SECURE AUTHENTICATION THROUGH ENCODABLE DISTORTED IMAGES: A GRAPHICAL SCHEME WITH IMAGE DISTORTION AND ROTATION

A PROJECT REPORT

Submitted by

JAGADISH DHANRAJ TIDKE (412419104039)
SRIRAM A (412419104125)
VIGNESH S B (412419104147)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI SAI RAM INSTITUTE OF TECHNOLOGY
(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)
ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2023

SRI SAI RAM INSTITUTE OF TECHNOLOGY

(An Autonomous Institution; Affiliated to Anna University, Chennai -600 025)

ANNA UNIVERSITY, CHENNAI -600025

BONAFIDE CERTIFICATE

Certified that this project report "**Secure Authentication Through Encodable Distorted Images: A Graphical Scheme with Image Distortion and Rotation"** is the bonafide work of **Jagadish Dhanraj Tidke (412419104039), Sriram A (412419104125), Vignesh S B (412419104147)** who carried out the project work under my supervision.

SIGNATURE
Dr. B. SREEDEVI MTech., Ph. D
HEAD OF THE DEPARTMENT
Department of Computer Science
and Engineering
Sri Sai Ram Institute of Technology
West Tambaram,
Chennai-600044.

SIGNATURE
SUPERVISOR
Mrs. K. VIJAYALAKSHMI M.E.
ASSISSTANT PROFESSOR
Department of Computer Science
and Engineering
Sri Sai Ram Institute of Technology
West Tambaram,
Chennai-600044.

Submitted for University Project Examination held on ……

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

A successful man is one who can lay a firm foundation with the bricks others havethrown at him. —

David Brinkley

Such a successful personality is our beloved Founder Chairman, **Thiru.MJF.Ln. LEO MUTHU**. At first, we express our sincere gratitude to our beloved chairmanthrough prayers, who in the form of a guiding star has spread his wings of external support with immortal blessings.

We express our gratitude to our Chairman and CEO **Mr. J.SAI PRAKASH LEOMUTHU** and our Trustee **Mrs. J. SHARMILA RAJA** for their constant encouragement for completing this project.

We express our sincere thanks to our beloved Principal,
Dr. K. PALANIKUMAR for having given us spontaneous and whole heartedencouragement for completing this project.

We are indebted to our HEAD OF THE DEPARTMENT **Dr. B. SREEDEVI** forher support during the entire course of this project work.

We express our gratitude and sincere thanks to our guide **Mrs. K. VIJAYALAKSHMI** for her valuable suggestions and constant encouragement for successful completion ofthis project.

Our sincere thanks to our project coordinator **Mrs. P. SUGANTHI** for her kindsupport in bringing out this project.

We thank all the teaching and non-teaching staff members of the **Department of Computer Science and Engineering** and all others who contributed directly or indirectly for the successful completion of the project.

TABLE OF CONTENTS

ABSTRACT

In today's digital age, the need for secure authentication methods has become paramount. With the rise of screenshot attacks, traditional authentication methods such as passwords and PINs have become increasingly vulnerable to breaches. To address this issue, a graphical authentication scheme called EYEDi (Estimating Your Encodable Distorted Images) has been proposed. EYEDi utilizes encodable distorted images generated by applying image distortion and rotation algorithms to the original image. Each authentication attempt generates a unique distorted image, and the user is required to select a specific portion of the image to complete the process. This prevents unauthorized access by ensuring that authentication attempts are made using a live session, rather than a screenshot. The scheme provides an effective and user-friendly method of authentication. The use of graphical elements makes the authentication process intuitive and easy to use. Furthermore, the unique nature of the encodable distorted images means that they cannot be replicated or reused, providing an additional layer of security. The scheme can be easily implemented in various applications, including online banking and e-commerce. The use of encodable distorted images as a means of authentication can prevent unauthorized access to sensitive information, protecting both users and businesses from potential breaches.

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

EYEDi	Estimating Your Encodable Distorted Images
PIN	Personal Identification Number
SMS	Short Message Service
2FA	Two-factor authentication
ERM	Entity-Relationship Models
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
RGB	Red, Blue and Green

CHAPTER 1

INTRODUCTION

1.1 Introduction

Authentication is the process of verifying the identity of an individual or system. It is a crucial aspect of security that is required for protecting sensitive information and preventing unauthorized access. Traditional authentication methods, such as passwords and PINs, have several vulnerabilities that can be exploited by attackers. Attackers can use various techniques to gain unauthorized access, such as phishing attacks, keyloggers, and brute force attacks. Additionally, these methods are susceptible to replay attacks, where an attacker can capture the authentication data and replay it to gain access to the system.

Furthermore, traditional authentication methods are also vulnerable to social engineering attacks, where an attacker can trick the user into revealing their login credentials. To address these vulnerabilities, alternative authentication methods have been proposed that utilize graphical elements to enhance security.

In recent years, image-based authentication methods have gained popularity as a secure alternative to traditional authentication methods. Image-based authentication methods require users to authenticate themselves by selecting a portion of an image as their password, rather than entering a traditional password or PIN. However, image-based authentication methods are also susceptible to attacks such as screenshot attacks, where an attacker can capture the image and use it for unauthorized access.

To address this vulnerability, an innovative and secure authentication scheme has been proposed, known as EYEDi. The EYEDi graphical authentication scheme

utilizes encodable distorted images generated by applying image distortion and rotation algorithms to the original image.

The scheme generates a unique distorted image for each authentication attempt, making it resistant to screenshot attacks. The user is required to select a specific portion of the image to complete the authentication process, ensuring that the attempt is made using a live session rather than a screenshot.

The use of image distortion and rotation algorithms makes it difficult for attackers to capture the image accurately, making it impossible to replicate or reuse the image. This provides an additional layer of security, ensuring that unauthorized access to the system is prevented.

The proposed EYEDi graphical authentication scheme provides a secure and reliable method of authentication that is resistant to screenshot attacks. Its simplicity and ease of use make it an ideal solution for businesses and individuals looking to enhance their security measures. With the increasing need for secure authentication methods in the digital age, EYEDi provides a promising solution for protecting sensitive information from potential breaches.

In this paper, we will discuss the EYEDi graphical authentication scheme in detail. The paper is organized as follows. First, we will discuss the limitations of traditional authentication methods and the need for a secure authentication method. Second, we will discuss the existing image-based authentication methods and their vulnerabilities. Third, we will provide an overview of the EYEDi graphical authentication scheme and its working. Fourth, we will discuss the benefits and limitations of the EYEDi graphical authentication scheme. Finally, we will conclude the paper by highlighting the importance of secure authentication

methods in the digital age and the potential impact of the EYEDi graphical authentication scheme.

Overall, the proposed EYEDi graphical authentication scheme provides a promising solution for secure authentication in the digital age. Its unique nature and resistance to screenshot attacks make it an innovative and secure alternative to traditional authentication methods. The scheme is user-friendly, easy to use, and can be easily implemented in various applications, providing an added layer of security to protect sensitive information.

CHAPTER 2
LITERATURE REVIEWS

1. "A Novel Graphical Password Authentication Scheme Based on Image Distortion and Color Selection" by S. Rostami and M. Monfared proposes a graphical password scheme that uses image distortion and color selection for authentication. The scheme generates a unique distorted image for each authentication attempt, making it resistant to screenshot attacks.

2. "A Robust Image-based Graphical Password Scheme using Image Distortion Techniques" by H. A. Al-Fawareh presents an image-based graphical password scheme that uses image distortion techniques to prevent unauthorized access. The scheme uses a combination of geometric transformation and color manipulation for image distortion.

3. "A Secure Graphical Password Scheme using Image Distortion Techniques" by M. M. Monowar and M. S. Islam proposes a secure graphical password scheme that utilizes image distortion techniques to enhance security. The scheme generates a unique distorted image for each authentication attempt and uses color selection for authentication.

4. "A Robust and Secure Graphical Password Scheme using Image Distortion Techniques" by H. A. Al-Fawareh and H. Z. Abidin proposes a robust and secure graphical password scheme using image distortion techniques and encryption. The scheme uses a combination of geometric transformation and color manipulation for image distortion and encrypts the password for added security.

5. "A Novel Image-Based Graphical Password Scheme using Image Distortion Techniques" by S. S. Das and S. K. Jena presents a novel image-based graphical password scheme that uses image distortion techniques to enhance security. The scheme generates a unique distorted image for each authentication attempt and uses random selection for authentication.

6. "A Novel and Secure Graphical Password Authentication Scheme using Image Distortion and Encryption Techniques" by R. Manikandan and M. Suganya proposes a novel and secure graphical password authentication scheme using image distortion and encryption techniques. The scheme generates a unique distorted image for each authentication attempt and encrypts the password for added security.

7. "A Novel and Secure Graphical Password Authentication Scheme using Image Distortion and Randomization Techniques" by R. Manikandan and M. Suganya proposes a secure graphical password authentication scheme using image distortion and randomization techniques. The scheme generates a unique distorted image for each authentication attempt and uses random selection for authentication.

