**FACE RECOGNITION BASED ATM OR BANKING SYSTEM**

**ABSTRACT:**

Now a days, Face recognition has a wide range of possible applications from person identification and surveillance to electronics marketing and advertising for selected customers . Haar Cascade based algorithm has been applied for fast and simple face detection from the input image. The face image is then being converted into grayscale image. After that, the iris candidates are extracted from the intensity valleys from the detected face. Costs of each iris candidates are calculated. Finally the iris candidates are paired up and the cost of each possible pairing is computed by a combination of mathematical models .Face recognition is considered as one of the most reliable solution .The result show that the proposed algorithm has able to train more amount of data and high accuracy..

.

**INTRODUCTION:**

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task. we can use machine learning technology ,open CV , sqllite database and Haar Cascade algorithm

With the technological advances in financial infrastructure, most bank customers prefer to use Automatic Teller Machines (ATMs) and Internet websites for carrying out their banking transactions. Financial users especially utilize ATMs for physical transactions like cash withdrawal or cash deposit. However, just like any other system, ATMs are also suffering from numerous issues caused by users. Among these problems, card and/or cash forgetting (CCF) is a common issue.The main goal of our work is to propose a computer vision framework which uses the embedded ATM camera to perform face detection and recognition in order to prevent such unnecessary losses generated by CCF. In the studied scenario, we consider the case where a customer withdraws money from an ATM in a conventional setting. After the customer inserts the card into the ATM, the proposed system starts to perform face detection and builds a temporary face database for the customer using the camera located inside the ATM. If the customer leaves the ATM without taking his/her card or cash, the ATM waits for the customer to be back instead of retracting the forgotten item. If the system finds out there is a different customer approaching the ATM before the card/cash holder, the card/cash will be retracted at that moment. This scenario is fundamentally different than biometric authentication scenarios, in which a person’s image is matched to a gallery image acquired, possibly, a long time before the matching, under different conditions. In this scenario, the matching image and the gallery image are separated by mere minutes at most.

**Purpose:**

Facial recognition software has become increasingly popular in the past several years .It is used everywhere airports, venues ,shopping centres and even by the law enforcement .While there are a few potential benefits to using the technology to prevent and solve crimes ,there are many concerns about the privacy ,safety and legislation regarding the use of the technology..

**Aim:**

This paper presents a proposed design that implements face authentication based

on opencv to reduce the risks of fraud in mobile banking. Thus, when a customer initiates a mobile banking transaction, the facial recognition system would request that the user takes a picture form the front facing camera of the mobile device, and the system would compare the captured picture to the one already stored on the banking transaction server.

.

**Objectives:**

Facial recognition software has a liveliness detection which prevents hackers from using a picture of the customer for impersonation purposes. It also applies to other biometric modalities such as fingerprints where the liveliness detection does exactly that it accesses the ‘liveliness’ of the facial image as it is known.

**LITERATURE SUVEY:**

**Existing system**

* Due to speedy development in science and technology, forthcoming innovations are being settled with sturdy security. However, on the opposite hand, threats are being expose to destroy this security level. Although sweetening in automation has created a positive impact overall, however numerous money establishments like banks and applications like AT M are still subjected to thefts and frauds. The present ATM model uses a card and a PIN which supplies rise to extend in attacks within the kind of taken cards, or because of statically allotted PINs, duplicity of cards and numerous different threats.

**Disadvantages:**

* The ATM card or PIN of a user are often spied upon and might be accessed simply by getting the cardboard by faulty suggests that. This may cause some serious consequence.
* Victimization the brute force, if we tend to attempt to crack the present static four-digit PIN it are often tired 9999 tries, therefore weakening the safety.

**Proposed System:**

* To beat, hybrid model that consists of standard options beside further options like face recognition and one-time countersign (OTP)is employed. Information holds data a couple of user’s account details, pictures of his/her face and a mobile variety which can improve security to an outsized extent.
* The system is enforced on the master card size with extended capability of open supply pc vision(OpenCV) software package that is employed for image process operation. High level security mechanism is provided by the consecutive actions like at first system captures the external body part(face) and check whether or not the external body part is detected properly or not. If the face is detected properly, ATM transaction willbe processed.

**Advantages:**

1. ATM security that is more reliable by using open CV Face Recognition and Deduction. It reduces the fraud actions in ATM.
2. security will improve the rate of transaction and also the bank can profit through this system. The experiment results in card loosing and PIN forgetting issues

**DESIGN AND ANALYSIS**

**SYSTEM ANALYSIS:**

#### SOFTWARE REQUIREMENT SPECIFICATION

#### Requirements Specification:

#### Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

#### HARDWARE REQUIREMENTS:

#### The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

#### System : intel core i3.

#### Hard Disk : 500 GB.

#### Monitor : 15 VGA Color.

#### Mouse : Logitech.

#### RAM : 2 GB.

#### SOFTWARE REQUIREMENTS:

#### The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

#### Operating system : Windows XP/7/10

* Coding Language : python
* Development environment : anaconda , Jupiter
* Dataset : cricket dataset
* IDE : anaconda prompt

**TECHNOLOGIES USED:**

**Python:**

**IMPLEMENTATION ON (PYTHON):**

**What Is A Script?**

Up to this point, I have concentrated on the interactive programming capability of Python.  This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode.

**Scripts are reusable:**

Basically, a script is a text file containing the statements that comprise a Python program.  Once you have created the script, you can execute it over and over without having to retype it each time.

**Scripts are editable:**

Perhaps, more importantly, you can make  different versions of the script by modifying the statements from one file to the next using a text editor.  Then you can execute each of the individual versions.  In this way, it is easy to create different programs with a minimum amount of typing.

**You will need a text editor:**

Just about any text editor will suffice for creating Python script files.

You can use Microsoft Notepad, Microsoft WordPad, Microsoft Word, or just about any word processor if you want to.

**Difference between a script and a program**

**Script:** Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, where as the applications they control are traditionally compiled to native machine code.

**Program:**

The program has an executable form that the computer can use directly to execute the instructions.

The same program in its human-readable source code form, from which executable programs are derived(e.g., compiled)

**Python**

what is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don’t know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to “big name” languages. Hopefully I can explain it for you.

**Python concepts**

If you not interested in the how and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it’s a great one to start programming with.

• Open source general-purpose language.

