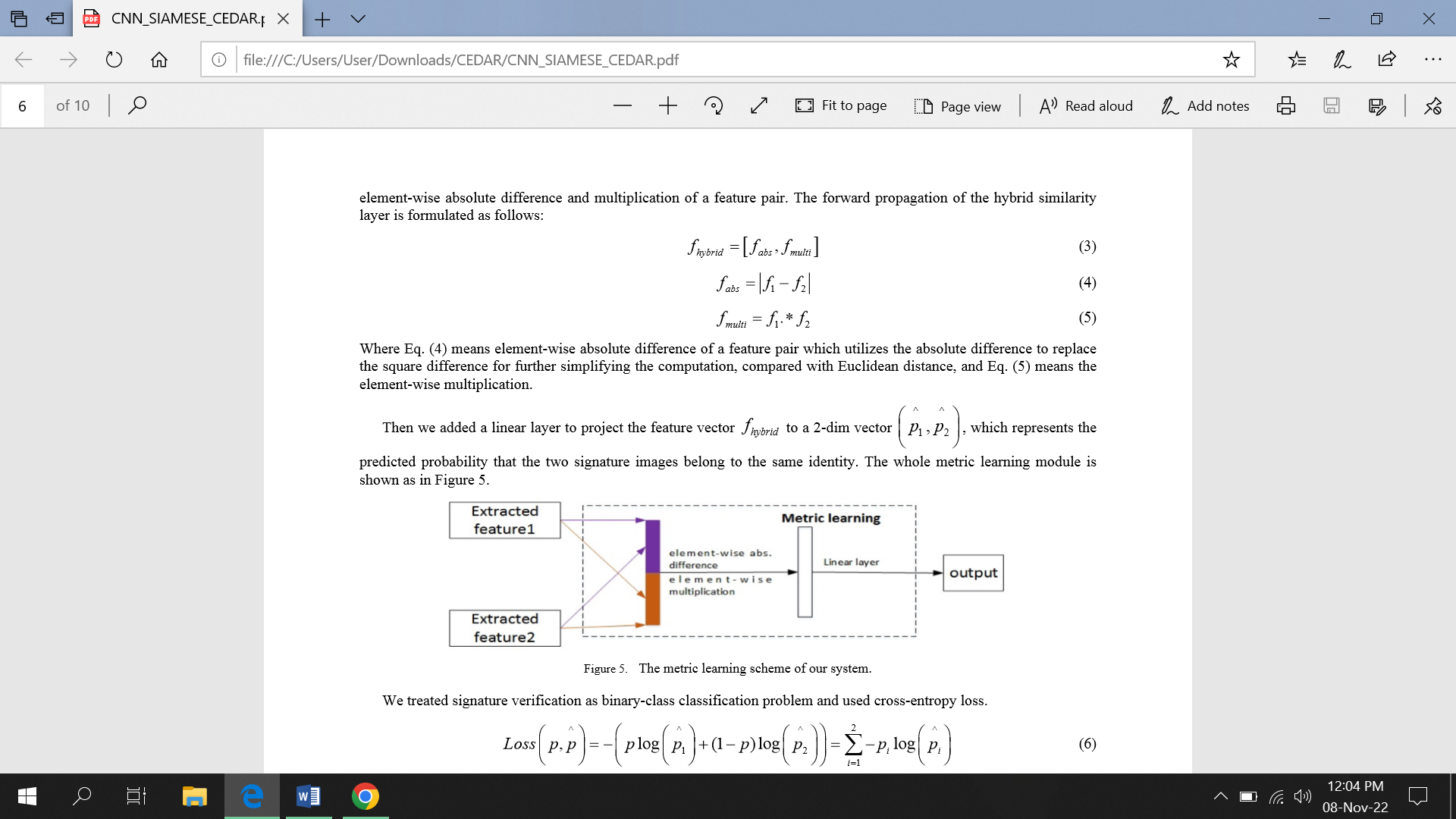
If features extracted from CNN are **f1** and **f2** now similarity need to be evaluated between these pairs f1 and f2 to measure similarity between feature pair **Euclidian** and **cosine** distances are used

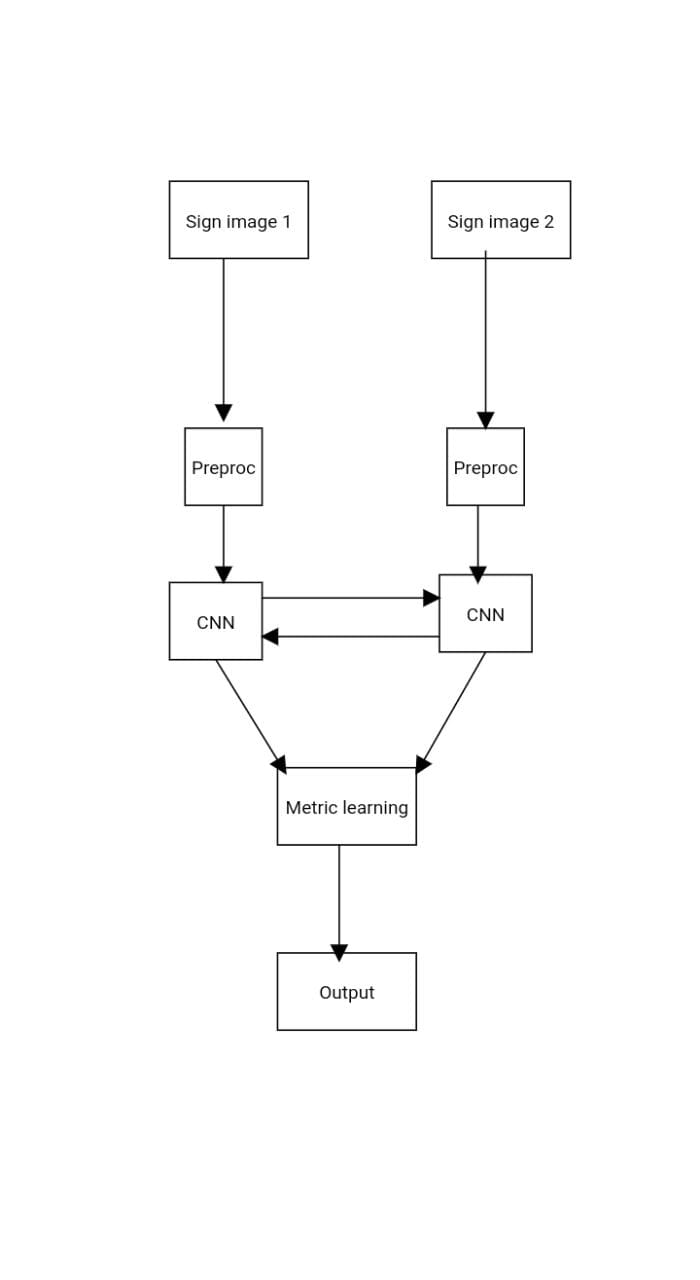
a hybrid similarity layer combining both Euclidian and cosine distance is formulated

where abs mean absolute, multi means element wise multiplication a 2d vector **[p1, p2]** representing predicted probability that 2 signature images belong to same identity or not

Signature verification is considered a binary class classification and use cross entropy loss

*p is target class and p hat is predicted probability if image belong to* ***same signer p1=1, p2=0*** *else* ***p1=0, p2=1***



****

**Result:**

in GSDPsynthetic there are 24 genuine and 30 forged signers so **276** genuine genuine pairs and **720** genuine forged pairs and totally there are **2000 x 2 x 276 = 1104000** signature pairs for training and testing

in **CEDAR** there are 1320 genuine and 1320 forged signs of **55** signers **5** signers are randomly selected for testing purpose and rest 50 for training every individual have 24 genuine and 24 forged pairs so total **276** genuine genuine pair and **576** genuine forged pair compared to all algorithms trained on GSDPsynthetic this combined algorithm performed better

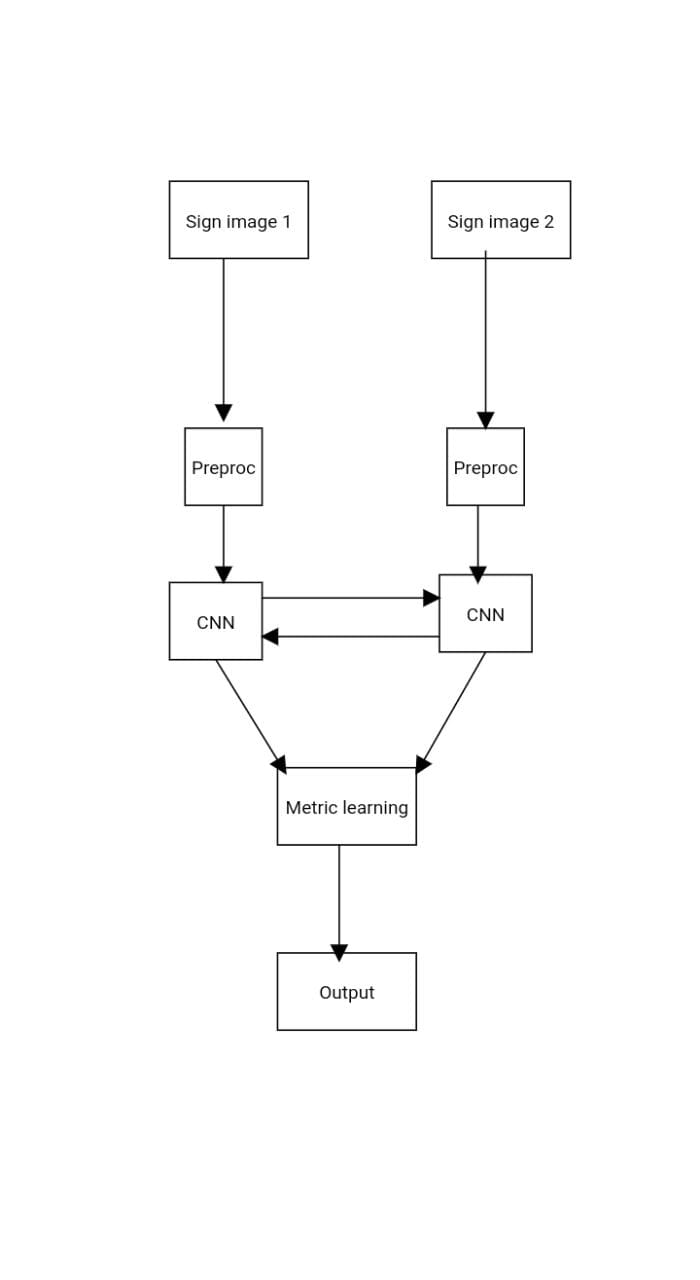
*EER USING DIFFERENT DISTANCE ALGO*

|  |  |
| --- | --- |
| ***METRIC METHOD*** | ***EER*** |
| *COSINE DISTANCE* | *20.46* |
| *EUCLIDEAN DISTANCE* | *12.30* |
| *HYBRID SIMILARITY LEARNING* | *10.37* |

*EER RESULT Of sampling GPDSsynthetic*

|  |  |  |
| --- | --- | --- |
| ***SAMPLING ON TRAIN*** | ***SAMPLING ON TEST*** | ***EER*** |
| *NO* | *NO* | *10.65* |
| *YES* | *NO* | *10.87* |
| *YES* | *YES* | *10.37* |

**Block diagram of Project:**



## Objective(s)/Aim(s)/Target(s)

The objectives of this project are making a deep learning model having N-way Siamese network implemented for signature verification, our main aim is to develop an application for organizations to automate the process of signature verification specifically in banks. We will be using modified version of Siamese network **[4]** which will have multiple CNNs implemented in it along with our final product we will also be conducting research, comparing our model with conventional models for signature verification and performance of the model under different circumstances.