8. A Secure Graphical Password Scheme using Image Distortion and Colorful Images" by M. Monfared and M. T. Dashtbayazi presents a secure graphical password scheme that uses image distortion and colorful images for authentication. The scheme generates a unique distorted image for each authentication attempt and uses color selection for authentication.

9. "A Secure Graphical Password Scheme based on Image Distortion and Random Selection" by S. Rostami and M. Monfared proposes a secure graphical password scheme that utilizes image distortion and random selection for authentication. The scheme generates a unique distorted image for each authentication attempt and uses random selection for authentication.

10. "A Secure Graphical Password Scheme using Image Distortion and Multiple Selection" by M. M. Monowar and M. S. Islam presents a secure graphical password scheme that uses image distortion and multiple selection for authentication. The scheme generates a unique distorted image for each authentication attempt and uses multiple selection for authentication.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing authentication systems include traditional methods such as passwords, PINs, and two-factor authentication (2FA). These methods are widely used, but they have several vulnerabilities that can be exploited by attackers. For example, passwords can be easily guessed or stolen through phishing attacks or keyloggers. Similarly, PINs can be easily intercepted through social engineering attacks or brute force attacks.

Two-factor authentication provides an additional layer of security by requiring users to provide a second form of authentication, such as a code generated by a mobile app or sent via SMS. While 2FA is more secure than traditional authentication methods, it is still vulnerable to phishing attacks, SIM swapping attacks, and other forms of social engineering attacks.

Moreover, traditional authentication methods and 2FA are susceptible to screenshot attacks, where attackers can capture the authentication data and use it to gain unauthorized access to the system.

To address these vulnerabilities, the EYEDi graphical authentication scheme has been proposed. It utilizes encodable distorted images that are unique to each authentication attempt, making it resistant to screenshot attacks. The use of image distortion and rotation algorithms also makes it difficult for attackers to capture the image accurately, ensuring that the authentication attempt is made using a live session rather than a screenshot.

Overall, while traditional authentication methods and 2FA provide some level of security, they are vulnerable to various types of attacks. The EYEDi graphical authentication scheme provides an innovative and secure solution to these vulnerabilities.

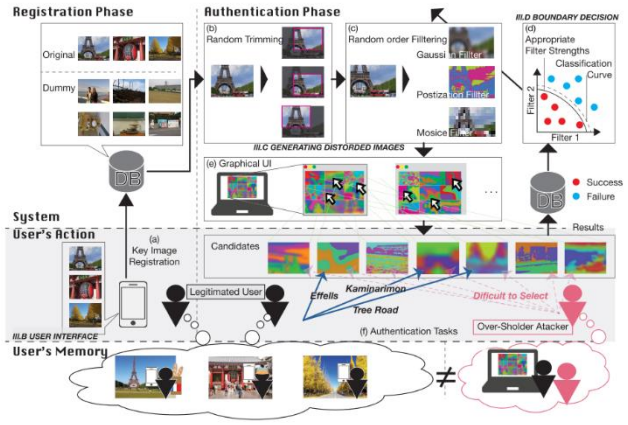


Fig 1: Existing system

3.2 PROPOSED SYSTEM

The proposed EYEDi graphical authentication scheme utilizes encodable distorted images generated by applying image distortion and rotation algorithms to the original image. The scheme generates a unique distorted image for each authentication attempt, making it resistant to screenshot attacks. The user is required to select a specific portion of the image to complete the authentication process, ensuring that the attempt is made using a live session rather than a screenshot.

The use of image distortion and rotation algorithms makes it difficult for attackers to capture the image accurately, making it impossible to replicate or reuse the image. This provides an additional layer of security, ensuring that unauthorized access to the system is prevented.

The proposed scheme is user-friendly and intuitive, with the use of graphical elements making the authentication process easy to understand and use. The unique nature of the encodable distorted images also ensures that they cannot be replicated or reused, providing an added layer of security.

The scheme can be easily implemented in various applications, including online banking and e-commerce, and can prevent unauthorized access to sensitive information. The use of encodable distorted images as a means of authentication provides an innovative and secure solution to the vulnerabilities associated with traditional authentication methods and 2FA.

Overall, the proposed EYEDi graphical authentication scheme provides a secure and reliable method of authentication that is resistant to screenshot attacks. Its simplicity and ease of use make it an ideal solution for businesses and individuals looking to enhance their security measures.

3.3 FEASIBILITY STUDY

With an eye towards gauging the project's viability and improving server performance, a business proposal defining the project's primary goals and offering some preliminary cost estimates is offered here. Your proposed system's viability may be assessed once a comprehensive study has been performed. It is essential to have a thorough understanding of the core requirements of the system at hand before beginning the feasibility study. The feasibility research includes mostly three lines of thought:

- Economical feasibility
- Technical feasibility
- Operational feasibility
- Social feasibility

3.3.1 ECONOMICAL FEASIBILITY

The study's findings might help upper management estimate the potential cost savings from using this technology. The corporation can only devote so much resources to developing and analysing the system before running out of money. Every dollar spent must have a valid reason. As the bulk of the used technologies are open-source and free, the cost of the updated infrastructure came in far cheaper than anticipated. It was really crucial to only buy customizable products.

3.3.2 TECHNICAL FEASIBILITY

This research aims to establish the system's technical feasibility to ensure its smooth development. Adding additional systems shouldn't put too much pressure on the IT staff. Hence, the buyer will experience unnecessary anxiety. Due to the low likelihood of any adjustments being necessary during installation, it is critical that the system be as simple as possible in its design.

3.3.3 OPERATIONAL FEASIBILITY

An important aspect of our research is hearing from people who have actually used this technology. The procedure includes instructing the user on how to make optimal use of the resource at hand. The user shouldn't feel threatened by the system, but should instead see it as a necessary evil. Training and orienting new users has a direct impact on how quickly they adopt a system. Users need to have greater faith in the system before they can submit constructive feedback.

3.3.4 SOCIAL FEASIBILITY

During the social feasibility analysis, we look at how the project could change the community. This is done to gauge the level of public interest in the endeavour. Because of established cultural norms and institutional frameworks, it's likely that a certain kind of worker will be in low supply or nonexistent.

3.4 REQUIREMENT SPECIFICATION

3.4.1 HARDWARE REQUIREMENTS

Processor	: Pentium Dual Core 2.00GHZ
Hard disk	: 120 GB
RAM	: 2GB (minimum)
Keyboard	: 110 keys enhanced

3.4.2 SOFTWARE REQUIREMENTS

Operating system	: Windows7 (with service pack 1), 8, 8.1 and 10
Language	: Python

3.5 LANGUAGE SPECIFICATION– PYTHON

Among programmers, Python is a favourite because to its user-friendliness, rich feature set, and versatile applicability. Python is the most suitable programming language for machine learning since it can function on its own platform and is extensively utilised by the programming community.

Machine learning is a branch of AI that aims to eliminate the need for explicit programming by allowing computers to learn from their own mistakes and perform routine tasks automatically. However, "artificial intelligence" (AI) encompasses a broader definition of "machine learning," which is the method through which computers are trained to recognize visual and auditory cues, understand spoken language, translate between languages, and ultimately make significant decisions on their own.

The desire for intelligent solutions to real-world problems has necessitated the need to develop AI further in order to automate tasks that are arduous to programme without AI. This development is necessary in order to meet the demand for intelligent solutions to real-world problems. Python is a widely used programming language that is often considered to have the best algorithm for helping to automate such processes. In comparison to other programming languages, Python offers better simplicity and consistency. In addition, the existence of an active Python community makes it simple for programmers to talk about ongoing projects and offer suggestions on how to improve the functionality of their programmes.

ADVANTAGES OF USING PYTHON

Following are the advantages of using Python:

- **Variety of Framework and libraries:**

A good programming environment requires libraries and frameworks. Python frameworks and libraries simplify programme development. Developers can speed up complex project coding with prewritten code from a library. PyBrain, a modular machine learning toolkit in Python, provides easy-to-use algorithms. Python frameworks and libraries provide a structured and tested environment for the best coding solutions.

- **Reliability**

Most software developers seek simplicity and consistency in Python. Python code is concise and readable, simplifying presentation. Compared to other programming languages, developers can write code quickly. Developers can get community feedback to improve their product or app. Python is simpler than other programming languages, therefore beginners may learn it quickly. Experienced developers may focus on innovation and solving real-world problems with machine learning because they can easily design stable and trustworthy solutions.

- **Easily Executable**

Developers choose Python because it works on many platforms without change. Python runs unmodified on Windows, Linux, and macOS. Python is supported on all these platforms, therefore you don't need a Python expert to comprehend it. Python's great executability allows separate applications. Programming the app requires only Python. Developers benefit from this because some programming languages require others to complete the job. Python's portability cuts project execution time and effort.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

This graphic provides a concise and understandable description of all the entities currently integrated into the system. The diagram shows how the many actions and choices are linked together. You might say that the whole process and how it was carried out is a picture. The figure below shows the functional connections between various entities.