• Object Oriented, Procedural, Functional

• Easy to interface with C/ObjC/Java/Fortran

• Easy-ish to interface with C++ (via SWIG)

• Great interactive environment

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## **MACHINE LEARNING**

One of the finest word heard in today time is Machine Learning. Either it be at work or different places the machine learning has been an integral part of today’s technology. Though its evolutional zing and developing in a rapid rate and development and deployment of the same is still in progress. The machine learning itself had brought a random change in today worlds because of which automation is in frame which was a mere existence in the past.

It's an aspiring term in today’s time. One of the moves that all the firm are interested into. It’s a leading pillar for tomorrow leading the world to a better future of evolution where the customization and labor work can be reduce to half and the safety of the survival can be withheld to stand tall for the better utilization of human mind. Keeping that in picture it’s been a hazard to many more in terms of irrespective field of interest. Since Machine is considered most efficient and the level of mistakes are kept at the minimum the level of work flow can be a work of hazard and further improvement on the same may create a thousand sitting idle in home creating a larger impact on unemployment and livelihood. Which in other way is a threat to the society too.

ML is the abbreviation for Machine Learning. In other word it is making a human mind fitting inside a machine which uses the same to perform the task of thousands. Machine Learning deals with the higher aspects of learning techniques and algorithm which are highly aligned to make the work flow seamlessly effortless with the human tendency of doing work.

Algorithm of such are improvising in nature which learns by themselves and fit themselves in the world of impairment by getting the required data and adjusting with the same giving the effective results out of the same. ML is a subsidiary or the subset of an AI(Artificial Intelligence). It is a mathematical model where computation of the testcases plays the major role in driving of the results.

A wide level of machine learning architecture are implemented today to turn on the yield factor and make people life more efficient in terms of livelihood. Various use of such in Message Filtering like spams, Trash automation are automated and carried out by the same. Since the efficiency is way more than a human tendency. Multi-tasking and processing is also initiated by the same giving a dual output which a human can never ever possibly be able to.

Statistics is the major key role in driving the machine learning in figure. It deals with computation of statistics in a wide range view and processing the same to give an data driven output causing it more sensible and resourcesable. Not only to the same it optimizes the resources and the efficiency is unbitable and reliable in terms of any means.

Though its being evolutional in nature but it has integrated itself well with the terms of computational and digitalization. Various computational fields like Data Mining, Statistical Analysis, Optimization of resources, Automation are a major part of it. Here the machine has the capacity to process the result on its own as same as the human bring. This process can be initiator as well as the derivable. The statistical flow is mainly reasonable with data driven pattern even the unstructured or the semi-structured data can be processed and approximate answer to the same can be derived. All the equations are derived and the closest value to it’s aligned field is found and the proximity is determined.

The classification of the same can be listed as follows:

### **ANACONDA:**

A conda environment is a directory that contains a specific collection of conda packages that you have installed. For example, you may have one environment with NumPy 1.7 and its dependencies, and another environment with NumPy 1.6 for legacy testing. If you change one environment, your other environments are not affected. You can easily activate or deactivate environments, which is how you switch between them.

#### Creating a virtual environment-

A virtual environment is a tool that helps to keep dependencies required by different projects separate by creating isolated spaces for them that contain per-project dependencies for them.

**Python Libraries:**

**Numpy:**

* NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more.
* At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:
  + NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.
  + The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
  + NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python’s built-in sequences.
  + A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today’s scientific/mathematical Python-based software, just knowing how to use Python’s built-in sequence types is insufficient - one also needs to know how to use NumPy arrays.

**import pandas as pd**

* Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.
* Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

**import sklearn**

* In general, a learning problem considers a set of n samples of data and then tries to predict properties of unknown data. If each sample is more than a single number and, for instance, a multi-dimensional entry (aka multivariate data), it is said to have several attributes or features.
* Learning problems fall into a few categories:
  + supervised learning, in which the data comes with additional attributes that we want to predict (Click here to go to the scikit-learn supervised learning page).This problem can be either:
* classification: samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class.
* regression: if the desired output consists of one or more continuous variables, then the task is called regression. An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight.
  + unsupervised learning, in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization (Click here to go to the Scikit-Learn unsupervised learning page).

**import sklearn.metrics as m**

There are 3 different APIs for evaluating the quality of a model’s predictions:

• Estimator score method: Estimators have a score method providing a default evaluation criterion for the problem they are designed to solve. This is not discussed on this page, but in each estimator’s documentation.

• Scoring parameter: Model-evaluation tools using cross-validation (such as model\_selection.cross\_val\_score and model\_selection.GridSearchCV) rely on an internal scoring strategy. This is discussed in the section The scoring parameter: defining model evaluation rules.

• Metric functions: The metrics module implements functions assessing prediction error for specific purposes. These metrics are detailed in sections on Classification metrics, Multilabel ranking metrics, Regression metrics and Clustering metrics.

Finally, Dummy estimators are useful to get a baseline value of those metrics for random predictions.

**from sklearn import metrics**

Accuracy classification score.

In multilabel classification, this function computes subset accuracy: the set of labels predicted for a sample must exactly match the corresponding set of labels in y\_true.

**SYSTEM DESIGN AND UML DIAGRAMS:**

## **SYSTEM ARCHITECTURE**

## 

## The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It’s a serious impact on the later part, notably testing and maintenance.

## The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

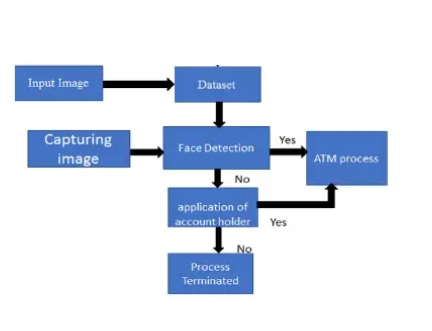
## System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

## At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

## Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

## In system design the main target is on distinguishing the modules, whereas throughout careful style the main target is on planning the logic for every of the modules.

## 

****Figure 7.1: Architecture diagram

## **DATA FLOW DIAGRAMS**

## Data Flow Diagram can also be termed as bubble chart. It is a pictorial or graphical form, which can be applied to represent the input data to a system and multiple functions carried out on the data and the generated output by the system.

## A graphical tool accustomed describe and analyze the instant of knowledge through a system manual or automatic together with the method, stores of knowledge, and delays within the system. The transformation of knowledge from input to output, through processes, is also delineate logically and severally of the physical elements related to the system. The DFD is also known as a data flow graph or a bubble chart.The BasicNotation used to create a DFD’s are as follows:

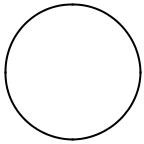
* **Dataflow:**





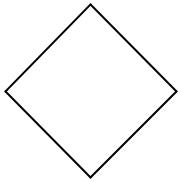
* **Process:**

.