Once our project is completed we will have a 5-way Siamese network model implemented along with our research work.

The research work will benefit researchers working in field of deep learning and machine learning, beside signature verification this model can be used in various problems like facial recognition, image matching etc., providing better accuracy.

## Scope

Every application and model have their scope defined and in that scope they prove to be useful for humans, beside the use of that application or model there are also some limitations so is with SVS.

Areas where our product will provide benefit to the people or organizations are:

* It will help organization automating the process of signature verification
* Our product will be able to identify signatures more accurately than humans as computer can keep track of minutest details like angle and strokes which naked human eye can’t
* As we are working on offline signature verification so this will be less costly compared to other digital signature verification methods which require special equipment, this model will only use image processing and deep learning techniques
* We are going to implement a new model which is advancement of simple CNN so it will give better accuracy with better learning outcomes
* At the end comparison will be made with models having single CNN implemented

The limitation of our project are:

* As this product is for signature verification and sometimes only signatures are means of protecting confidential information and assets of a person so a small mistake by system can have extreme consequences
* Sometime Signature verification systems are unable to catch forged signatures done by skilled people

## Business Goals

The primary objective of this application is to automate the process of manual signature verification, with the aim of increasing precision and ease of use. Signature verification is a commonly used biometric method, unlike other biometrics like fingerprints and retinal scans, signatures can be easily forged and detecting forged signatures can be a challenging task. By providing an automated solution, this application aims to assist banks and other organizations where signature verification is required. Given the widespread use of signature verification, this application has the potential to generate significant revenue through targeted marketing to such organizations.

## Document Conventions

* Referencing style **IEEE** and **APA**
* Font **Times New Roman**
* Headings **BOLD**
* Size of headings **12**
* Size of font **12**
* Captions of Images and diagrams= Times new roman italic size **12**
* Line spacing **1.5**

## Miscellaneous

This application is designed for use by specific organizations and their registered employees. The administrator of the organization will be responsible for registering employees as users, who will then have access to the app and be able to perform designated tasks, such as verifying signatures of clients. This is distinct from a consumer-facing application that is open to registration by any individual.

# Overall Description

## Product Features

Our product/Desktop app will be able to take scanned signatures as input. There can be two type of users, Admin users and simple users (i.e., those users who are registered by the admin and are employee of organization). Firstly, user will login to his/her account by entering username and assigned ID. If the user is also an Admin, he/she will also have administrative rights like registering a user, deleting existing users, registering new clients or deletion and updation of old client’s data. If the person is a simple user, let say cashier of the bank. He can sign in to his account by username and id. After signing in, he will have option to upload a test signature image along with the registration number and name of that client and will also have option to see the genuine signature images of that client. Once the image is uploaded, it will be given to the back-end model, where features will be extracted and decision will be given based on extracted features. User will then be able to see that either the signatures are fake or real.

At the back end of the app Deep learning model will be implemented which will use CNN to extract features from signature images and then match them to see the similarity. Siamese architecture will be used in this problem. Siamese architecture gives better results in verification problems like face matching, signature verification etc., so we will be using Siamese architecture for this problem traditionally Siamese network have two subnetworks but we will be modifying that architecture and will make it 5 way architecture or N-ways Siamese network which will have 5 subnetworks (CNN) in which 4 genuine signatures and 1 test signatures will be given and decision of signature originality will be made.

## Functional Description

We will have two type of users i.e., Admin user and Regular users. Regular users will be registered by the Admin. Functionalities performed by the system when users interact with the system are as under:

**Regular User:**

* The user will upload scanned signatures to the system
* In response system will give an outcome either positive or negative

**Admin User:**

* Admin will able to register/ add a person he may be a user or client**.**
* Admin will be able to add new record i.e., 4 genuine signatures of a client
* Admin also have right to upload a Test signature image
* Admin will be able to delete a person
* Admin will be able to change credentials of a user
* In response of every Admin request system will show results in response

## User Classes and Characteristics

The classes and characteristics in this case will be:

* **Regular User**

Most of the time these users will be interacting with the system the users will be registered by admin and will have following characteristics

* + User will have an id assigned by organization
  + Will be able to upload signatures to be verified
  + Will be able to see the result of uploaded signatures
  + Will be able to see the original signatures of the customer
  + Will be able to see the list of registered customers
* **Admin user**
  + Admin user will have administrative rights
  + Admin user will be able to register a new user
  + Admin user will be able to delete a user
  + Admin will be able to add, update and delete a customer’s data
  + Admin will be able to see the registered customer and its data
* **Customer/Signer**
  + Customers are not users of the system
  + Customers will be registered by providing 4 genuine signatures
  + Customer will provide a signature image which will be tested against original signature

## Design and Implementation Constraints

The quality and availability of the dataset is a crucial factor in the successful completion of this project. The requirement to identify signatures with varying nib types, sizes, orientations, angles, and colors presents a challenge in obtaining a suitable dataset. Despite utilizing two standard datasets, SIGCOMP 2011 **[1]** and CEDAR **[2],** for training the model, the desired level of diversity in the signatures may prove difficult to obtain. Additionally, the limited time frame for publication of a conference paper and practical implementation of the project adds to the pressure, as this project involves both research and development aspects.

## Assumptions and Dependencies

The present project aims to develop a desktop application and an algorithm, utilizing the N-way Siamese Network Convolutional Neural Network (CNN) model as the architecture. The programming language chosen for both the algorithm and desktop application is Python, which provides access to a variety of libraries such as NumPy and Pandas for data processing and calculation and OpenCV for image preprocessing. The utilization of Python and its libraries allows for a seamless integration of the various components and efficient implementation of the algorithm and application.

# Technical Architecture

Our system will be a custom built system COTS as it will target a specific audience i.e., banks and other organization which requires signature verification.

There will be certain steps included in signature verification which are data acquisition, preprocessing, feature extraction and verification the processing type will be online processing as user will provide test signature at the runtime which will be matched with 4 genuine signatures in the database and decision will be made and signatures will be validated the current system will contain user id, user credentials, customer id, customer credentials and 4 genuine signatures in database and will be able to take new signature image which will be of test signature as input. Client server model will be followed in this problem where signatures and other information of users will be stored on the server. The client can then send a request to the server with the signature to be verified, and receive a response indicating whether the signature is valid or not. This allows for efficient processing of large amounts of data and reduces the load on the client side. The current system will be using python as the main programming language for algorithm development as well as desktop app development deep learning api and libraries like Keras, Tensorflow, opencv, skimage, numpy, pandas will also be used.

As our main audience is banks and organizations and user will be registered employee of that organization let say cashier of a bank most of the time these people don’t have technical background and are unware of operating systems and other it related stuff, mostly windows OS is used by banks and other organizations so we will be making a Windows compatible desktop app, the app will also run on OS like Linux and mac but main target will be windows OS at the backend.

Signatures are the most widely used biometrics worldwide and signature verification is a very sensitive task where one mistake can lead to worst consequences as signatures are the way to protect one’s confidential information and assets so an ACID compliant system will be required hence we will be using relational database at the back end to store the signatures. As the system is for organizations so multiple systems may be connected through LAN network or internet can also be used to connect systems.

## Application and Data Architecture

There will be several components in SVS application which will be able to take and pass data and will able to perform some functionality. User will interact with the model using desktop app so a good GUI will be made. There will be a module to take input of signatures and also a separate module to verify signatures and tell the results also there will be a database where genuine signatures of a person are stored.

At first user and clients will be registered by admin once the user is registered by admin they will be able to log in to their account by entering their assigned id, once signed in they will then be able to upload test signatures of a client to match it with genuine signature images in database based on that result he will then allow or reject the customer’s query.

The diagrams given below will further help in understanding of application data architecture



Fig 3: *ER diagram showing interaction between user and component of system*

In the above given ER diagram there are 5 entities namely Bank, Employee, Account, Client and Signatures where signature and account are weak entities as signature are dependent on customers and account is dependent on both bank. SVS can be used for any organization for signature verification but in this case and in ER we are specifically targeting bank as our audience so organization in this case will be bank. In above given diagram there are certain relations between the entities i.e., 1 bank can have multiple employees also 1 bank can have multiple accounts similarly 1 client can have multiple accounts but one account cannot be allotted to more than one person, 1 client can have multiple signatures i.e., he can do signature in various styles.