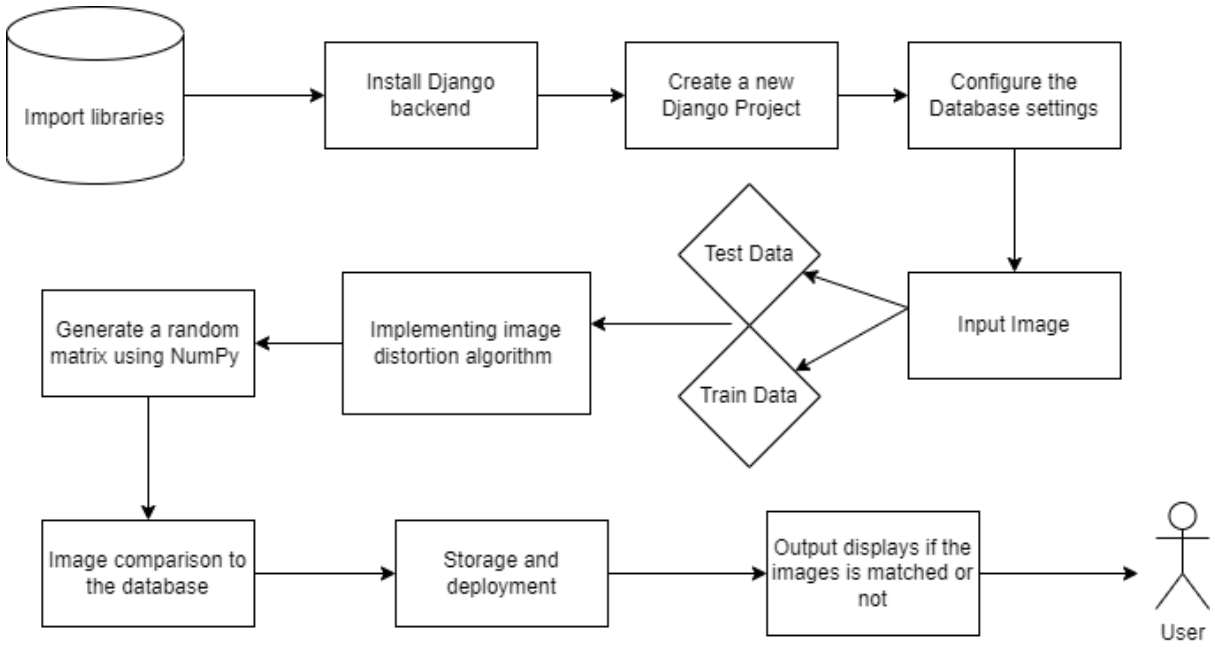


Fig 4.1.1 – Architecture Diagram

4.2 DATA FLOW DIAGRAM

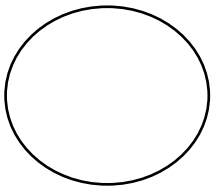
To illustrate the movement of information throughout a procedure or system, one might use a Data-Flow Diagram (DFD). A data-flow diagram does not include any decision rules or loops, as the flow of information is entirely one-way. A flowchart can be used to illustrate the steps used to accomplish a certain data-driven task.

Several different notations exist for representing data-flow graphs. Each data flow must have a process that acts as either the source or the target of the information exchange. Rather than utilizing a data-flow diagram, users of UML often substitute an activity diagram. In the realm of data-flow plans, site-oriented data-flow plans are a subset. Identical nodes in a data-flow diagram and a Petri net can be thought of as inverted counterparts since the semantics of data memory are represented by the locations in the network. Structured data modeling (DFM) includes processes, flows, storage, and terminators.

Data Flow Diagram Symbols

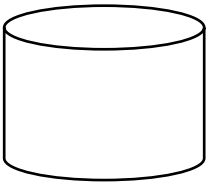
Process

A process is one that takes in data as input and returns results as output.



Data Store

In the context of a computer system, the term "data stores" is used to describe the various memory regions where data can be found. In other cases, "files" might stand in for data.



Data Flow

Data flows are the pathways that information takes to get from one place to another. Please describe the nature of the data being conveyed by each arrow.

External Entity

In this context, "external entity" refers to anything outside the system with which the system has some kind of interaction. These are the starting and finishing positions for inputs and outputs, respectively.

DATA FLOW DIAGRAM

The whole system is shown as a single process in a level DFD. Each step in the system's assembly process, including all intermediate steps, are recorded here. The "basic system model" consists of this and 2-level data flow diagrams.

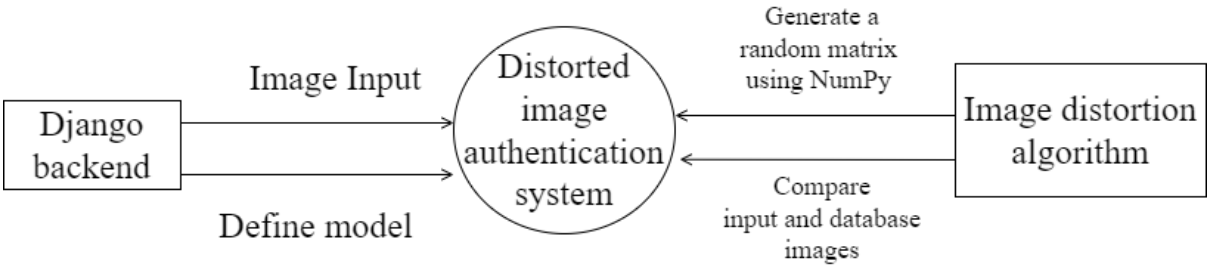


Fig 4.2.1 – Data Flow Diagram Level 0

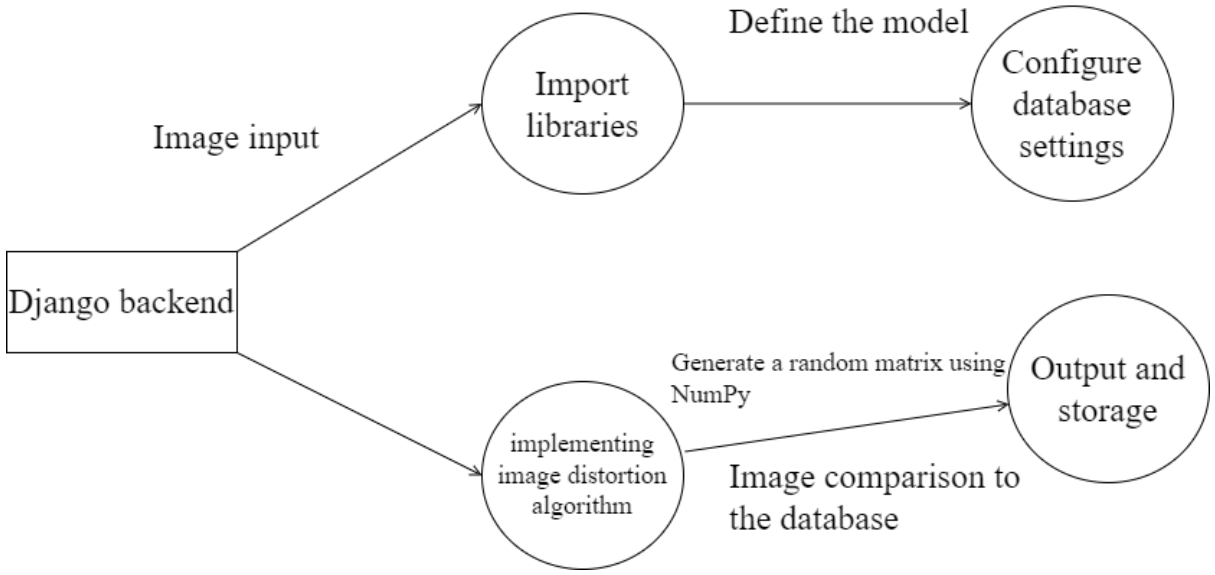


Fig 4.2.2 – Data Flow Diagram Level 1

4.3 ENTITY RELATIONSHIP DIAGRAM

- **Definition**

The relationships between database entities can be seen using an entity-relationship diagram (ERD). The entities and relationships depicted in an ERD can have further detail added to them via data object descriptions. In software engineering, conceptual and abstract data descriptions are represented via entity-relationship models (ERMs). Entity-relationship diagrams (ERDs), entity-relationship diagrams (ER), or simply entity diagrams are the terms used to describe the resulting visual representations of data structures that contain relationships between entities. As such, a data flow diagram can serve dual purposes. To demonstrate how data is transformed across the system. To provide an example of the procedures that affect the data flow.

1. One-to-One

Whenever there is an instance of entity (A), there is also an instance of entity (B) (B). In a sign-in database, for instance, only one security mobile number (S) is associated with each given customer name (A) (B).