* **Source:**
* **Data Store:**



* **Rhombus**: decision

**7.3 UML DIAGRAMS**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

**User Model View**

This view represents the system from the users perspective. The analysis representation describes a usage scenario from the end-users perspective.

**Structural Model view**

In this model the data and functionality are arrived from inside the system. This model view models the static structures.

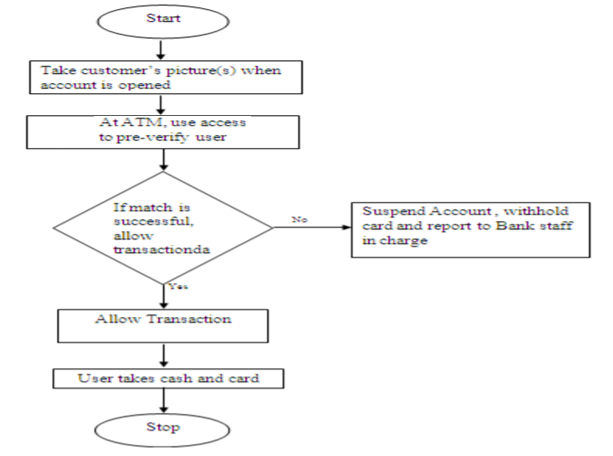
**Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**Implementation Model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

**DATA FLOW DIAGRAM:**



**USE CASE DIAGRAM**

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

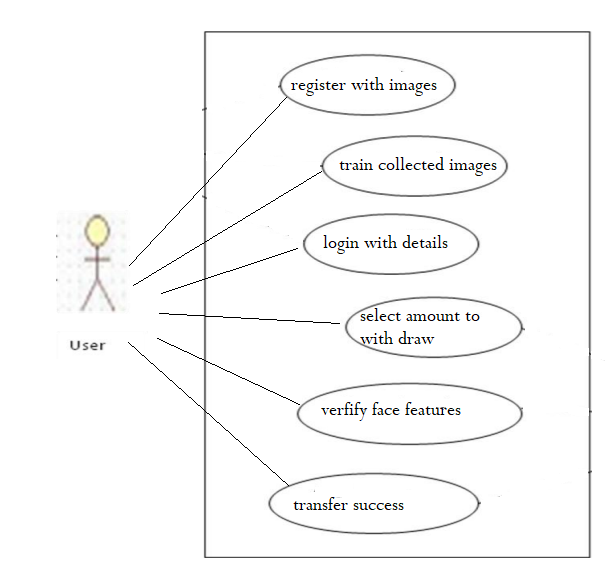


Figure 7.3.1 Use Case Diagram

**CLASS DIAGRAM**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. A class with three sections, in the diagram, classes is represented with boxes which contain three parts:

The upper part holds the name of the class

The middle part contains the attributes of the class

The bottom part gives the methods or operations the class can take or undertake.

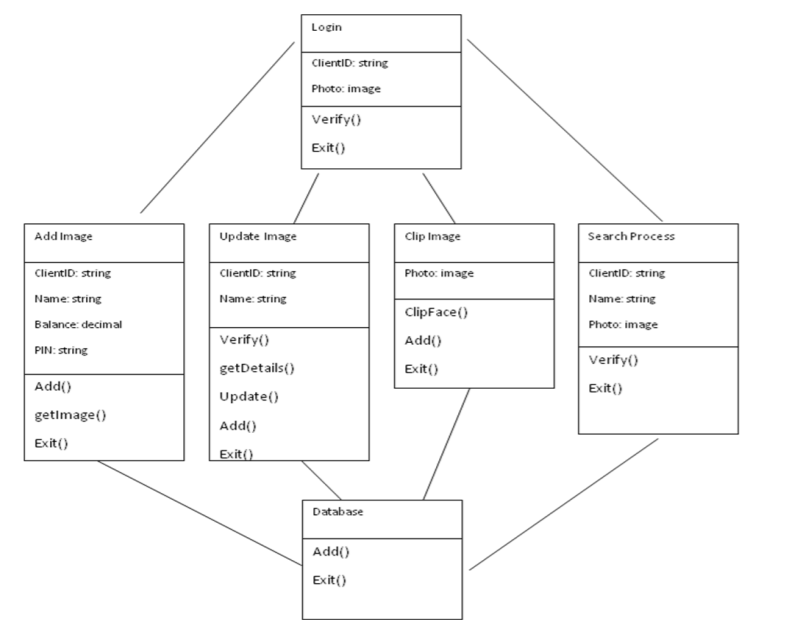


Figure 7.3.2: Class Diagram.