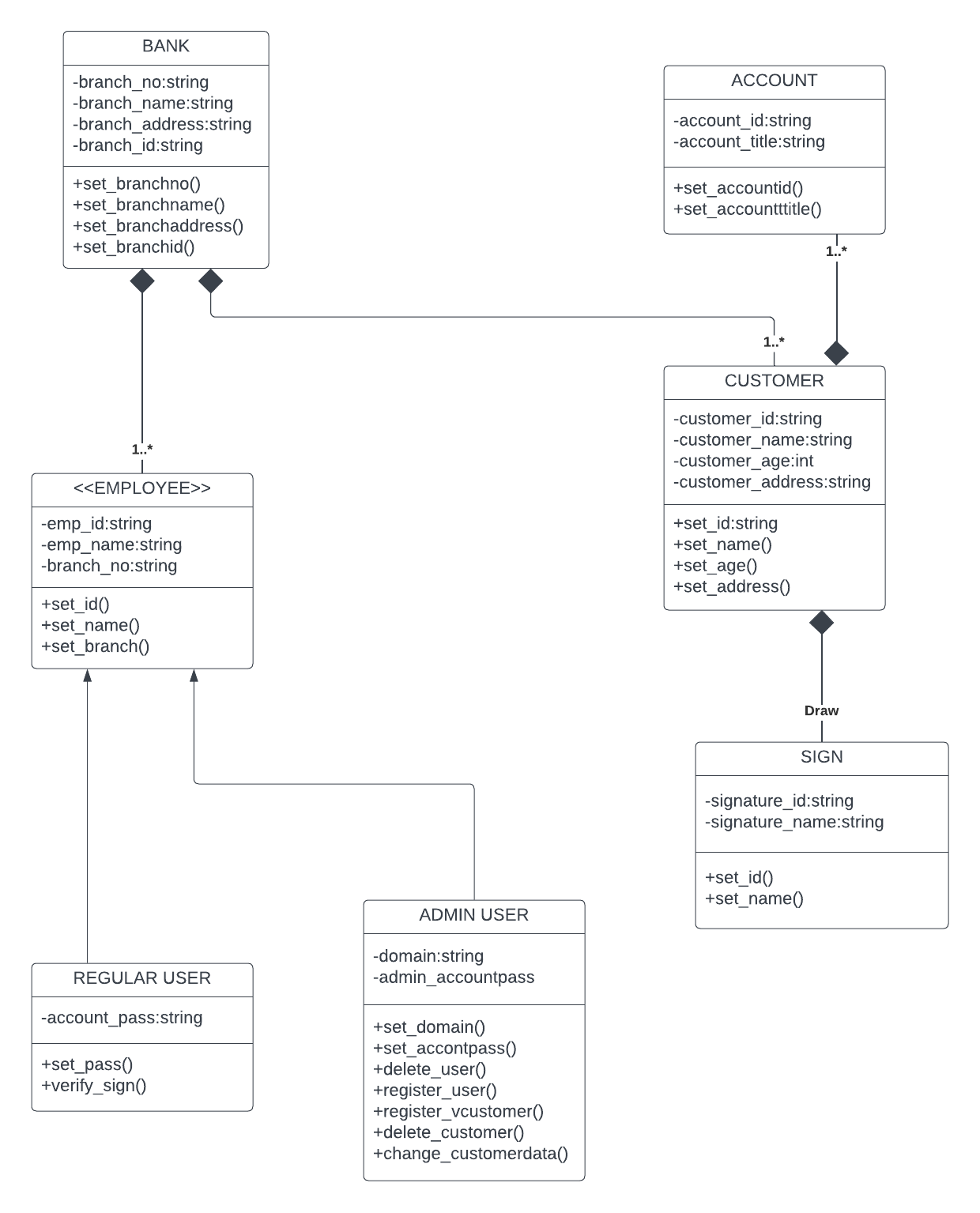


Fig 4: *Class diagram of the system*

Above is the class diagram of the system we have identified 7 main classes from the system which are:

1. Bank
2. Account
3. Employee
4. Customer
5. Regular user
6. Admin user
7. Signatures

There are also some attributes and methods related to each class for example employees and customers have name, id etc., and also there are some special methods related to each class for example admin class have methods which allow it to add/delete a user or customer and defines other administrative rights similarly Bank have some methods defined like branch no branch address etc., and their modification so like these classes all other classes also have their attributes and methods defined relations between these classes are also defined i.e., account is related with bank and have association in relation with the bank there is no composition between bank and account because of account can be online or in a virtual space but bank in this case is a physical entity so account does not depend on bank for its existence that’s why there is simple association between them also account is related with customer and have composition relation with the client because client is the customer of the bank either he prefer online banking or offline banking he is a customer and that specific bank account is associated with that customer once the customer leaves the bank i.e., he is no more the customer of the bank his bank account is also deleted when he leave so it’s a composite relation also customers have composition relationship with the bank i.e., people can remain customer of the bank as long as it exists, so as long as bank exists till then its customers will exist and similar is the case with employees further there are two more classes inherited from employees i.e., employee can be of two types the regular employee and Admins, Admins are the people who have administrative rights and are able to make changes at administration level like registering or deleting users and customers updating users or customers credentials etc., and regular users are the one who will be using our system for signature verification.

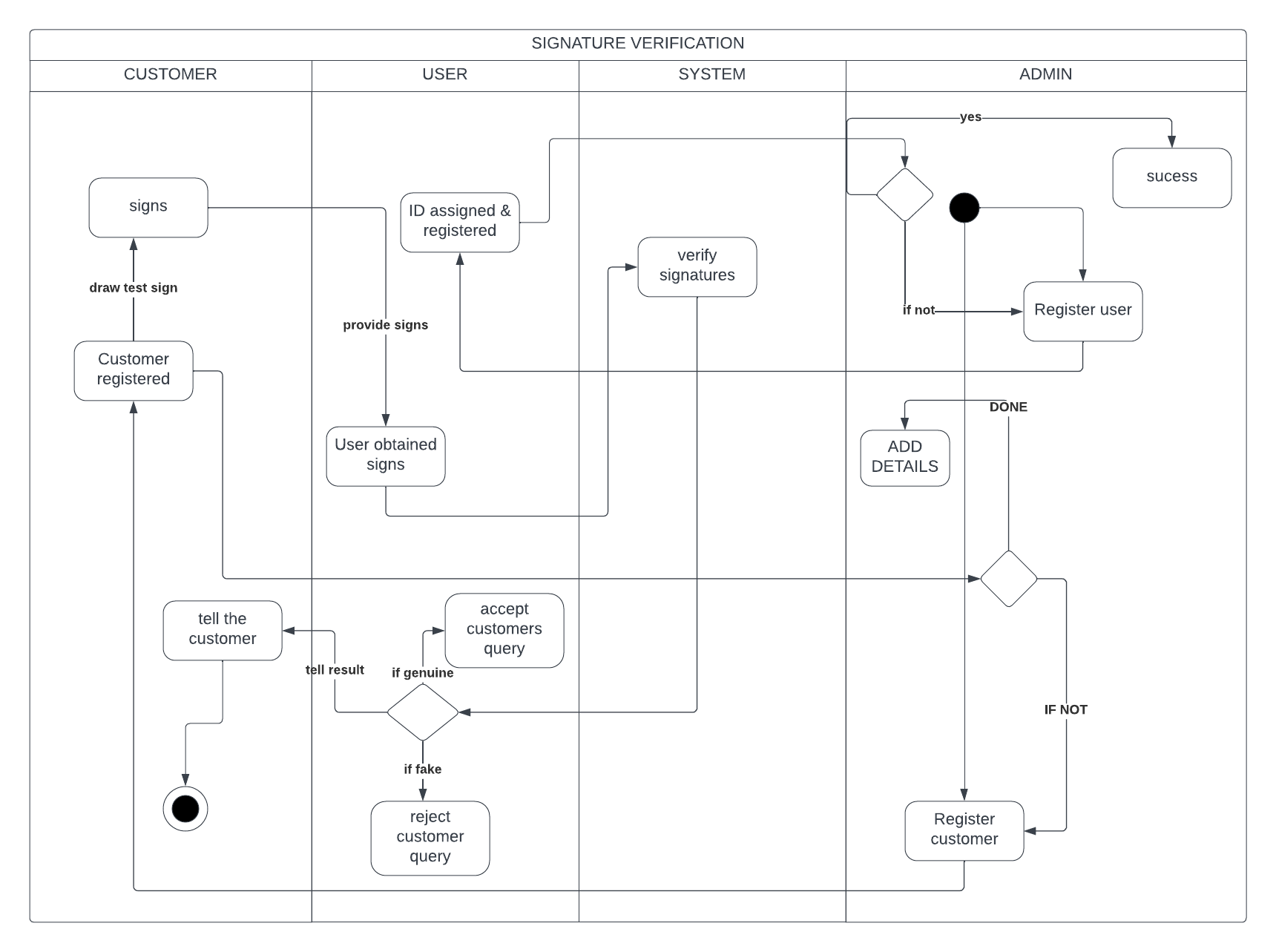


Fig 5: *Activity diagram of the system*

Above is the activity diagram of the system in our system the whole activity will be started from Admin’s side which means that he will be responsible to add a customer or a user so whenever a customer first interact with the system he got registered into database by the admin along with his 4 genuine signature images and other credentials and an id is assigned to the customer similarly employees are also registered by the admin which then become user of our app once a customer is registered that registered person will provide the registered user with the signs now task of user is to identify that either the provided signatures are real or fake he then give those signatures to the system where the verifier has been implemented than after processing the signatures system tells that either the signatures are forged or real in either cases results are given to the user and based on that results user either rejects customer’s query or accepts customer’s query user also tells the result to the customer and after that systems terminates.

## Component Interactions and Collaborations

In a system there are several components which interact with each other to make the system work we also have several components in our system which will collaborate and interact with each other to make SVS work their interaction and collaborations are described by the diagrams below:

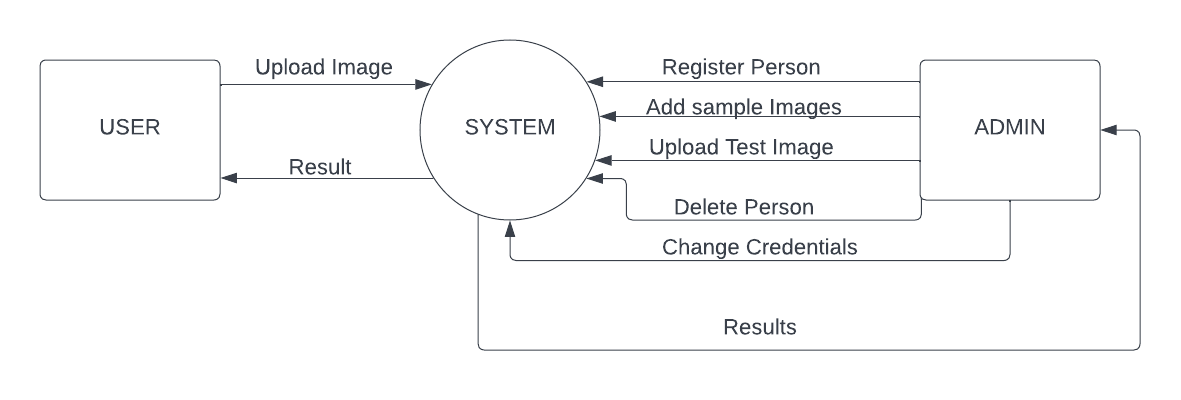


Fig 6: *Context level diagram of the system*

The context level diagram of system shows the interaction of regular users and admin users with the system, unlike user and admin customer cannot directly interact with the system the interaction between system and other component is as follows:

1. Admin requests system to register a person
2. Admin can request system to delete a person
3. Admin can request the system to change credentials of a person
4. In response of all the requests made by admin side system will be showing appropriate result to admin
5. The other component interacting with the system is user, user can also make certain requests to the system
6. User can upload image of test signatures to the system
7. In response to that uploaded sign image user will be able to see that either the signatures are fake or real

Admin can also act as a regular user hence he can also upload test sign image to the system and see the results.