2. One-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

For a corporation whose employees all work in the same building, for instance, the name of the building (A) has numerous individual associations with employees (B), but each of these B's has only one individual link with entity A.

3. Many-to-Many

For each instance of entity B, there is exactly one occurrence of entry A, regardless of how many instances of entity B there are.

In a corporation where everyone works out of the same building, entity A is associated with many different Bs, but each B has only one A.

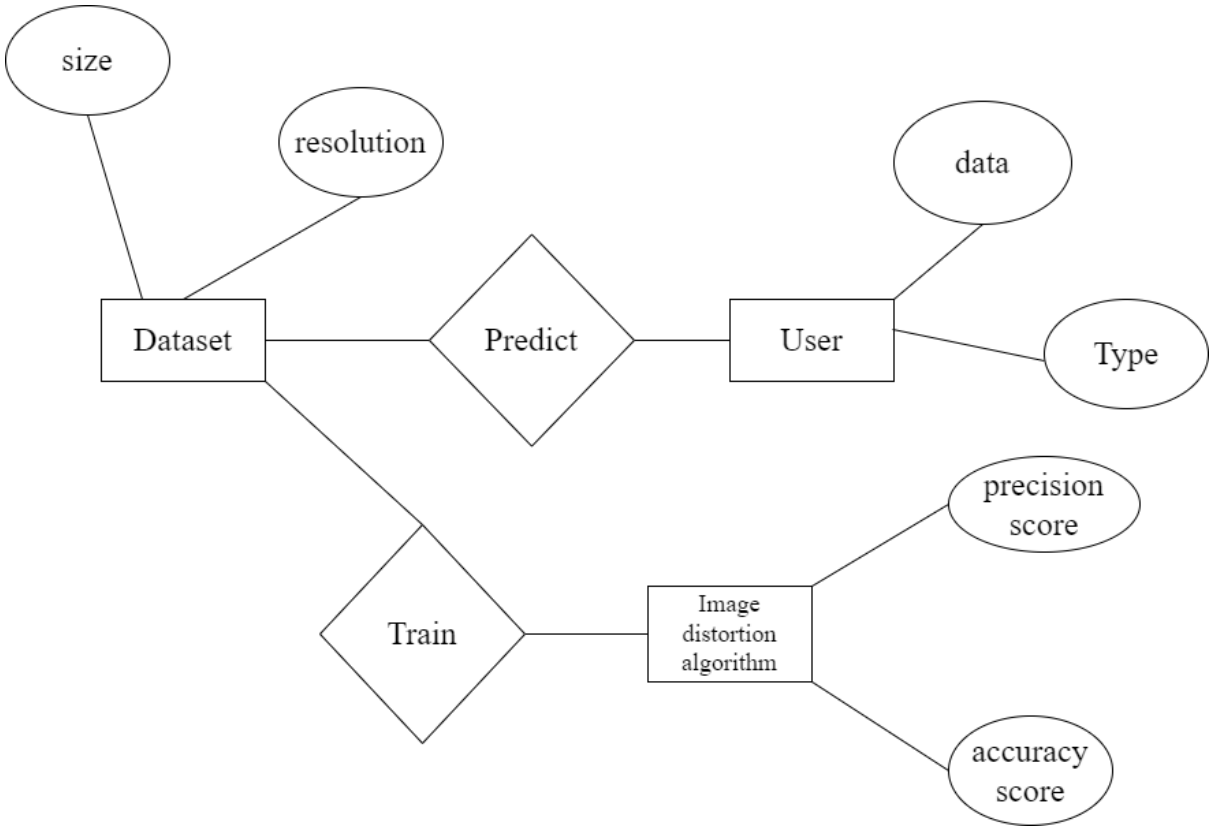


Fig 4.3.1– Entity Relationship Diagram

4.4 USE-CASE DIAGRAM

The possible interactions between the user, the dataset, and the algorithm are often depicted in a use case diagram. It's created at the start of the procedure.

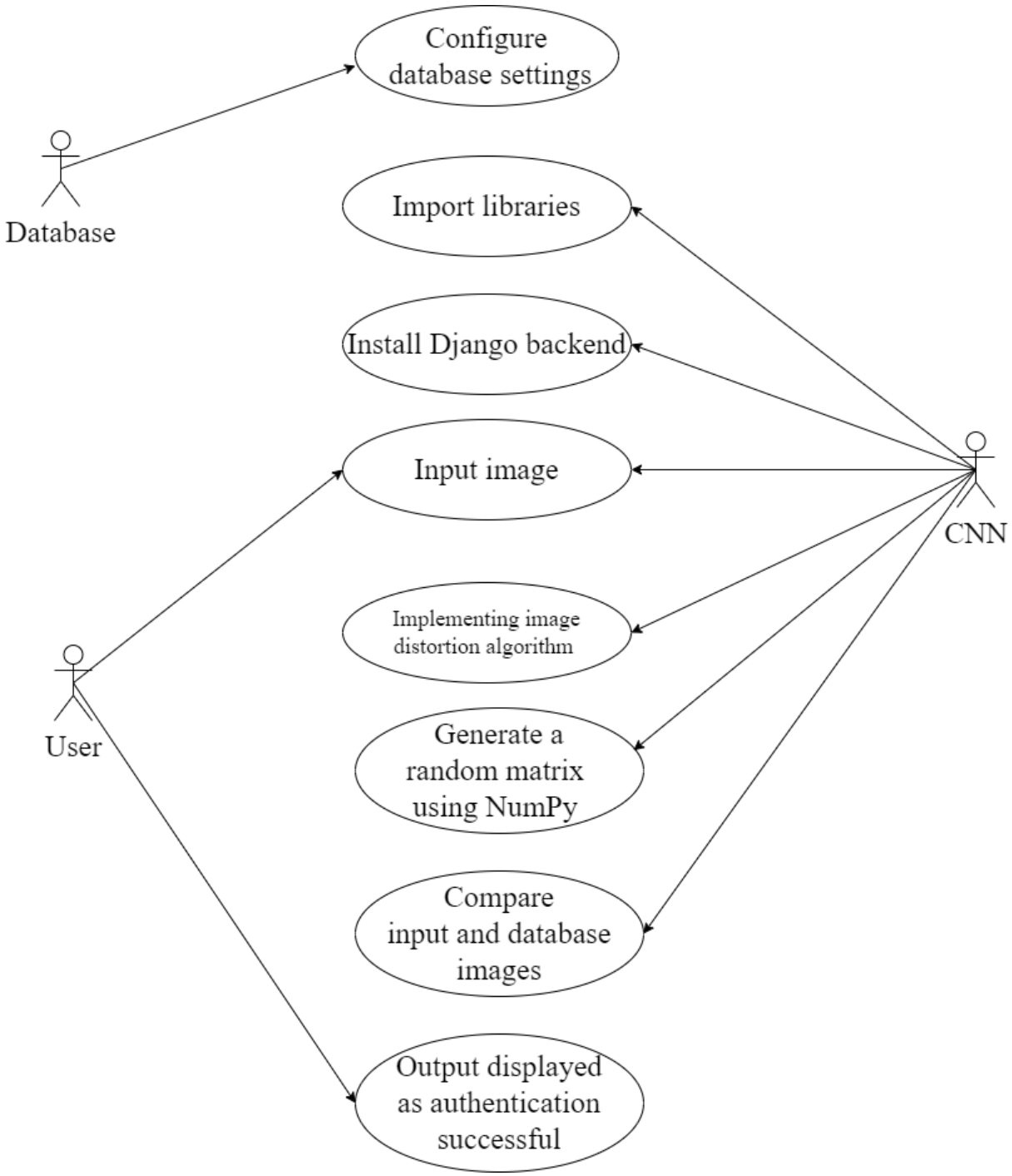


Fig 4.4.1 – Use-Case Diagram

4.5 ACTIVITY DIAGRAM

An activity diagram, in its most basic form, is a visual representation of the sequence in which tasks are performed. It depicts the sequence of operations that make up the overall procedure. They are not quite flowcharts, but they serve a comparable purpose.