## **SEQUENCEDIAGRAM**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

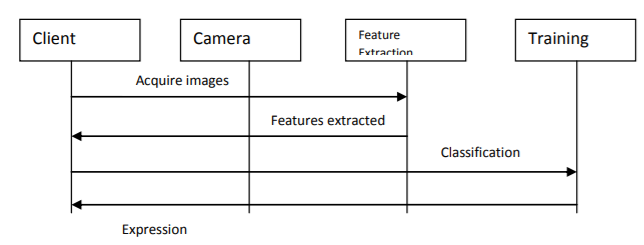


Figure 7.3.3: Sequence diagram

## **ACTIVITY DIAGRAM**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

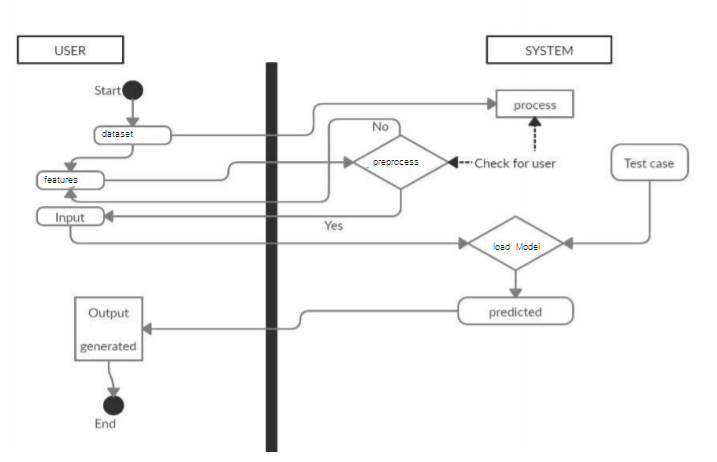
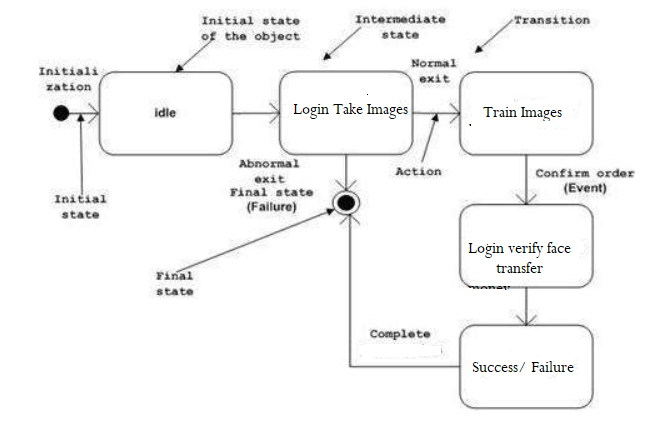
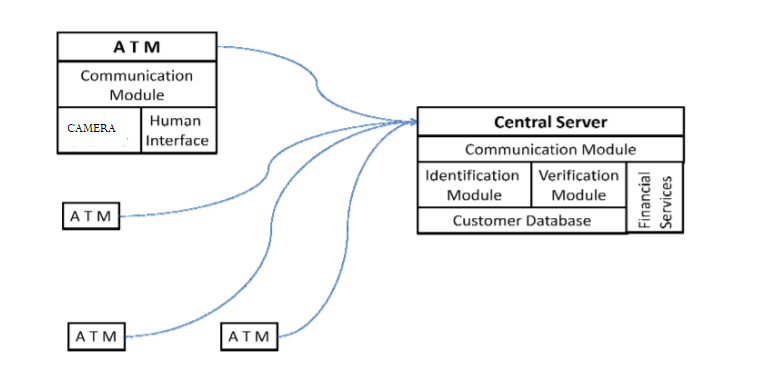


Figure 7.3.4: Activity Diagram

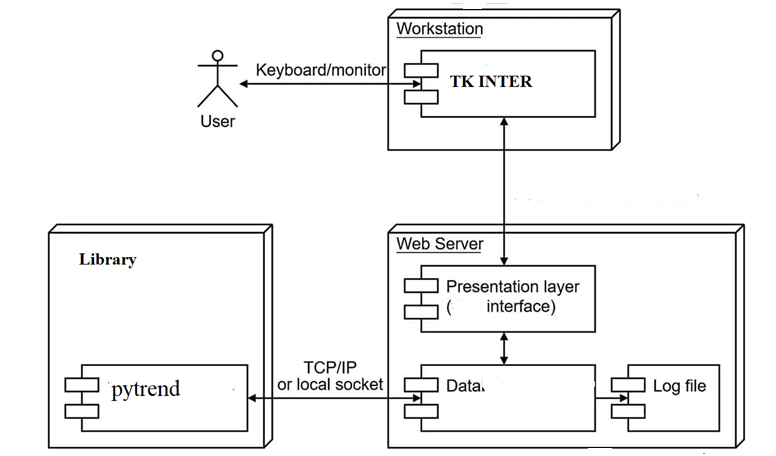
**State Chart Diagram**



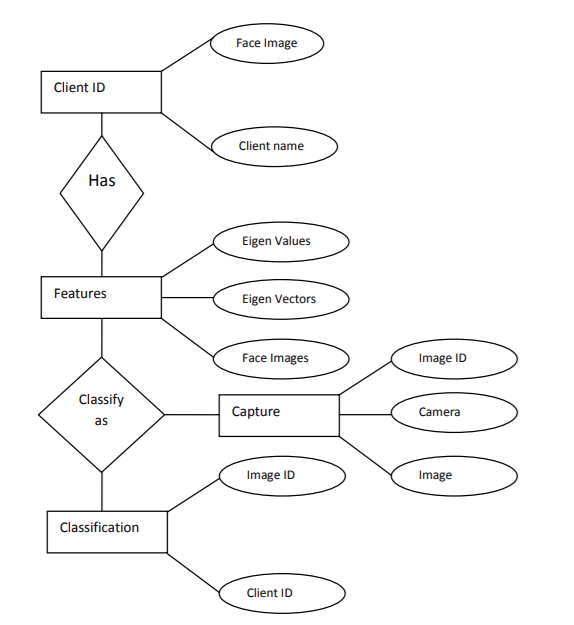
**COMPONENT DIAGRAM**



**DEPLOYMENT DIAGRAM**



**ER DIAGRAM**



**Module Design and organisation**

#### 1. Preprocessing: After cropping out the face region, we performed histogram equalization so as to avoid illumination change. Then an ellipseshaped mask is used to remove some of the corner regions such as the collar, for reducing the unnecessary effects of these regions. The masking step, however, is not employed for the detected out-of-session face images since the images are of low quality and have small size.

#### Feature extraction:

#### After the preprocessing step, the Local Binary Pattern (LBP) operator is used for our face description. LBP is one of the best performing texture descriptors and it has been widely used in various applications. It is proven to be highly discriminative and its key advantages, namely, its invariance to monotonic gray-level changes and computational efficiency, make it suitable for demanding image analysis tasks.

#### Verification System:

#### Its used to verify the user face and body recognition and compare sample gallery then its give valid results to make correct decision its verifies the valid user or invalid user using the frontal face images captured by an embedded ATM camera.

#### Object Detection: This module is used to detect the user may be wear mask and have any object (strike , gun , knife) using OpenCV’s built-in Haar cascade profile face detector. User wearing big mask incase using best rotation angle for maximizing the face detection.

## OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. Let’s start the chapter by defining the term "Computer Vision".:

## Computer Vision overlaps significantly with the following fields:

## • Image Processing: It focuses on image manipulation.

## • Pattern Recognition: It explains various techniques to classify patterns. • Photogrammetry: It is concerned with obtaining accurate measurements from images. Computer Vision Vs Image Processing Image processing deals with image-to-image transformation. The input and output of image processing are both images. Computer vision is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene

**INPUT OUTPUT DESIGN**

**.INPUT DESIGN**

Input design is a part of overall system design. The main objective during the input design is as given below:

* To produce a cost-effective method of input.
* To achieve the highest possible level of accuracy.
* To ensure that the input is acceptable and understood by the user.