When admin is registering a person a person can either be client or employee if he is registering a client he will also add 4 genuine scanned signatures of images to the system or database.

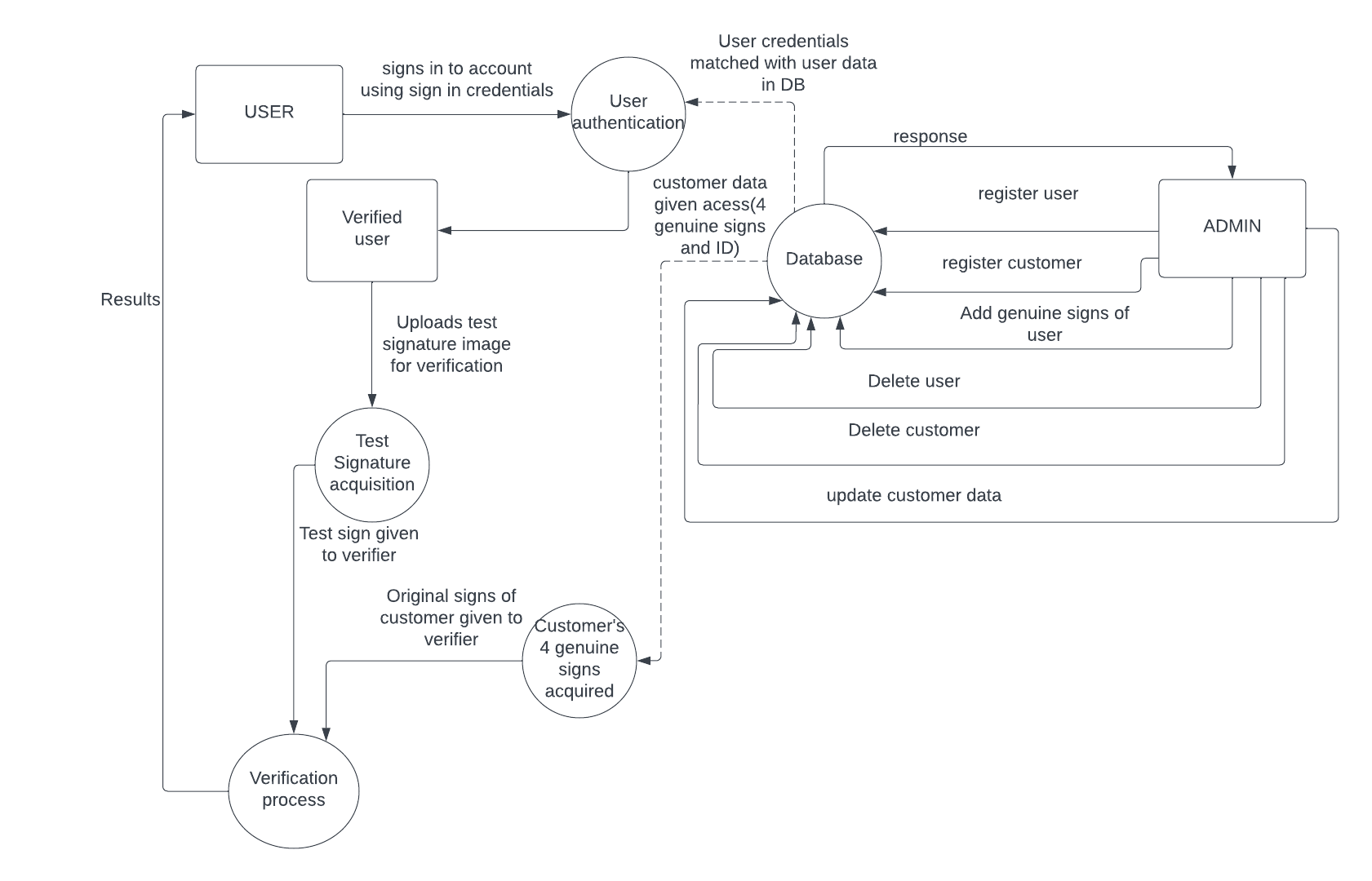


Fig 7: *Level 1 DFD of the system*

Above is the in depth DFD of the system the tasks performed by users and admins are same as they were in context level DFD but this diagram shows the levels more clearly and more detailed. Admin will add & delete users and customers and will also be able to update customer’s data and user credentials which were stored in database. Once the user is registered, he will then be able to test signatures. First user sign in to account by entering his username and id once a user is verified then he upload test signature image once test signatures are acquired from user, 4 genuine signatures are also acquired from database then these signs are given to verification process after which user will be able to see the results and accept or reject customer’s query.

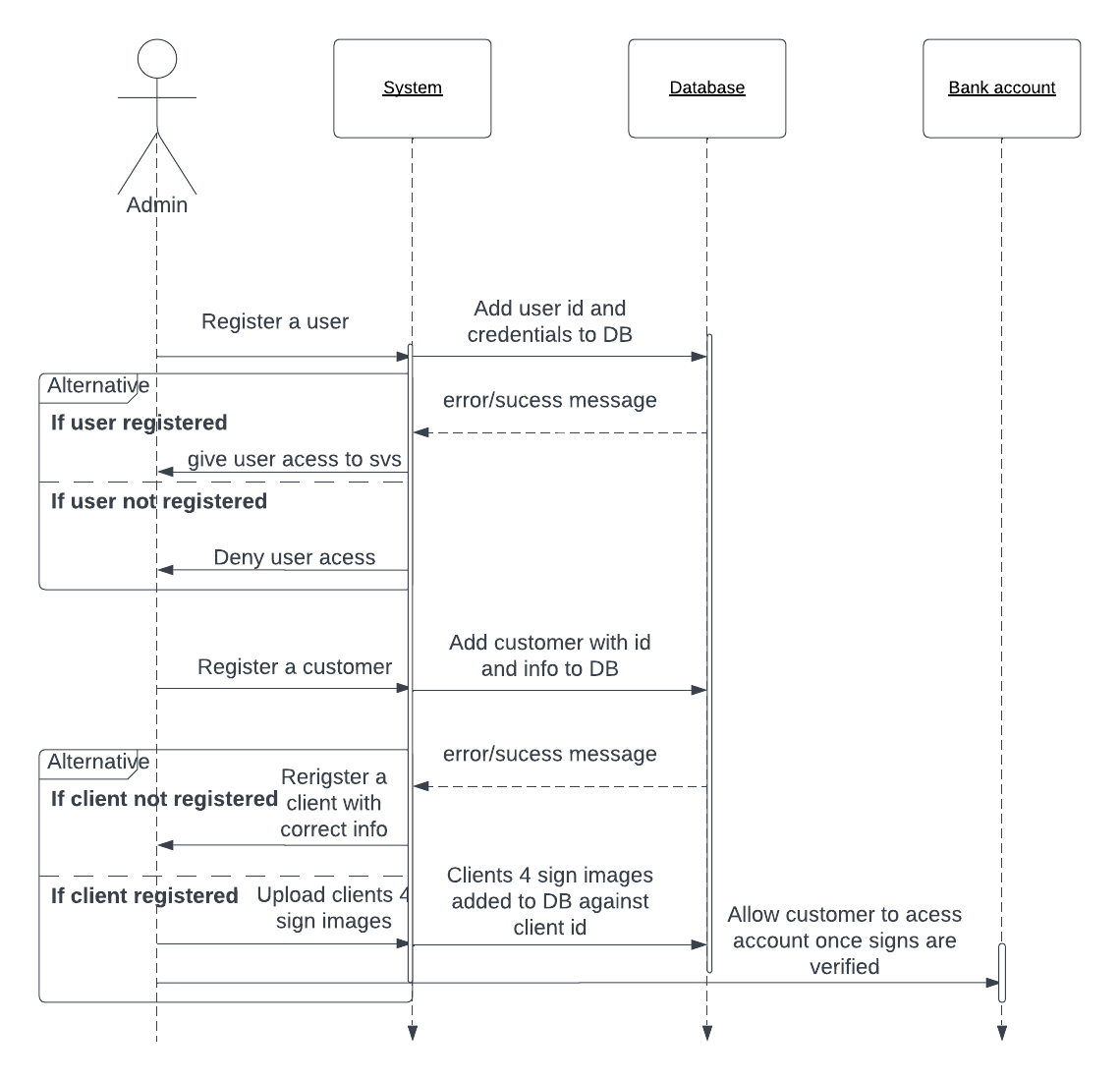


Fig 8: *Sequence diagram showing admin’s interaction*

Above is the sequence diagram showing the interaction of the Admin with the system and response back from the system admin send user registration request to the system then the user credentials along with id are stored into database when user is stored into database in response there is an error or success message from the database toward the system, if the user is added in database system will tell admin to allow that user access else deny. Similarly, admin can register customer and then get error or success response from DB then if a customer is successfully registered admin will add his 4 genuine signature images along with other credentials in database and will also give access to the customer’s account after signature verification.