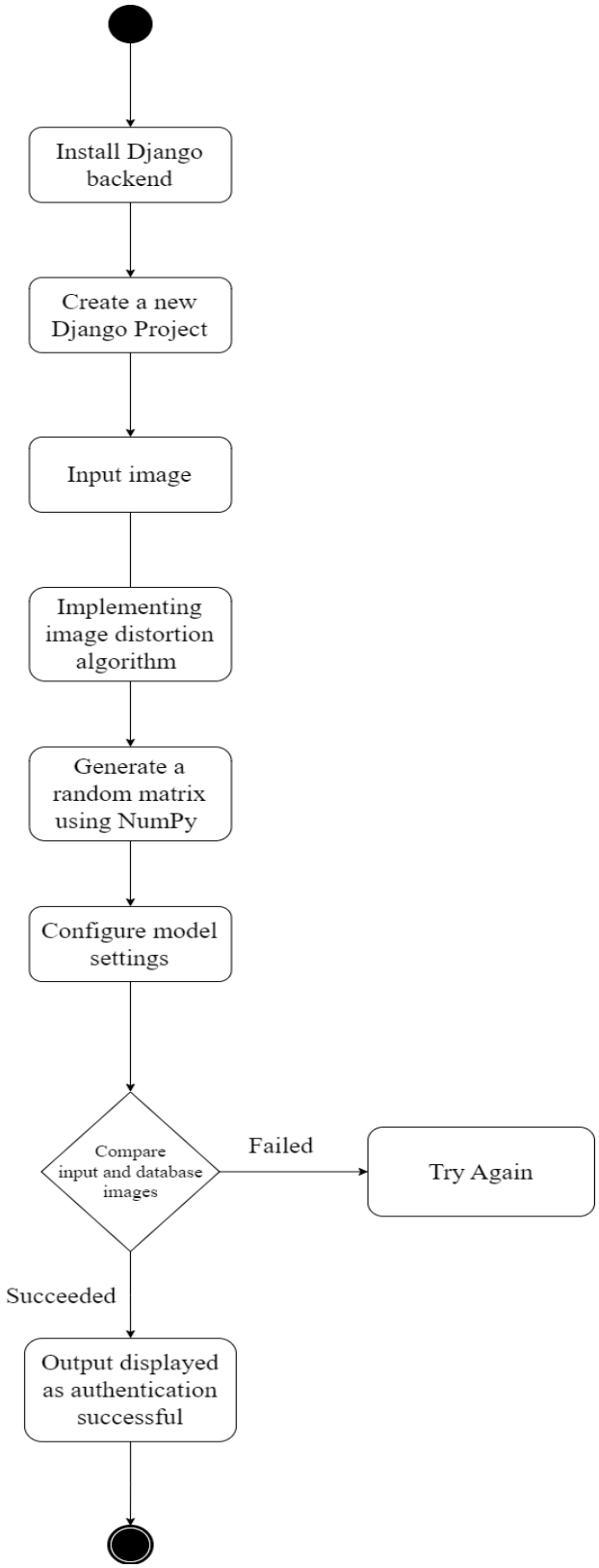


Fig 4.5.1 – Activity Diagram

4.6 SEQUENCE DIAGRAM

These are another type of interaction-based diagram used to display the workings of the system. They record the conditions under which objects and processes cooperate.

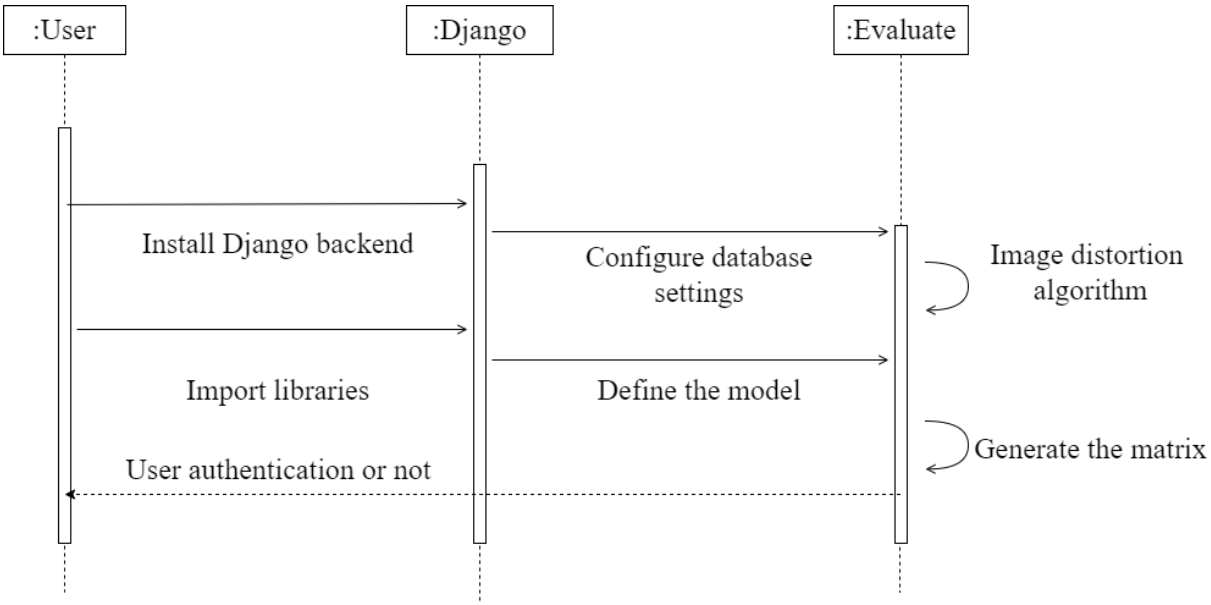


Fig 4.6.1 – Sequence Diagram

4.7 CLASS DIAGRAM

In essence, this is a "context diagram," another name for a contextual diagram. It simply stands for the very highest point, the 0 Level, of the procedure. As a whole, the system is shown as a single process, and the connection to externalities is shown in an abstract manner.

- A + indicates a publicly accessible characteristic or action.
- A - a privately accessible one.
- A # a protected one.
- A - denotes private attributes or operations.

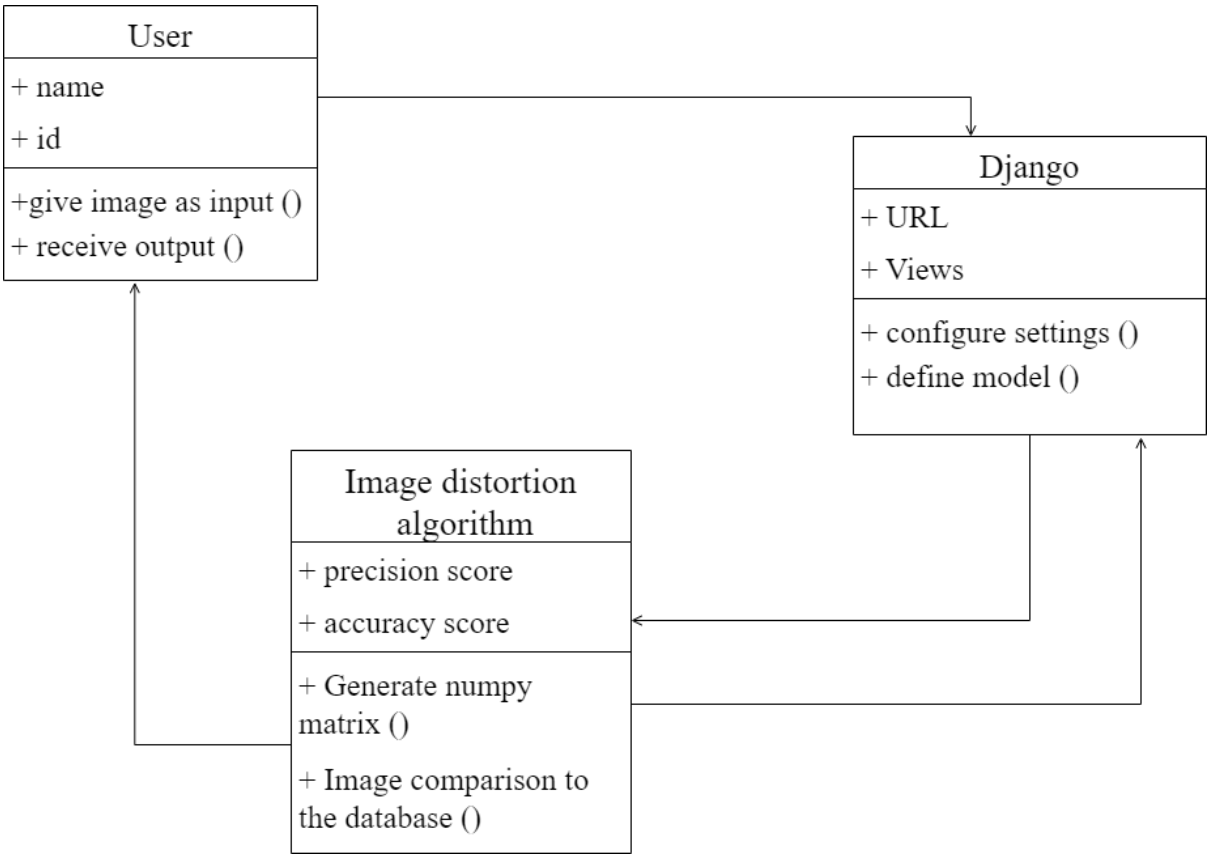


Fig 4.7.1 – Class Diagram

CHAPTER 5
MODULE DESCRIPTION

5.1 MODULE 1: BUILDING THE BACKEND

The first module of the proposed EYEDi graphical authentication scheme involves building the web application backend using Python Django. Django is a high-level Python web framework that provides an inbuilt authentication system that can be extended to include additional fields required for image-based authentication.