**INPUT STAGES:**

The main input stages can be listed as below:

* Data recording
* Data transcription
* Data conversion
* Data verification
* Data control
* Data transmission
* Data validation
* Data correction

**INPUT TYPES:**

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

* External inputs, which are prime inputs for the system.
* Internal inputs, which are user communications with the system.
* Operational, which are computer department’s communications to the system?
* Interactive, which are inputs entered during a dialogue.

**INPUTMEDIA:**

At this stage choice has to be made about the input media. To conclude about the input media consideration has to be given to;

* Type of input
* Flexibility of format
* Speed
* Accuracy
* Verification methods
* Rejection rates
* Ease of correction
* Storage and handling requirements
* Security
* Easy to use
* Portability

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

**OUTPUT DESIGN**

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

* External Outputs, whose destination is outside the organization
* Internal Outputs whose destination is within organization and they are the
* User’s main interface with the computer.
* Operational outputs whose use is purely within the computer department.
* Interface outputs, which involve the user in communicating directly.

**OUTPUT DEFINITION:**

# The outputs should be defined in terms of the following points:

* + - Type of the output
    - Content of the output
    - Format of the output
    - Location of the output
    - Frequency of the output
    - Volume of the output
    - Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It should be decided as which form of the output is the most suitable.

**IMPLEMENTATION AND RESULTS**

**Introduction:**

The primary contributor to this module was Aleksandr Rybnikov, and Rybnikov included accurate, deep learning face detector. Caffe-based face detector can be found in the face\_detector

To use OpenCV Deep Neural Network module with Caffe models you will need two files and both files

.prototxt file which defines model architecture

.caffemodel file which contains the weights for the actual layers

OpenCV’s face detector is based on the Single Shot Detector framework with a ResNet base network. We won’t go in any details inside architecture model since this is a beginner’s tutorial and we are using pre-trained model.

**METHOD OF IMPLEMENTAION:**

**Face detection in images**

First we are importing necessary packages so that our model could run.

After that, we are parsing command line arguments which we will use at the end to run our model.

A short explanation of parsing code:

— image path for the input image

— prototxt path to prototxt file

— model path to the pre-trained model

— confidence with this we can overwrite the default threshold of 0.5

Now we load our model and create a blob from our image:

Here we are load our model in “net”variable using cv2.dnn.readNetFromCaffe() function for reading a network model stored in Caffe framework with args for “prototxt ”and “model” file paths.

# load the input image and construct an input blob for the image

fter that we load image with cv2.imread() function and args “image” and then extracting height and weight dimensions for image with image.shape[:2] where we extract first and second dimension.

With cv2.dnn.blobFromImage() function we resize image to 300x300 dimension, 1.0 is scalefactor and here we use default value so there is no scaling, after that is spatial size that Convolutional Neural Network expects, last values are mean subtraction values in tuple and they are RGB means, and at the end function returns a “blob” which is our input image after resizing, mean subtraction and normalizing.

In short, mean subtraction is technique used to help our Convolutional Neural Networks since with mean subtraction we are helping to fight with illumination changes in our images. We are using mean values from ImageNet training set, and values are in order for RED, GREEN and BLUE channels.

Now we are applying face detection.

With setInput() we are setting new input value for the network which is our “blob” and with forward() we are running forward pass to compute output of layer.

Now we are looping through detections and drawing boxes around detected faces.

We are extracting confidence and compare it to the confidence threshold so we can filter out detections that are weak.

If confidence is a minimum threshold we proceed to draw a rectangle with the probability of the detection.

First, we calculate x,y coordinates of the bounding box and build confidence with “text ”string which is holding probability of the detection.

With cv2.rectangle() and cv2.putText() we draw rectangle and text on our image. And we are repeating loop process to check if there is any other detections on the image and if there isn’t we are output image on the screen.

With python we are calling face\_detector\_image.py file and adding arguments — image and name of the file, — prototxt for architecture model file and — model for pre-trained model.

Exit from the output is with “q” on your keyboard.

**Data Pre-processing**: It is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis .Data goes through a series of steps during

**pre-processing**: Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data. Data Integration: Data with different representations are put together and conflicts within the data are resolved. Data Transformation: Data is normalized, aggregated and generalized. Data Reduction: This step aims to present a reduced representation of the data in a data warehouse

**FACE DETECTION**: Haar Cascade based algorithm has been applied for fast and simple face detection from the input images. The face images is then being converted into grayscale image .After that , the iris candidates are extracted from the intensity valleys from the detected face. Face detection modules analyses each captured frame and extracts valid faces from each frame. This is very easy humans , but computers need precise instruction. It is distinct from other computer vision technologies that involve human faces , like facial recognition ,analysis ,and tracking

**HAAR FEATURE SELECTION**: Haar -Like is a rectangular simple feature that is used as an input feature for cascaded classifier In there are some filters based on Haar -Like feature. By applying every one of these filters into one special area of the image, the pixel sums under white areas are subtracted from the pixel sums under the black areas. That is the weight of white and black area can be considered as "1" and "-1", respectively.

**FACE RECOGNITION:** Face recognition technology: Ideal for access control, financial transactions and ATM machines. The face key recognition technology performs the following tasks: a. Locates a moving object within the camera view b. Determines if the moving object is face c. Compares live faces with samples from database

**FACE AUTHENTICATION AND TRANSACTION:** This phase is a biometric software application capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person's facial contours. Facial recognition is mostly used for security purposes In this phase the transaction is proceed when the face is matched with the registered user otherwise the transaction is terminated .

**FEATURE EXTRACTION:**

It is the process of transforming the raw pixel values from an image, to a more meaningful and useful information that can be used in other techniques, such as point matching or machine learning. Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval.

**SVM Algorithm**

SVM is a sparse technique. Like nonparametric methods, SVM requires that all the training data be available, that is, stored in memory during the training phase, when the parameters of the SVM model are learned. However, once the model parameters are identified, SVM depends only on a subset of these training instances, called support vectors, for future prediction. Support vectors define the margins of the hyperplanes. Support vectors are found after an optimization step involving an objective function regularized by an error term and a constraint, using Lagrangian relaxation.1 The complexity of the classification task with SVM depends on the number of support vectors rather than the dimensionality of the input space. The number of support vectors that are ultimately retained from the original dataset is data dependent and varies, based on the data complexity, which is captured by the data dimensionality and class separability. The upper bound for the number of support vectors is half the size of the training dataset, but in practice this is rarely the case. The SVM model described mathematically in this chapter is written as a weighted sum of the support vectors, which gives the SVM framework the same advantages as parametric techniques in terms of reduced computational time for testing and storage requirements

##### **TESTING**

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

**Testing Principle**

Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

Testing Methods

There are different methods that can be used for software testing. They are,

**Black­Box Testing**

The technique of testing without having any knowledge of the interior workings of the application is called black­box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black­box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

**White­Box Testing**

White­box testing is the detailed investigation of internal logic and structure of the code. White­box testing is also called glass testing or open­box testing. In order to perform white­ box testing on an application, a tester needs to know the internal workings of the code. The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately.