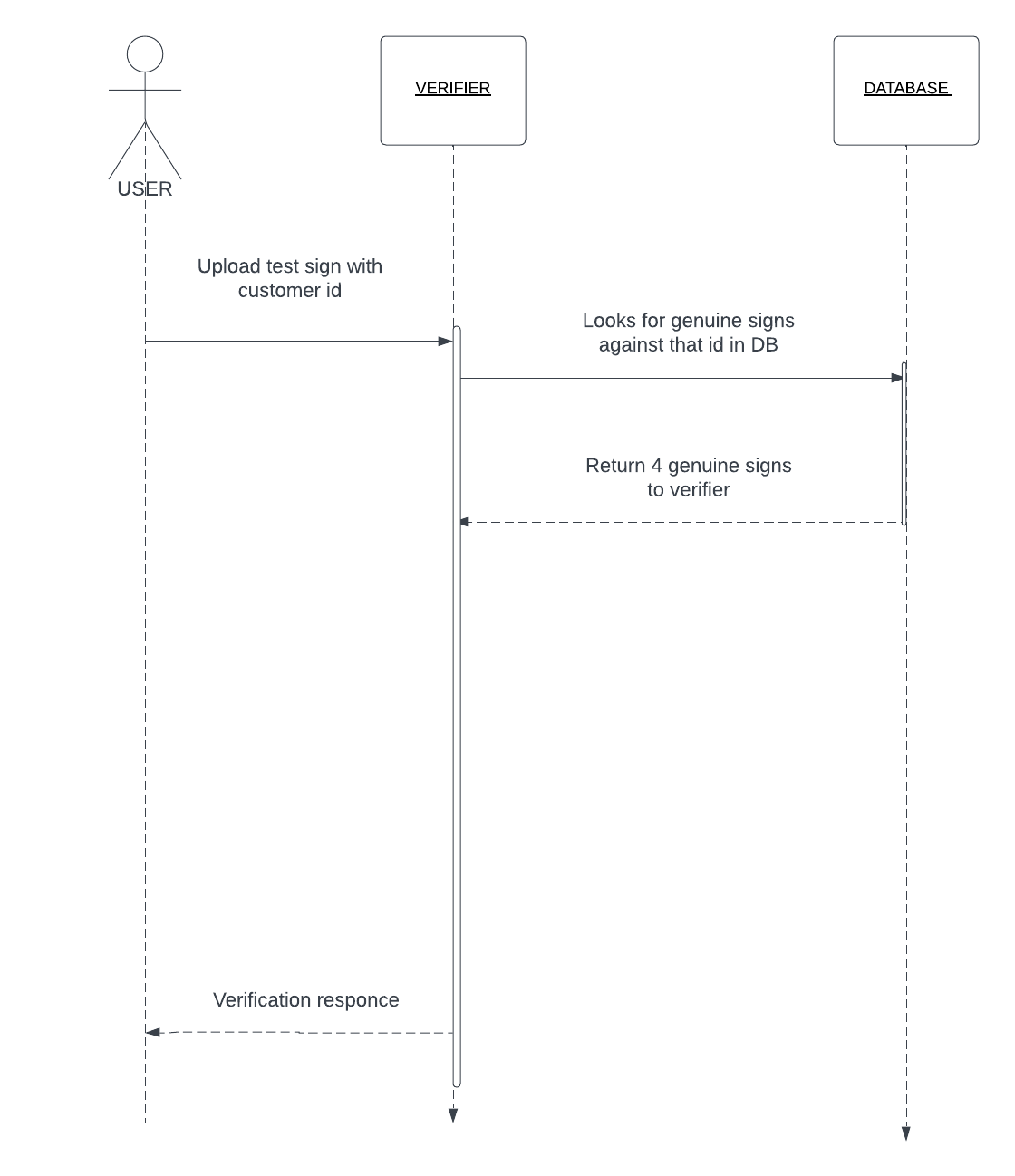


Fig 9:  *Sequence diagram of user’s interaction with system*

Once the user is registered he is then able to verify signatures user gives the test sign image along with id of the customer to the verifier then it looks for the 4 genuine signature images of that customer in database, if genuine signs exist in database DB return those 4 signs to verifier as a response. Verifier then runs algorithm on the images i.e., genuine images obtained from database and test image provided by the user, if the signs are genuine user is directed to accept the customer’s query else if signs are forged he is directed to reject customers query.

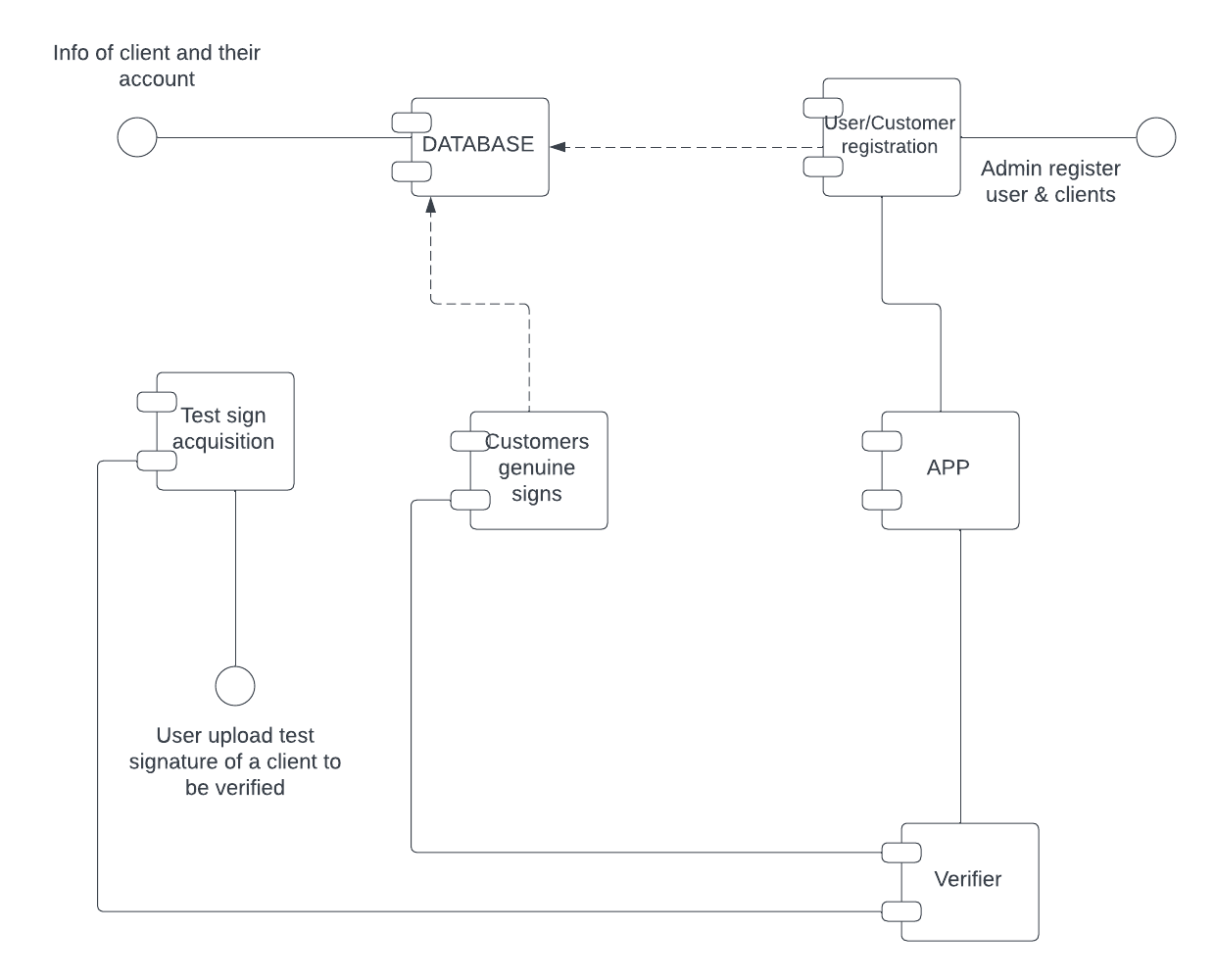


Fig 10: *Component diagram of the system*

The components in the system are:

1. Database
2. Test sign acquisition
3. Customers genuine signs
4. User and customer registration
5. Verification process or Verifier
6. APP.

Information about clients and their accounts are stored in database. Admin registers user and clients into registration section of the SVS app which is connected to the app test sign acquisition is a component which takes the test sign image uploaded by user and is connected to verifier also there is a component customer genuine signature which takes 4 genuine signature images of customer from database and is also connected with verifier, verifier verifies that either the signatures are forged or real and is connected to the app.

## Design Reuse and Design Patterns

Observer design pattern will be suitable in this case as it allows loosely coupled communication between objects and change in one part of the system can be notified to other parts of system more easily. In this case signature verification process and result showing will be observers and signature of the customer will be subject then the subject will notify the observers about changes in the signature and observer will perform required action to verify signatures and tell results observer design pattern can be applied in this problem as follows:

1. Customer provide user (employee of bank) with signatures
2. User upload test signature to the system desktop app
3. Test signatures are subject now whereas comparison process and display of results become observers
4. Subject tells observer if there are any changes
5. Observer compare test signatures with genuine signatures stored in database
6. Based on the comparison, the observers determine if the signature is real or fake.

Also a design architecture name MVC (Model View Controller) can be used to implement this in this architecture application is divided into three main components model, view and controller. The model contains customer signature and comparison process to verify signatures, viewer represent the user interface and display the result of comparison, controller act as a bridge between model and view and handle user interaction and updates model based on user input

## Technology Architecture

Following will be the anticipated infrastructure that will be required to support application and information architecture:

* The platform for this app can be windows or macos which are commonly used by organizations
* The application will be hosted on a local server or pc as it is an offline desktop app
* As this an offline app so internet will not be required
* The application will require a good amount of storage to store customer’s information and signatures
* As image processing and verification is required so a good amount of processing power is required
* For efficient performance of comparison operations good amount of memory will also be required