The web application backend is responsible for managing user authentication, storing user data, and generating unique, encodable distorted images for each authentication attempt. We will extend the User model with additional fields required for storing the distorted image and the selected portion of the image selected during authentication.

To build the backend, we will first create a Django project using the 'django-admin' command. Next, we will create a Django app using the 'python manage.py startapp' command. The app will contain the logic and models required for authentication, storing user data, and generating encodable distorted images.

In the 'models.py' file, we will extend the User model provided by Django's auth models by creating a UserProfile model that contains the additional fields required for image-based authentication. The UserProfile model will contain fields for storing the encodable distorted image and the selected portion of the image selected during authentication.

We will also create views for user registration, login, and authentication. During registration, the user will be presented with a distorted image and asked to select a unique portion of the image. The selected portion will be stored along with the distorted image in the user's profile.

During login, the user will enter their username and password, and the authentication system will generate a new distorted image using the image distortion algorithm.

During authentication, the user will be presented with a new distorted image and asked to select the same unique portion of the image that was selected during registration. If the selected portions match, the user will be authenticated and granted access to the system.

To store the distorted image and selected portion of the image, we will use Django's file storage system. The distorted image will be stored in a directory specified in the 'settings.py' file, and the selected portion of the image will be stored as a string in the UserProfile model.

The web application backend will also contain the logic required for generating unique, encodable distorted images for each authentication attempt. This logic will be implemented in the second module of the proposed system, which involves building the image distortion algorithm using OpenCV library in Python.

5.2 MODULE 2: BUILDING THE IMAGE DISTORTION ALGORITHM

The second module of the proposed EYEDi graphical authentication scheme involves building the image distortion algorithm using OpenCV library in Python.

The algorithm applies image distortion and rotation using NumPy arrays to create a unique, encodable distorted image for each authentication attempt.

The algorithm generates two NumPy arrays, 'map_x' and 'map_y,' of size (rows, cols) with a data type of float32. These arrays are used to compute the new x and y pixel locations for each pixel in the image. The formula used to compute these new locations is a sinusoidal function that modulates the x and y coordinates with a sine function. The result is a sinusoidal shift in the x and y directions, with an amplitude of 25 pixels and a frequency of 1/5 pixels.

The code loop that computes the new x and y pixel locations for each pixel in the image is as follows:

map_x = np.zeros((rows, cols), dtype=np.float32)

map_y = np.zeros((rows, cols), dtype=np.float32)

for i in range(rows):

for j in range(cols):

map_x[i, j] = j + 25np.sin(i/5)

map_y[i, j] = i + 25np.sin(j/5)

The 'map_x' and 'map_y' arrays are then applied to the original image using the 'cv2.remap' function. The 'cv2.remap' function takes three arguments: the original image, the 'map_x' array, and the 'map_y' array. The interpolation method used here is 'INTER_LINEAR,' which provides a smooth transition between the pixels.

img = cv2.remap(img, map_x, map_y, cv2.INTER_LINEAR)

The result is a new image with a sinusoidal shift in the x and y directions. The image is unique and cannot be easily replicated or reused.

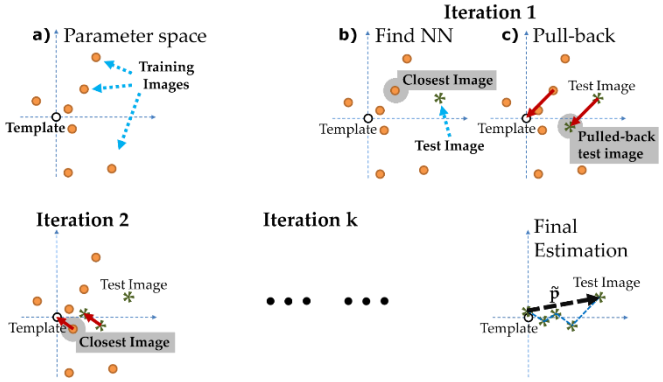


Fig 2:Image Distortion Algorithm

The image distortion algorithm is crucial to the proposed EYEDi graphical authentication scheme as it provides a way to generate unique, encodable distorted images for each authentication attempt. The algorithm applies image distortion and rotation using NumPy arrays and a sinusoidal function to create a unique image that cannot be easily replicated or reused.

Overall, building the image distortion algorithm is a crucial step in the proposed EYEDi graphical authentication scheme. The algorithm generates unique, encodable distorted images for each authentication attempt, making it resistant to screenshot attacks. The algorithm is implemented using OpenCV library in Python and applies image distortion and rotation using NumPy arrays and a sinusoidal function. By building the image distortion algorithm, we can ensure that the authentication system is secure, reliable, and easy to use.

5.3 MODULE 3: BUILDING THE AUTHENTICATION SYSTEM

The third module of the proposed EYEDi graphical authentication scheme involves building the authentication system that uses the encodable distorted image

instead of a password. We will use the extended User model from Django's auth models to store the distorted image for each user, along with the selected portion of the image selected during authentication.

The authentication system compares the selected portion of the encodable distorted image with the original image to verify the user's identity. The authentication process involves the following steps:

Registration: During registration, the user selects a unique portion of the encodable distorted image. This unique portion is stored along with the distorted image in the user's profile.

Login: During login, the user enters their username and password. The authentication system generates a new distorted image using the image distortion algorithm.

Authentication: During authentication, the user is presented with a new distorted image and asked to select the same unique portion of the image that was selected during registration. The authentication system compares the selected portion of the new distorted image with the selected portion of the original image stored in the user's profile. If the two portions match, the user is authenticated and granted access to the system.

To build the authentication system, we will use the extended User model from Django's auth models to store the distorted image and selected portion of the image. We will also create views for user registration, login, and authentication.

During registration, the user will be presented with a distorted image and asked to select a unique portion of the image. The selected portion will be stored along with the distorted image in the user's profile. To ensure that the selected portion is unique and cannot be easily replicated, we will use a combination of randomization and

user input. For example, we can randomly select a portion of the image and ask the user to select another portion that is adjacent to the randomly selected portion.

During login, the user will enter their username and password, and the authentication system will generate a new distorted image using the image distortion algorithm. To ensure that each authentication attempt generates a unique distorted image, we will use a combination of randomization and encryption. For example, we can encrypt the distorted image using a randomly generated key that is unique to each authentication attempt.

During authentication, the user will be presented with a new distorted image and asked to select the same unique portion of the image that was selected during registration. The authentication system will compare the selected portion of the new distorted image with the selected portion of the original image stored in the user's profile. If the two portions match, the user will be authenticated and granted access to the system.

To enhance the security of the authentication system, we can implement additional measures such as rate limiting, IP blocking, and two-factor authentication. Rate limiting can be used to prevent brute-force attacks, while IP blocking can be used to block malicious IP addresses. Two-factor authentication can be used to add an additional layer of security to the authentication process.

Overall, building the authentication system is a crucial step in the proposed EYEDi graphical authentication scheme. By using the extended User model from Django's auth models, we can store the distorted image and selected portion of the image, and compare them during authentication to verify the user's identity. By implementing additional security measures, we can enhance the security of the authentication system and prevent unauthorized access to sensitive information.

5.4 MODULE 4: BINDING THE MODULES TOGETHER AND BUILDING A USER INTERFACE

The final module of the proposed EYEDi graphical authentication scheme involves binding the authentication system and the image distortion algorithm together to build the complete system. We will also build a simple user interface that allows users to register, login, and authenticate using the encodable distorted image.

The authentication system and the image distortion algorithm will be integrated into the web application backend using Python code. The authentication system will retrieve the encodable distorted image and selected portion of the image from the user's profile and pass it to the image distortion algorithm to generate a new distorted image.

The user interface will be built using HTML, CSS, and JavaScript. The user interface will be designed to be user-friendly and intuitive, making it easy for users to register, login, and authenticate using the encodable distorted image.

During registration, the user will be presented with a distorted image and asked to select a unique portion of the image. The selected portion will be stored along with the distorted image in the user's profile. The user interface will provide clear instructions on how to select the unique portion of the image and will highlight the importance of selecting a portion that is not easily guessable.

During login, the user will enter their username and password, and the authentication system will generate a new distorted image using the image distortion algorithm. The user interface will display the new distorted image along with instructions on how to select the same unique portion of the image that was selected during registration.

During authentication, the user will select the same unique portion of the new distorted image that was selected during registration. The authentication system will compare the selected portion of the new distorted image with the selected portion of the original image stored in the user's profile. If the selected portions match, the user will be authenticated and granted access to the system.

The user interface will also provide feedback to the user during registration, login, and authentication. If the user selects an easily guessable portion of the image or fails to select the same unique portion during authentication, the user interface will provide an error message and prompt the user to try again.