**Levels of Testing**

There are different levels during the process of testing. Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

**Functional Testing:**

This is a type of black­box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. There are five steps that are involved while testing an application for functionality.

* The determination of the functionality that the intended application is meant to perform.
* The creation of test data based on the specifications of the application.
* The output based on the test data and the specifications of the application.
* The writing of test scenarios and the execution of test cases.
* The comparison of actual and expected results based on the executed test cases.
* **Non­functional Testing**

This section is based upon testing an application from its non-functional attributes. Non-functional testing involves testing software from the requirements which are non-functional in nature but important such as performance, security, user interface, etc. Testing can be done in different levels of SDLC. Few of them are

**Unit Testing:**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. Unit testing is often automated but it can also be done manually. The goal of unit testing is to isolate each part of the

program and show that individual parts are correct in terms of requirements and functionality. Test cases and results are shown in the Tables.

Unit Testing Benefits

* + - Unit testing increases confidence in changing/ maintaining code.
    - Codes are more reusable.
    - Development is faster.
    - The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.
    - Debugging is easy.
    - Codes are more reliable.

Unit testing:

|  |  |
| --- | --- |
| Sl # Test Case : ­ | UTC­1 |
| Name of Test: ­ | Register with applciation |
| Items being tested: ­ | Username and password input |
| Sample Input: ­ | User name and password |
| Expected output: ­ | Process to next page for image capturing |
| Actual output: ­ | Take valid details |
| **Remarks: ­** | **Pass.** |

|  |  |
| --- | --- |
| Sl # Test Case : ­ | UTC­2 |
| Name of Test: ­ | Collect user images |
| Items being tested: ­ | Click space bar to collect images |
| Sample Input: ­ | Each click with collect one image |
| Expected output: ­ | Images are stored in system folder |
| Actual output: ­ | User images are stored in system and excel sheet |
| Remarks: ­ | success |

**Integration Testing:**

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Integration testing is defined as the testing of combined parts of an application to determine if they function correctly. It occurs after unit testing and before validation testing. Integration testing can be done in two ways: Bottom­up integration testing and Top­down integration testing.

* + 1. **Bottom­up Integration**

This testing begins with unit testing, followed by tests of progressively higher­level combinations of units called modules or builds.

* + 1. **Top­down Integration**

In this testing, the highest­level modules are tested first and progressively, lower­level modules are tested thereafter.

In a comprehensive software development environment, bottom­up testing is usually done first, followed by top­down testing. The process concludes with multiple tests of the complete application, preferably in scenarios designed to mimic actual situations. Table 6.5 shows the test cases for integration testing and their results

|  |  |
| --- | --- |
| Sl # Test Case : ­ | ITC­1 |
| Name of Test: ­ | Train collected images |
| Item being tested: ­ | Training process is success or not |
| Sample Input: ­ | User images |
| Expected output: ­ | trained model is updated |
| Actual output: ­ | Features and labels must be loaded |
| Remarks: ­ | Pass. |

|  |  |
| --- | --- |
| Sl # Test Case : ­ | ITC­2 |
| Name of Test: ­ | Initialize Model |
| Item being tested: ­ | Model summary |
| Sample Input: ­ | Fit data with features and labels |
| Expected output: ­ | Model should be trained |
| Actual output: ­ | Training process is performed |
| Remarks: ­ | Pass. |

**System testing**:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black­box testing, and as such, should require no knowledge of the inner design of the code or logic. System testing is important because of the following reasons:

System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.

The application is tested thoroughly to verify that it meets the functional and technical specifications.

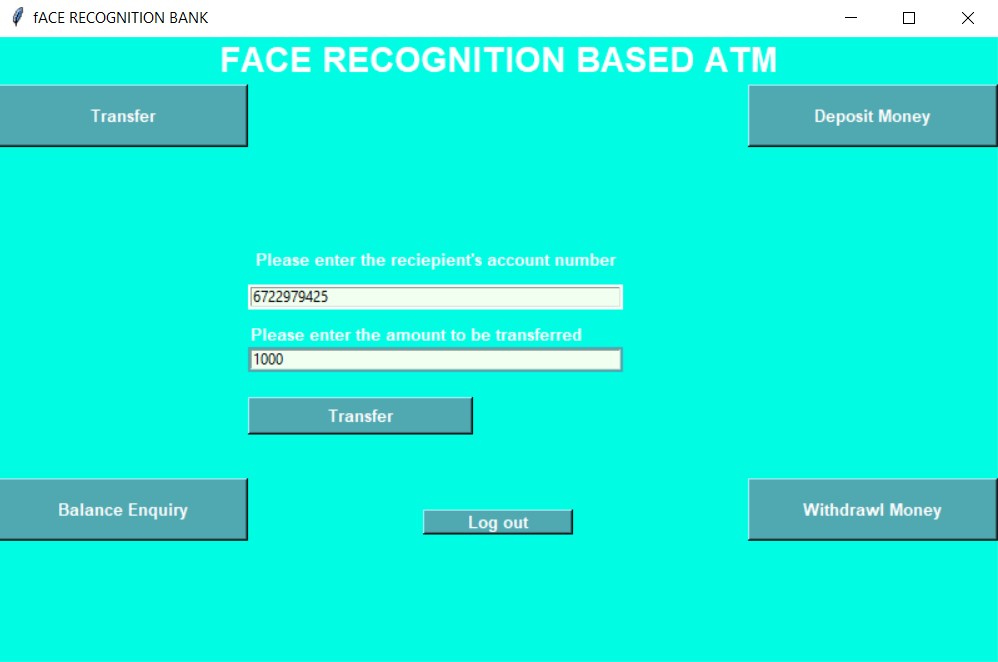
The application is tested in an environment that is very close to the production environment where the application will be deployed.