# Screenshots/Prototype

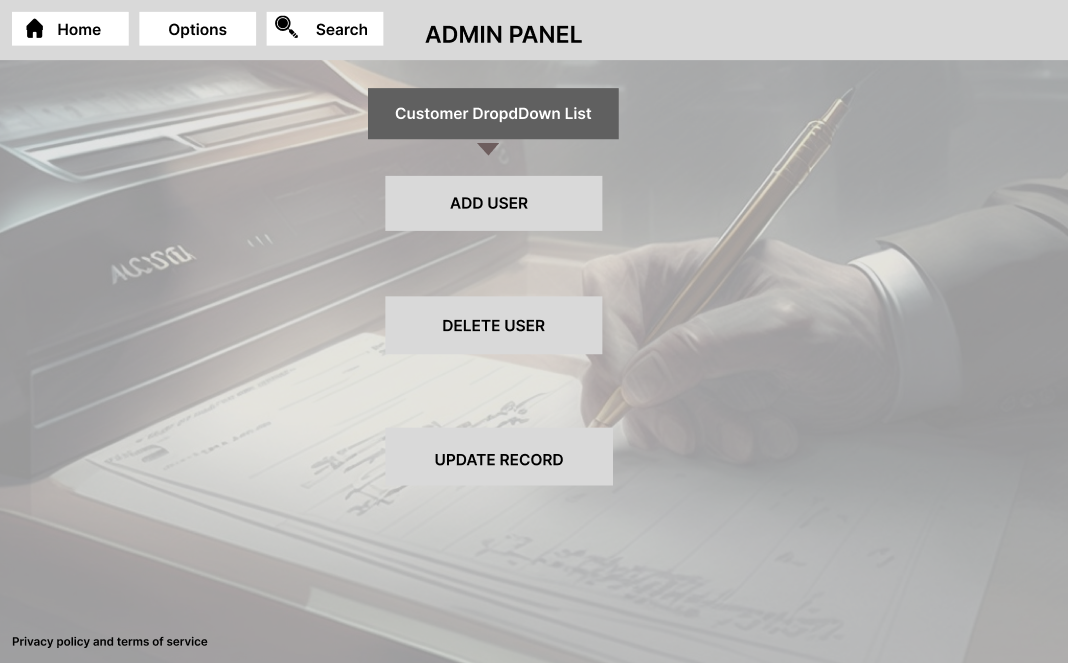
## Workflow

First Screen will pop up when person opens the Desktop App. The person will then have to choose whether he/she is a user of the Desktop App or an Admin. Respectively, he/she will choose the desired mode for login. Now, If the person is Admin, He will Sign In as Admin. Enter his credentials, i.e. Username and Password. Next the Admin panel will open once credentials are verified from the database. This panel will allow the Administrator to Update record of a certain customer, Delete his/her record and also allow adding of new customers. When the Admin will add a user, he will be asked to provide details of the customer. \the details will include information such as Name, CNIC, Contact No. , CNIC along with uploading his 4 Sample Signatures. Now, If the person using the Desktop app is customer and wants to validate his signatures. He will Sign in as User and Enter his credentials, i.e. Username & password. After user logs in, His unique Id will be displayed in front of him/her and will be asked to upload his test signature in the Space provided. The user will then test the Signature by tapping the Test Button. Next, the customer will be shown his 4 Sample Signatures along with his uploaded signature. And will be shown a message of whether His Signature is Matched or Not.

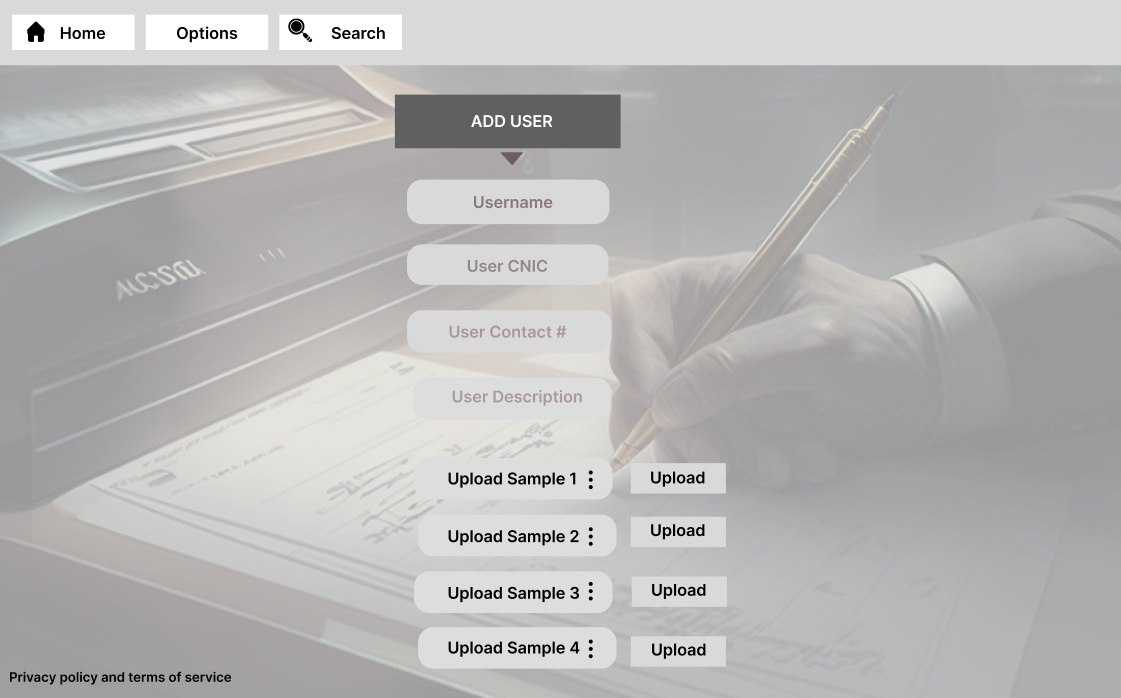
## Screens

**First Screen:**

**2nd Screen if a person is admin:**

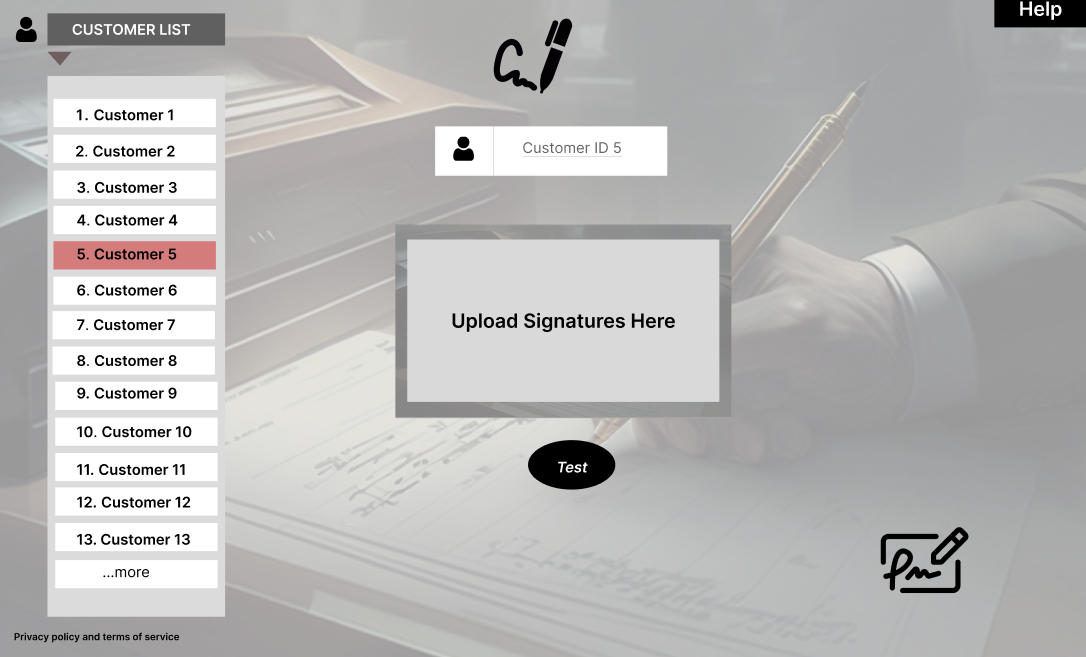
**SCREEN 3:**

**SCREEN FOR REGISTRATION OF USER BY ADMIN:**

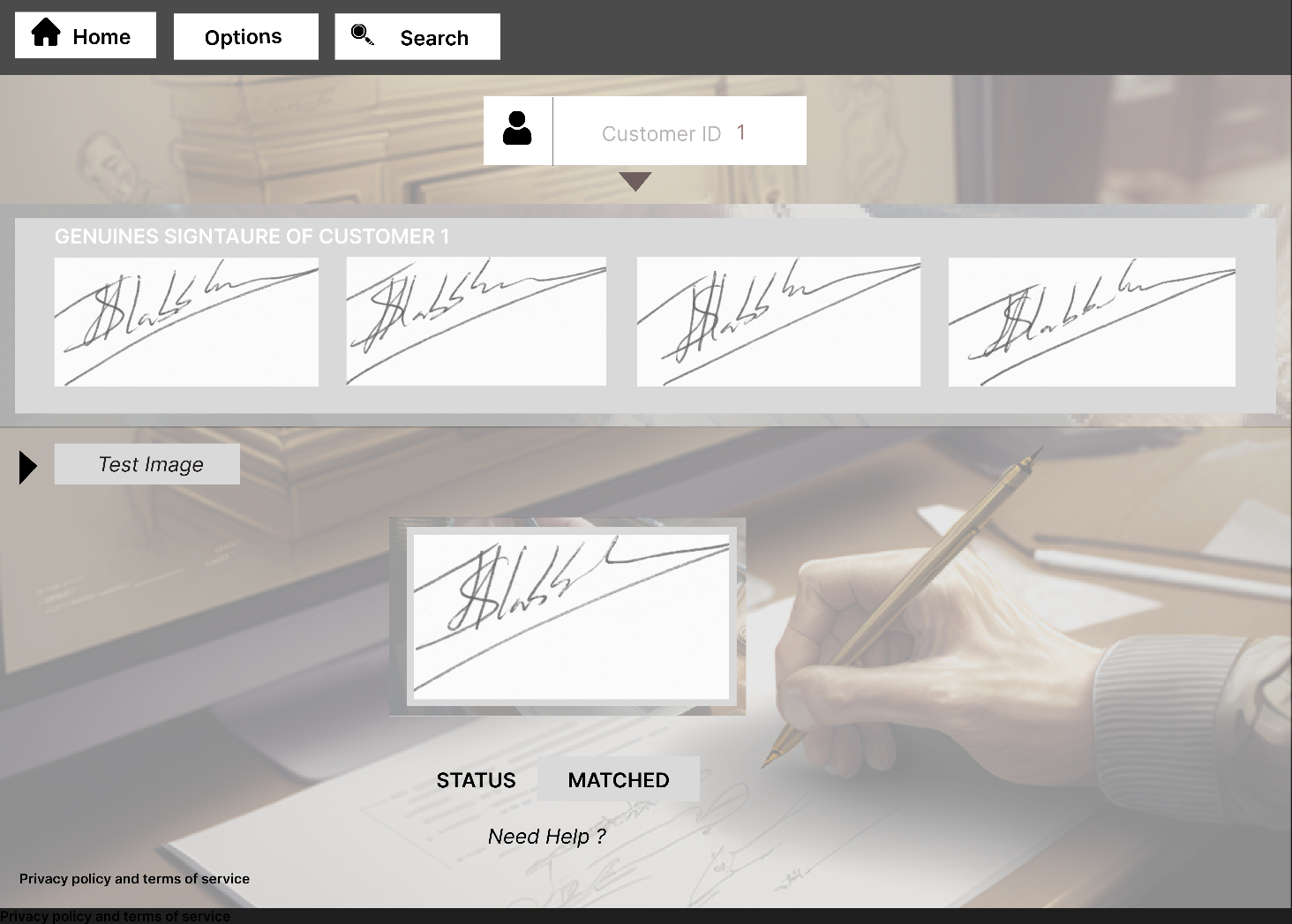


**USER LOGIN SCREEN:**

**AFTER LOGIN MAIN DISPLAY**

****

**SCREEN SHOWING RESULTS:**



## Additional Information

No Additional information

# Other Design Details

Total six research papers were reviewed 2 papers by each group member which include