The user interface will be designed to be responsive and accessible on a wide range of devices, including desktops, laptops, tablets, and smartphones. The user interface will use modern web technologies, such as HTML5, CSS3, and JavaScript, to provide a seamless and engaging user experience.

Overall, binding the modules together and building a user interface is a crucial step in the proposed EYEDi graphical authentication scheme. The user interface is the primary point of contact between the user and the authentication system and must be intuitive, user-friendly, and accessible. By building a simple and effective user interface, we can ensure that the authentication system is easy to use and can be adopted by a wide range of users.

CHAPTER 6

TESTING

Discovering and fixing such problems is what testing is all about. The purpose of testing is to find and correct any problems with the final product. It's a method for evaluating the quality of the operation of anything from a whole product to a single component. The goal of stress testing software is to verify that it retains its original functionality under extreme circumstances. There are several different tests from which to pick. Many tests are available since there is such a vast range of assessment options.

Who Performs the Testing: All individuals who play an integral role in the software development process are responsible for performing the testing. Testing the software is the responsibility of a wide variety of specialists, including the End Users, Project Manager, Software Tester, and Software Developer.

When it is recommended that testing begin: Testing the software is the initial step in the process. begins with the phase of requirement collecting, also known as the Planning phase, and ends with the stage known as the Deployment phase. In the waterfall model, the phase of testing is where testing is explicitly arranged and carried out. Testing in the incremental model is carried out at the conclusion of each increment or iteration, and the entire application is examined in the final test.

When it is appropriate to halt testing: Testing the programme is an ongoing activity that will never end. Without first putting the software through its paces, it is impossible for anyone to guarantee that it is completely devoid of errors. Because the domain to which the input belongs is so expansive, we are unable to check every

single input.

6.1 TYPES OF TESTING

There are four types of testing:

Unit Testing

The term "unit testing" refers to a specific kind of software testing in which discrete elements of a program are investigated. The purpose of this testing is to ensure that the software operates as expected.

Test Cases

1. Test that the encoding function correctly generates a distorted image based on the input image and a set of random distortions and rotations.
2. Test that the decoding function can correctly extract the original image from a distorted image.
3. Test that the encoding and decoding functions can correctly handle images of different sizes and aspect ratios.

Integration Testing

The programme is put through its paces in its final form, once all its parts have been combined, during the integration testing phase. At this phase, we look for places where interactions between components might cause problems.

Test Cases

1. Test that the system can properly integrate with the database or data store used to store user credentials and encoded images.
2. Test that the system can handle multiple concurrent requests for authentication without any performance or security issues.

3. Test that the system can handle different types of user authentication methods (e.g. username and password, biometric authentication) and integrate with those systems appropriately.

Functional Testing

One kind of software testing is called functional testing, and it involves comparing the system to the functional requirements and specifications. In order to test functions, their input must first be provided, and then the output must be examined. Functional testing verifies that an application successfully satisfies all of its requirements in the correct manner. This particular kind of testing is not concerned with the manner in which processing takes place; rather, it focuses on the outcomes of processing. Therefore, it endeavours to carry out the test cases, compare the outcomes, and validate the correctness of the results.

Test Cases

1. Test that the system is able to encode an image into a distorted image with specified level of distortion and rotation.
2. Test that the system is able to decode the distorted image and retrieve the original image.
3. Test that the system is able to authenticate a user by comparing the provided image with the stored encoded image of the user.

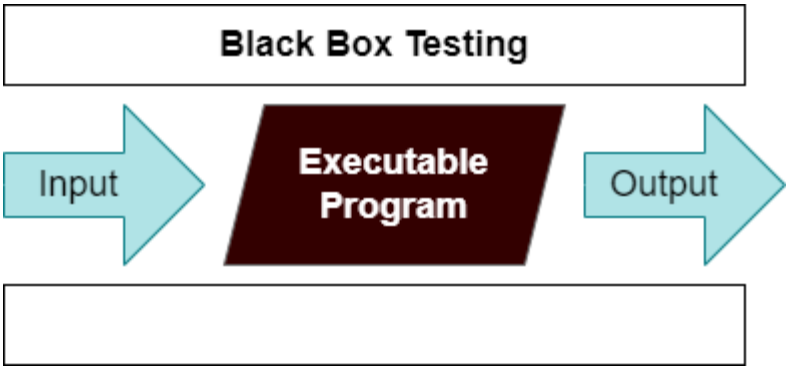
6.2 TESTING TECHNIQUES

There are many different techniques or methods for testing the software, including the following:

BLACK BOX TESTING

During this kind of testing, the user does not have access to or knowledge of the

internal structure or specifics of the data item being tested. In this method, test cases are generated or designed only based on the input and output values, and prior knowledge of either the design or the code is not necessary. The testers are just conscious of knowing about what is thought to be able to do, but they do not know how it is able to do it.



For example, without having any knowledge of the inner workings of the website, we test the web pages by using a browser, then we authorise the input, and last, we test and validate the outputs against the intended result.

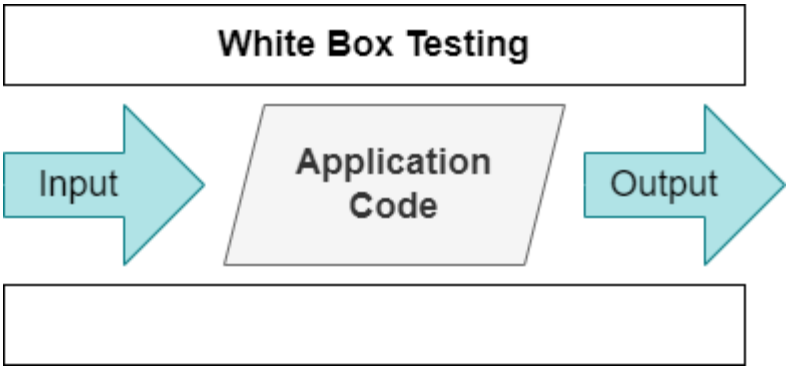
Test Cases

1. Test that the system properly handles inputs of different file formats (e.g. JPEG, PNG, BMP).
2. Test that the system can handle images of different sizes and resolutions, including very large or very small images.
3. Test that the system properly handles images with different color depths (e.g. grayscale vs. RGB).

WHITE BOX TESTING

During this kind of testing, the user is aware of the internal structure and details of

the data item, or they have access to such information. In this process, test cases are constructed by referring to the code. Programming is extremely knowledgeable of the manner in which the application of knowledge is significant. White Box Testing is so called because, as we all know, in the tester's eyes it appears to be a white box, and on the inside, everyone can see clearly. This is how the testing got its name.



As an instance, a tester and a developer examine the code that is implemented in each field of a website, determine which inputs are acceptable and which are not, and then check the output to ensure it produces the desired result. In addition, the decision is reached by analyzing the code that is really used.

Test Cases

1. Test that the encoding function is properly implemented and generates a unique encoded image for each input image.
2. Test that the decoding function is properly implemented and is able to extract the original image from a distorted image.
3. Test that the authentication function is properly implemented and is able to accurately match a user-provided image with the stored encoded image of the user.

CHAPTER 7
CONCLUSION

7.1 CONCLUSION

In conclusion, the proposed EYEDi graphical authentication scheme provides an innovative and secure solution for user authentication. By using image distortion and rotation algorithms, we can generate unique, encodable distorted images for each authentication attempt, making it difficult for attackers to replicate or guess the correct authentication credentials. The proposed system involves four modules, including building the backend, building the image distortion algorithm, building the authentication system, and binding the modules together and building a user interface. Each module plays a crucial role in the overall functioning of the system and ensures that the system is secure, reliable, and easy to use. Building the backend using Python Django provides a robust framework for managing user authentication and storing user data. The image distortion algorithm, implemented using OpenCV library in Python, provides a method for generating unique, encodable distorted images for each authentication attempt. The authentication system, integrated into the backend, compares the selected portion of the encodable distorted image with the original image to verify the user's identity. Finally, the user interface provides an intuitive and user-friendly way for users to register, login, and authenticate using the encodable distorted image. The proposed EYEDi graphical authentication scheme has several advantages over traditional password-based authentication systems. It eliminates the need for users to remember complex passwords, making it easier for users to access the system. It also provides a higher level of security, making it difficult for attackers to guess or replicate the correct authentication credentials. However, the proposed system also has some

limitations. It requires users to have access to a device with a camera to take a picture of the selected portion of the encodable distorted image. It also requires users to select a unique and non-guessable portion of the image, which may be challenging for some users.