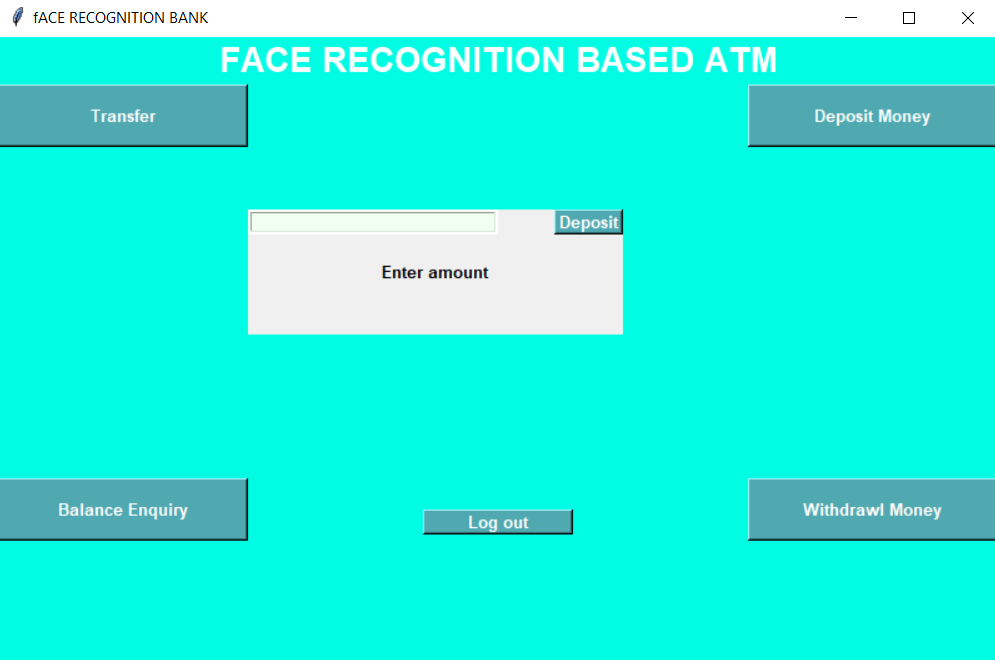
System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

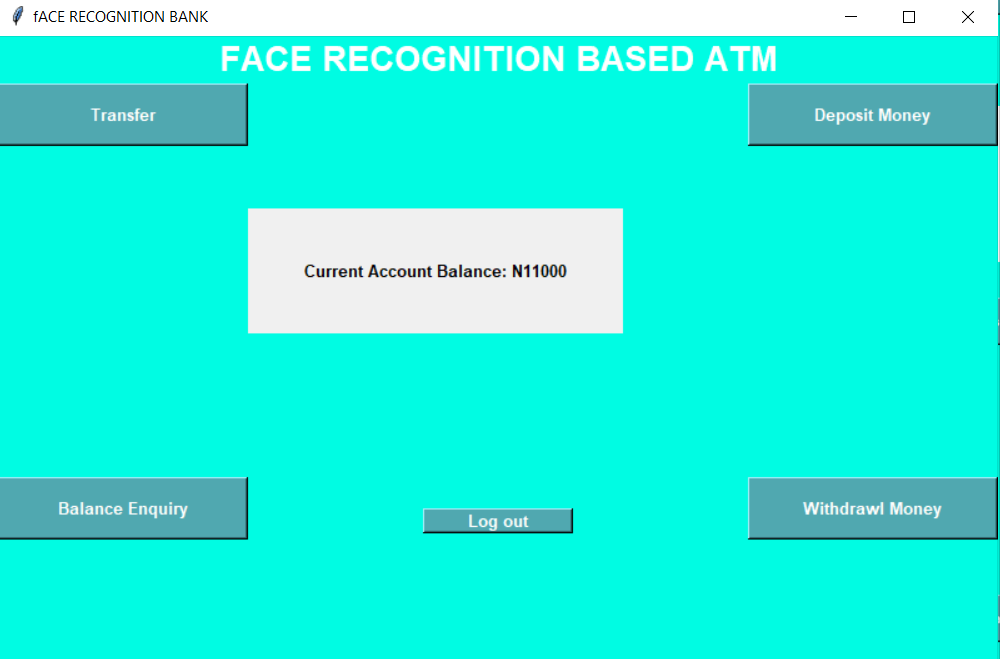
System Testing is shown in below tables

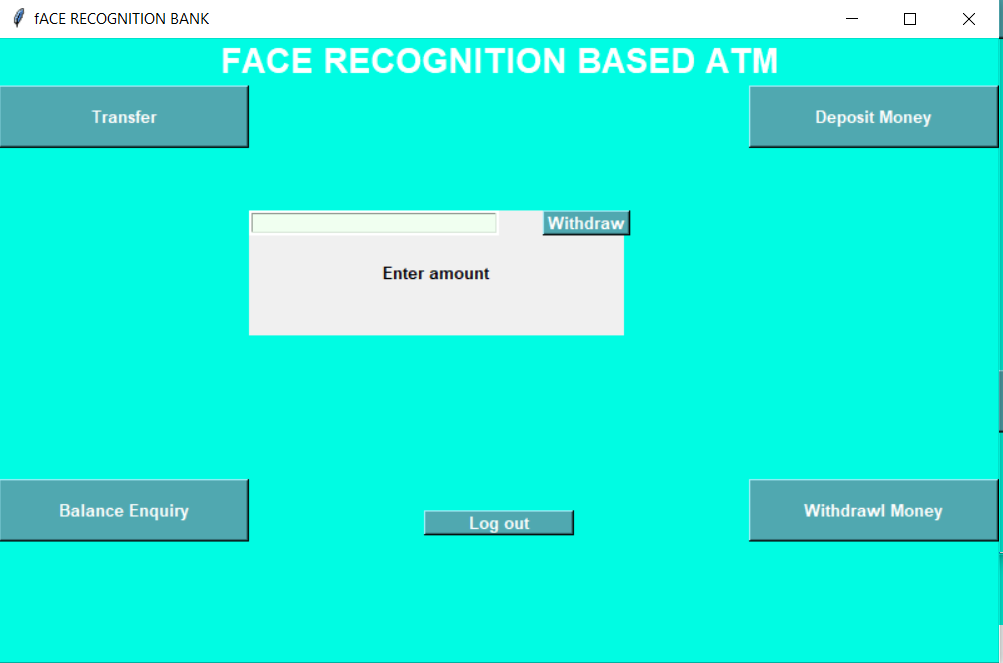
|  |  |
| --- | --- |
| Sl # Test Case : ­ | STC­1 |
| Name of Test: ­ | System testing in various versions of OS |
| Item being tested: ­ | OS compatibility. |
| Sample Input: ­ | Execute the program in windows XP/ Windows­7/8 |
| Expected output: ­ | Performance is better in windows­7 |
| Actual output: ­ | Same as expected output, performance is better in windows­7 |
| Remarks: ­ | Pass |