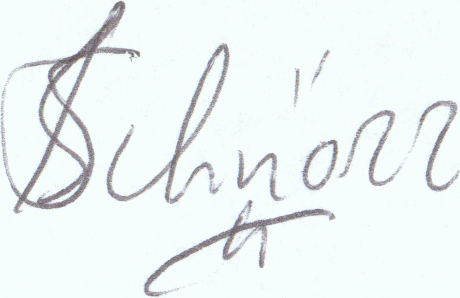
* Signet convolutional Siamese network for writer independent offline signature verification **[7]**
* Proceeding of SPIE offline SVS using Convolutional Siamese network
* Signature verification using Siamese network based on one shot learning **[6]**
* signature verification using a convolutional neural network **[4]**
* Offline signature verification with CNN [**5]**
* Machine learning based offline signature verification system A systematic review **[8]**

4 papers Focus on **CEDAR [2]** dataset and two papers focus on **SIGCOMP 2011 [1]** dataset by studying these papers we got knowledge about the architecture, the model and algorithms we are going to use in this project 3 papers focus on convolutional Siamese network 1 paper talked about machine learning techniques for signature verification and 2 papers were on signature verification using CNN, we understood algorithm and architecture of the model we are going to implement to a great extent by studying these papers, we got knowledge about Siamese networks the layers in Siamese network like the layer which calculate distance or also there can be averaging layer, we got knowledge about the CNN architecture and how it extracts features, Loss functions how to calculate loss how to calculate accuracy of a system how to verify error rate true positives true negatives and overall accuracy of the system along with specificity and sensitivity, papers also talked about the results and performance of algorithm on different datasets. Our primary focus was on CEDAR and SIGCOMP datasets, by reading papers we came to know that algorithm achieve maximum efficiency 100% in some cases where CEDAR dataset is used and a less accuracy when SIGCOMP dataset is used, we came to know that images in CEDAR are available in grayscale mode so preprocessing will be easy in cedar dataset on the contrary data in SIGCOMP is not well structured and images are also not available in grayscale mode so preprocessing will be difficult but SIGCOMP being a large dataset is good for training the algorithm as compared to CEDAR as cedar has less images available than SIGCOMP, we came to know that CEDAR give accurate results when the algorithm is being tested whereas SIGCOMP provide better results when better training of the model is required, we also got knowledge about preprocessing techniques especially image preprocessing techniques superiority of Siamese network was proved over simple CNN architecture by reading these research papers, we got information about neural networks and there layers also we came to know about the effective data split ratio into training and testing parts, we gained knowledge about the implementation of algorithm, we also came to know about APIS and backend like KERAS and TENSORFLOW for model implementation and efficient pairing of data for model training. Multiple image processing techniques were written I paper like Hough transformation and Otsu’s method also different distance calculating algorithms were understood by us like Euclidian distance cosine distance or any combination of two distance calculating algorithms i.e., hybrid distance also there comparison was made in paper, also the reviewed papers talked about what kind of databases, preprocessing techniques and feature extraction methods are required, comparisons between preprocessing techniques and datasets were made some research questions were raised and answered to give a clear understanding about the concepts, paper also talked about deep neural network and shallow neural network, usage of performance evaluation metrics etc., after studying these research papers we have clear understanding of the problem and model we are going to implement.

**Preprocessing done on dataset and images:**

Also we have worked on dataset preprocessing and image preprocessing we arranged data in the format where there are 4 genuine signature images which are matched by test signature image, the test signature image can either be fake or real so for every test signature there will be 4 genuine images to compare and obtain results complete details about this are given in phase 1 document also we have done image preprocessing first we targeted ICDAR SIGCOMP 2011 **[1]** dataset in this dataset there are black signatures on blue background, also in real world the signatures can be of any color or the background on which signatures are done can be of multiple colors which will make it very tough for model to get trained on this type of data so in this case we will do gray scalingof the images to eliminate this problem some of the images of signatures from dataset are below:

**ORIGINAL IMAGES IN DATASET:**

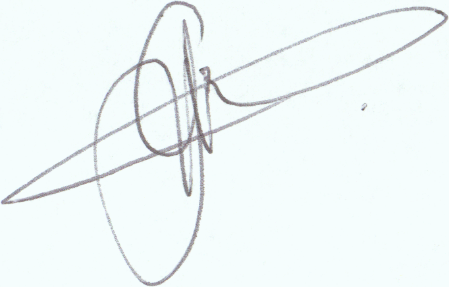
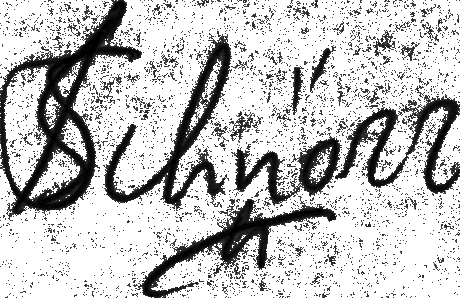
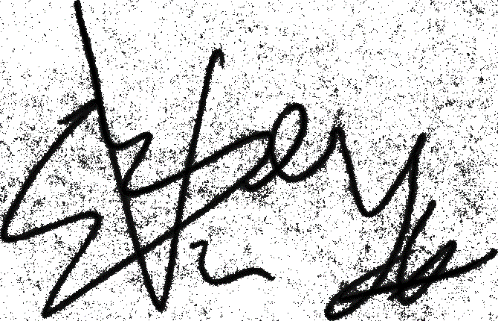
 

Fig 11: Sample signature images from dataset

**GRAY SCALED IMAGES USING OPENCV:**

Firstly, we tried to convert these images to grayscale mode using OPENCV library images were converted to grayscale but the problem with opencv was that it reads image in BGR mode which can sometime cause problem in image color or processing. The images after gray scaling using opencv are:

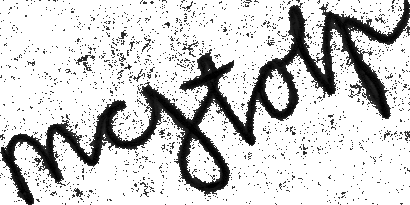
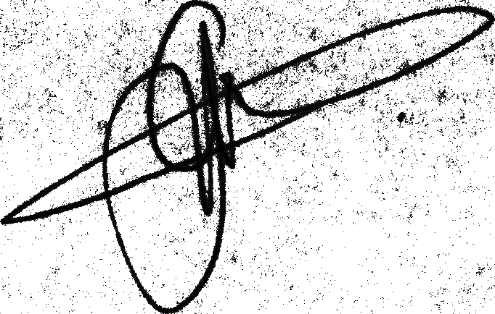
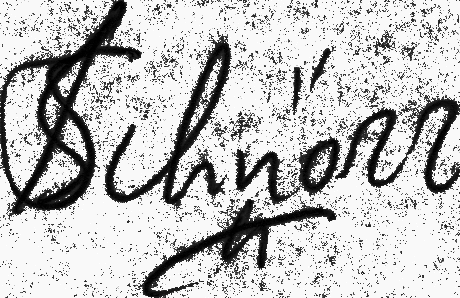
 

Fig 12: Gray scaled images using Opencv library

**GRAY SCALED IMAGES USING SKIMAGE LIBRARY:**

We also tried SKIMAGE library to convert colored signature image to grayscale and it gave better results compared to opencv library as it reads image in RGB format. The sample of grayscaled sign images using skimage library are:

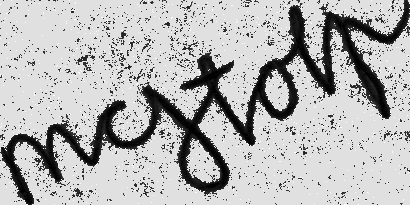
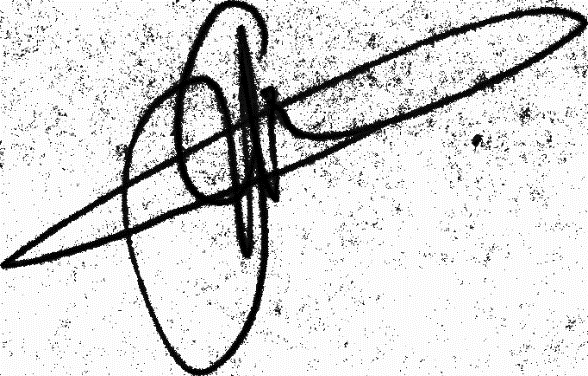
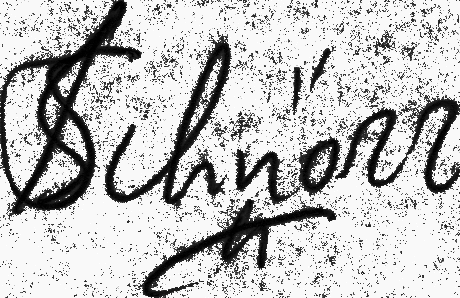
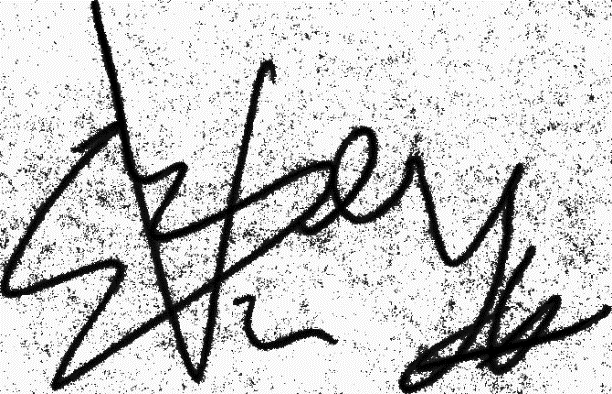
 

Fig 13: Gray scaled signature images using SKIMAGE library

**GRAYSCALE IMAGES WITH GLOBAL HISTOGRAM EQUALIZATION:**

After comparison of these results, we decided to continue with skimage library to further preprocess the images. After converting the images to grayscale mode, we decided to do histogram equalization to enhance the contrast so our model can get better training. We tried two equalization techniques i.e., global histogram equalization and local area histogram equalization the result of global histogram equalization on grayscaled images are:

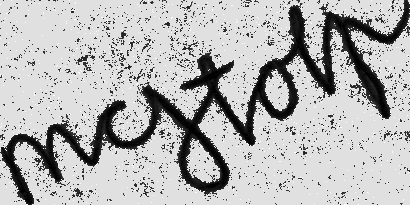
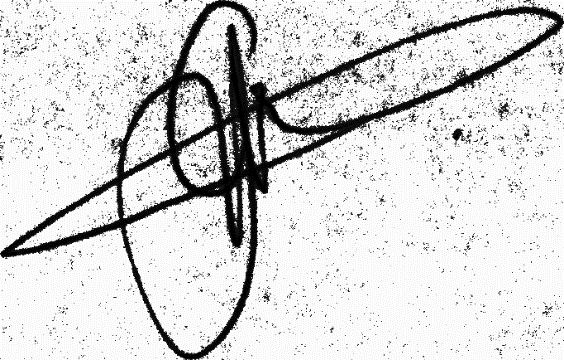
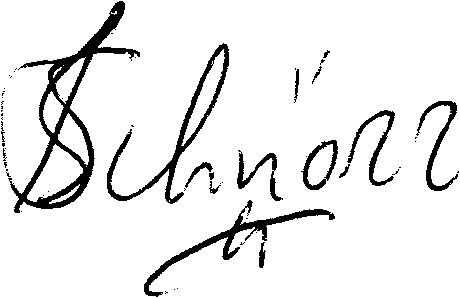
 

Fig 14: gray scaled images after global histogram equalization

**IMAGES AFTER GRAYSCALING AND GLOBAL HIST EQUALIZATION HAVING OTSU METHOD IMPLEMENTED:**

Now after converting images to grayscale mode and doing histogram equalization noise can be seen clearly in the background to eliminate this noise we will use a method called OTSU’s method or OTSU thresholding result of images after applying OTSU thresholding are:

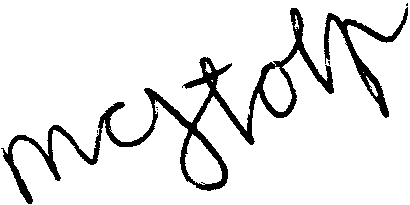
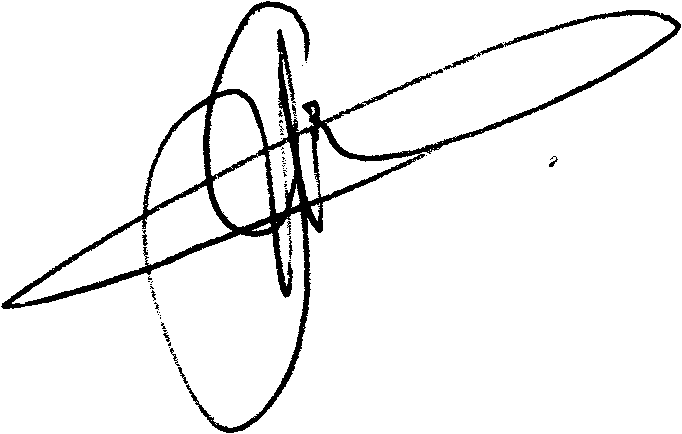
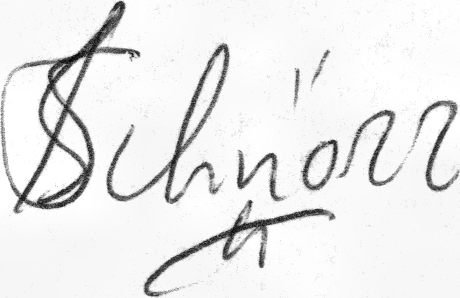
 

Fig 15: Signature images after applying Otsu thresholding, histogram equalization and gray scaling

**GRAY SCALED IMAGES HAVING CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION (CLAHE) IMPLEMENTED INSTEAD OF GLOBAL HIST EQ:**

There is a type of histogram equalization known as local area histogram equalization (CLAHE) which enhance contrast of the image locally unlike global histogram equalization the result of CLAHE on grayscale images are as under:

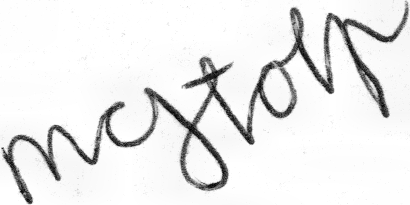
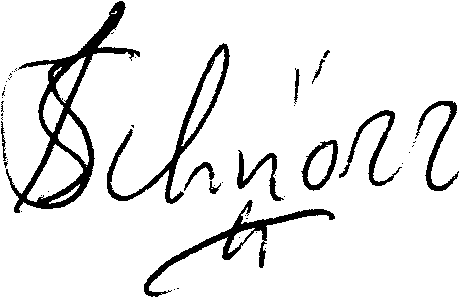
 

Fig 16: Images after applying CLAHE on grayscale images

**OTSU AND CLAHE IMPLEMENTED ON GRAYSCALED IMAGES**

Finally, we applied Otsus thresholding after local area histogram equalization and the results are:

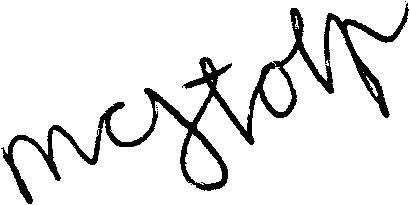
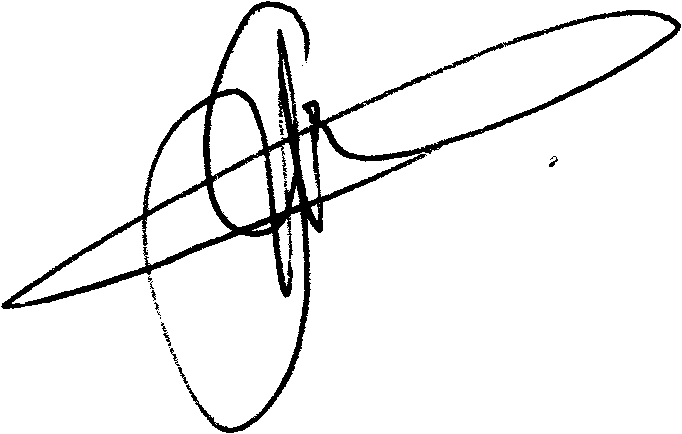
 

Fig 17: Final results obtained after applying Otsu on CLAHE

# Revised Project Plan

In the first phase of project we reviewed, studied and understood related papers, we also analyzed dataset the structure/arrangement of data in dataset and it’s splitting into training and testing parts, we understood Siamese architecture and CNN algorithm with the help of literature review after literature review, we worked on dataset and performed dataset preprocessing by arranging it into desired form i.e.

4 original signature images compared with test image that can either be forged signatures or real signatures, after dataset preprocessing, we worked on image preprocessing and understood that what we can do and what we can’t do for example we can convert image to grayscale mode we can augment images i.e., we can rotate images but to a certain extent. Now in the second phase of our project we are implementing existing Algorithms that we understood from literature review and will train and test model for our understanding also we have done image preprocessing and converted images into grayscale mode then removed noise from image background using OTSU thresholding. We also implemented different image enhancement techniques like CLAHE and global histogram equalization then compare them and find out that which technique will be best for image preprocessing. We also came up with the idea of using KERAS Deep Learning API after comparison with PYTORCH. We considered KERAS because of better community support and ease of use, for implementation of existing algorithm we consulted KERAS’s own site keras.io **[10]** for understanding the implementation and code of Siamese network using KERAS and now we are trying to modify the model according to the approach implemented in research paper **[7]**. In the third phase of project our Algorithm i.e. modified version of Siamese network (N-Way Siamese network) **[9]** will be implemented and we will train the algorithm based on preprocessed dataset i.e., 4 original images compared with 1 test image and result of the algorithms will be obtained finally in 4th phase of our project we will make a user friendly desktop app which will provide interface to our model and will publish a research paper in which we will compare the result of our algorithm with traditional model which uses simple CNN and superiority or inferiority of our system will be proved and comparison will be made

# References

**[1]** M. Liwicki et al., "Signature Verification Competition for Online and Offline Skilled Forgeries (SigComp2011)," 2011 International Conference on Document Analysis and Recognition, 2011, pp. 1480-1484, doi: 10.1109/ICDAR.2011.294.

**[2]**Kaggle [cedar\_dataset\_university\_of\_buffalo](https://www.kaggle.com/datasets/shreelakshmigp/cedardataset)

**[3]** Wikipedia. <https://en.wikipedia.org/wiki/Siamese_neural_network>

**[4]** Navid, S. M. A., Priya, S. H., Khandakar, N. H., Ferdous, Z., & Haque, A. B. (2019). Signature verification using convolutional neural network. In *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)* (pp. 35-39). IEEE.

**[5]** Alvarez, G., Sheffer, B., & Bryant, M. (2016). Offline signature verification with convolutional neural networks. *Tech. Report*.

**[6]** ARISOY, M. V. SIGNATURE VERIFICATION USING SIAMESE NEURAL NETWORK ONE-SHOT LEARNING. *International Journal of Engineering and Innovative Research*, *3*(3), 248-260.

**[7]** Dey, S., Dutta, A., Toledo, J. I., Ghosh, S. K., Lladós, J., & Pal, U. (2017). Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*.

**[8]** Hameed, M. M., Ahmad, R., Kiah, M. L. M., & Murtaza, G. (2021). Machine learning-based offline signature verification systems: a systematic review. *Signal Processing: Image Communication*, *93*, 116139.

Appendix A: Glossary

**SVS:** SVS means Signature Verification System it’s a process in which signatures are verified using computer aided technology instead of human

**CNN:** convolutional Neural network, a type of artificial neural network, which is widely used for image/object recognition and classification.

**Siamese network:** A neural network architecture containing two or more identical subnetworks

**Dataset:** A structured collection of data

**Preprocessing:** preliminary processing of data to prepare it for the primary processing or for further analysis

**Grayscale:** in digital images, grayscale means that the value of each pixel represents only the intensity information of the light

**Histogram Equalization:** A technique for adjusting image intensities to enhance contrast

**CLAHE:** CLAHE means Contrast Limited Adaptive Histogram Equalization. It is a technique used to improve the visibility level of foggy images.

**N-WAY Siamese network:** A modified version of conventional Siamese architecture where there can be N number of subnetworks.

Appendix B: IV & V Report

**(Independent verification & validation)**

**IV & V Resource**

Name Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S#** | **Defect Description** | **Origin Stage** | **Status** | **Fix Time** | |
| **Hours** | **Minutes** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| … |  |  |  |  |  |

**Table 1: List of non-trivial defects**

This document has been adapted from the following:

1. Previous project templates at UCP
2. High-level Technical Design, Centers for Medicare & Medicaid Services. (www.cms.gov)