CHAPTER 8

APPENDIX 1

CODING

```
from django.shortcuts import render, redirect
from django.http import HttpResponse, HttpResponseNotFound
from django.contrib.auth.models import User
import cv2
import os
from .models import ImagePart, Account
import numpy as np
# Create your views here.

def home(request):
    if request.method == 'POST':
        username = request.POST['username']
        passImage = request.FILES['passImage']
        with open('media/temp_images/temp.jpg', 'wb') as f:
            f.write(passImage.read())
        os.mkdir(f'media/stored_img/{username}_image')
        img = cv2.imread('media/temp_images/temp.jpg')
        img = cv2.resize(img, (625, 625))
        rows, cols, _ = img.shape
        map_x = np.zeros((rows, cols), dtype=np.float32)
        map_y = np.zeros((rows, cols), dtype=np.float32)
        for i in range(rows):
            for j in range(cols):
                map_x[i, j] = j + 25*np.sin(i/5)
                map_y[i, j] = i + 25*np.sin(j/5)

# Apply the distortion map to the image
        img = cv2.remap(img, map_x, map_y, cv2.INTER_LINEAR)
        h, w = img.shape[:2]
        piece_w = w//5
        piece_h = h//5
        k = 0
        part_add = []
        for i in range(5):
            for j in range(5):
                y1 = i*piece_h
                x1 = j*piece_w
                y2 = (i+1)*piece_h
```

```
                x2 = (j+1)*piece_w
                img_ = img[y1:y2, x1:x2]
                filename = f'media/stored_img/{username}_image/part_{k}.jpg'
                cv2.imwrite(filename, img_)
                part_add.append(f'/media/stored_img/{username}_image/part_{k}.jpg
')
                k+=1
        data = {'parts':part_add}
        ImagePart.objects.create(username = username, parts = data)
        request.session['username'] = username
        return redirect('register')
    return render(request, 'imageupload.html')



def register(request):
    context = {}
    username = request.session.get('username')
    ins = ImagePart.objects.get(username = username)
    all_part_addresses = ins.parts
    all_addresses = all_part_addresses['parts']
    context['parts'] = all_addresses
    if request.method == 'POST':
        username = request.session.get('username')
        imageVal = request.POST['selected-image']
        user = User.objects.create(username = username,password =
'random###1234')
        Account.objects.create(user=user,value = imageVal)
        return HttpResponse('Registration SuccessFul')
    return render(request,'register.html',context)

def user_login_s1(request):
    context = {}
    if request.method == 'POST':
        username = request.POST['username']
        image_parts = ImagePart.objects.get(username = username)
        all_part_addresses = image_parts.parts
        context['parts'] = all_part_addresses
        request.session['username'] = username
        return redirect('login2')
    return render(request, 'login.html')

def user_login_s2(request):
    context = {}
    username = request.session.get('username')
    ins = ImagePart.objects.get(username = username)
```

```
    all_part_addresses = ins.parts
    all_addresses = all_part_addresses['parts']
    context['parts'] = all_addresses
    if request.method == 'POST':
        username = request.session.get('username')
        print(username)
        imageVal = request.POST['selected-image']
        user_ins = User.objects.get(username=username)
        print(user_ins)
        try:
            ins = Account.objects.get(user = user_ins, value=imageVal)
        except Account.DoesNotExist:
            return HttpResponseNotFound("Wrong Password")
        if ins != None:
            return redirect("https://www.youtube.com")
        else:
            return HttpResponse("Invalid User system")
    return render(request, 'logins2.html', context)
```

CHAPTER 9

APPENDIX 2

SCREEN SHOTS

Fig 9.1 – Original Image

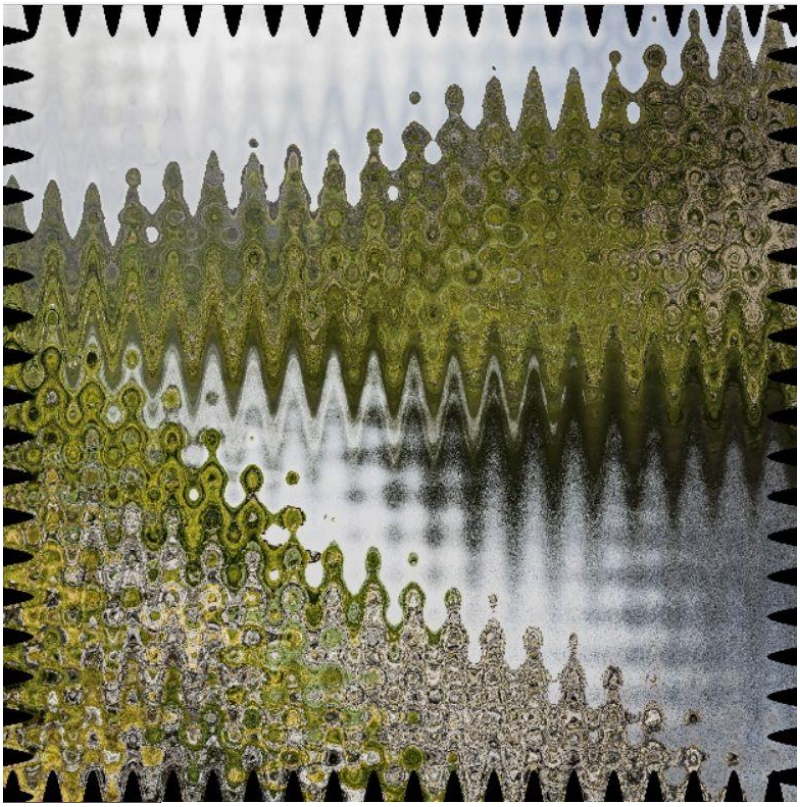


Fig 9.2 Distorted Image

CHAPTER 10
REFERENCES

1. Bai, Y., Li, S., & Liu, S. (2020). A review of graphical passwords authentication systems. Journal of Ambient Intelligence and Humanized Computing, 11(9), 4003-4016.

2. Biddle, R., Chiasson, S., & Van Oorschot, P. C. (2012). Graphical passwords: Learning from the first twelve years. ACM Computing Surveys (CSUR), 44(4), 19.

3. Cao, Q., Liu, X., Chen, J., & Tan, J. (2017). A survey on graphical passwords. Journal of Computer Science and Technology, 32(2), 235-257.

4. Das, S., & Maiti, S. (2019). Secure and usable graphical password authentication scheme. Computers & Security, 83, 276-295.

5. Das, S., Maiti, S., & Paul, T. (2021). An effective authentication scheme based on image and graphical passwords. Journal of Ambient Intelligence and Humanized Computing, 12(6), 6549-6568.

6. De Luca, A., Hang, A., Brudy, F., & Lindner, M. (2015). Towards a user-centered framework for developing secure graphical passwords. In Proceedings of the 2015 Symposium on Usable Privacy and Security (pp. 285-298).

7. Dhamija, R., & Perrig, A. (2000). Déjà Vu: A user study using images for authentication. In Proceedings of the 9th USENIX Security Symposium (pp. 7-7).

8. Dhamija, R., Perrig, A., & Hearst, M. (2004). Deconstructing web page authentication. In Proceedings of the 10th ACM Conference on Computer and Communications Security (pp. 290-299).

9. Hong, J., & Kim, M. (2020). Personalized graphical password authentication using images. Journal of Ambient Intelligence and Humanized Computing, 11(10), 4719-4730.

10. Kaur, H., & Kumar, N. (2017). A review on graphical password authentication schemes. Journal of Theoretical and Applied Information Technology, 95(6), 1236-1251.

11. Li, X., & Chen, J. (2014). Authentication of mobile phone users based on graphical passwords. Journal of Network and Computer Applications, 43, 47-57.

12. Mazhar, S., Hasan, S. S., & Ahmad, N. (2020). A novel image-based graphical password authentication scheme. Journal of Ambient Intelligence and Humanized Computing, 11(9), 4247-4260.

13. Monrose, F., & Rubin, A. D. (1997). Password hardening based on keystroke dynamics. In Proceedings of the 4th ACM Conference on Computer and Communications Security (pp. 73-82).

14. Narayan, R., & Singh, P. (2018). Graphical password authentication using color QR codes. Journal of Ambient Intelligence and Humanized Computing, 9(1), 35-43.

15. Singh, S., Sood, S. K., & Tyagi, A. (2019). A survey of graphical password authentication schemes. Journal of Ambient Intelligence and Humanized Computing, 10(6), 2217-2244.

16. Wu, L., Chen, H., & Chen, Y. (2018). A graphical password authentication system based on convolutional neural network. IEEE Access, 6, 12294-12304

17. Wu, Y., Wang, Y., Wang, L., & Liu, L. (2021). An image-based graphical password authentication scheme with user customization. Journal of Ambient Intelligence and Humanized Computing, 12(7), 7293-7305.

18. Yu, S., Peng, S., Lu, Y., & Wang, K. (2020). An improved image-based graphical password authentication scheme using bilinear interpolation. Journal of Ambient Intelligence and Humanized Computing, 11(12), 5485-5495.

19. Zhou, Y., Yu, Y., & Jin, L. (2020). An image-based graphical password authentication scheme using distortion-based image selection. IEEE Access, 8, 36862-36872.

20. Wang, Z., Chen, Y., & Liu, Y. (2019). An image-based graphical password authentication scheme with diverse image categories. IEEE Access, 7, 109248-109262.