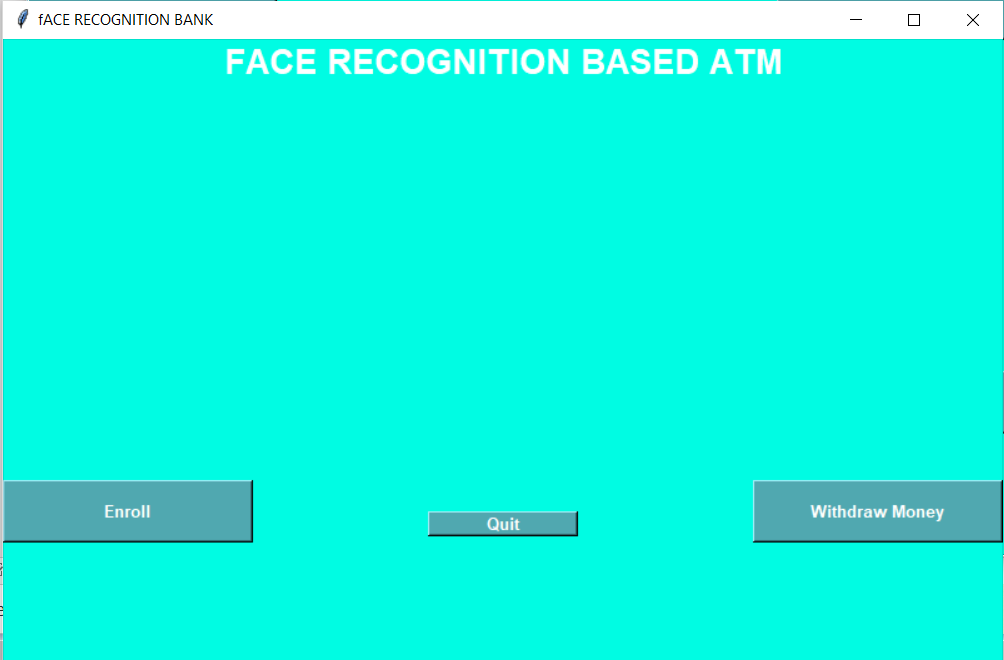
**OUTPUT SCREENS:**

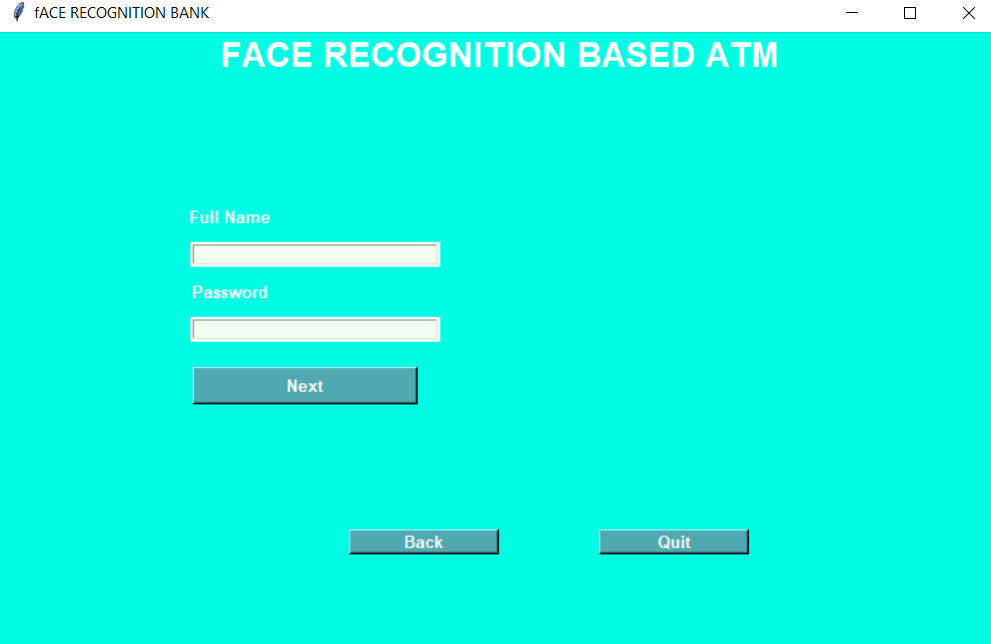












**Conclusion**

This project developed banking transactions using facial identification namely, transfer amounts, card details, names of participants, etc. once facial identity is matched then transaction will finish otherwise it will display “person not recognized”. existing system was over comed . If previous no one can implement this system in Indian bank sector .in this project main focus is safety our money and our transaction. We are using Haar Cascade algorithm for face recognition. Capture module deals with the configuration of video interface and performs the real-time video capture. Face Detection module analyses each captured frame and extracts valid faces from each frame. Face Identification deals with face recognition and verification of the detected face.

**Future Scope:**

* To avoid atm robberies and provide security for atm, To secure such a complex system will be even more difficult than design it. And now people just begin to discuss some issues of ATM security. It will provide some experience for us to implement security services in ATM network.

.

**References:**

Payment: Facial-Recognition An Example of Chinese Consumers Authors: Wen Kun ; Zhang Min Jung Kang, IEEE Access, Year: 2019

[2].Secure multifactor using NFC system payment authentication Authors: Anirudhan Adukkathayar; Gokul S ; Krishnan Rajashree Chinchole, 2015 10th International Conference on Computer Science & Education (ICCSE)\

[3].Biometric Face Recognition Payment System Authors: Surekha. R. Gondkar Saurab. Dr. C. S. Mala International Journal of Engineering Research & Technology NCESC - 2018 Conference Proceedings

[4].Facial Recognition in Banking – Current Applications Author:Niccolo Mejia,2019 Conference Proceedings

[5]."Face Detection and Recognition for Bank Transaction ", International Journal of Emerging Technologies and Innovative Research Authors Sudarshan Dumbre ,Shamita Kulkarni ,Devashree Deshpande ,P.V.Mulmule Journal of Emerging Technologies and Innovative Research 2018