

FACE LIVENESS DETECTION SYSTEM

Final Year Project Report

By

Arooba Siddiqi

Emaan Bashir

Rimsha Mirza

In Partial Fulfillment

Of the Requirements for the degree

Bachelors of Engineering in Software Engineering (BESE)

School of Electrical Engineering and Computer Science

National University of Sciences and Technology

Islamabad, Pakistan

(2023)

DECLARATION

We hereby declare that this project report entitled “**Face liveness Detection System**” submitted to the “SEECS”, is a record of an original work done the guidance of supervisor “Dr. Seemab Latif” and that no part has been plagiarized without citations. Also, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Software Engineering.

Team Members

Signature

Arooba Siddiqi

Emaan Bashir

Rimsha Mirza

Supervisor

Signature

Dr. Seemab Latif

Date:

May 19th, 2023

Place:

SEECS, NUST, H-12 ISLAMABAD

DEDICATION

To Allah the Almighty and Exalted

&

To our Family, Mentors and Faculty

ACKNOWLEDGEMENTS

This project is completed with the mentorship of our respected advisor and co-advisor, **“Dr. Seemab Latif”** and **“Dr. Syed Imran Ali”**. We express our indebted gratitude and special thanks to them who guided us throughout the project, despite their busy schedules. We would like to express deepest appreciation to our university, **“National University of Science and Technology (NUST)”**, school **“School of Electrical Engineering and Computer Science (SEECS)”**, and department **“Department of Computing (DOC)”**, in honing us to be globally competitive by equipping us with the knowledge and skills to make a lasting positive impact in the world.

TABLE OF CONTENT

Declaration	2
Dedication	3
Acknowledgements	4
Table Of Content	5
List Of Figures	7
List Of Tables	8
Abstract	9
1.Introduction.....	10
2.Literature Review.....	12
3.Problem Definition.....	15
3.1 Problem Statement	15
3.1.1 Who Needs It	15
4.Software Requirements Specifications	16
4.1 Face Detection And Feature Extraction.....	16
4.1.1 Description And Priority	16
4.1.2 Stimulus/Response Sequences	16
4.1.3 Functional Requirements	16
4.2 Process The Input And Generate Result	16
4.2.1 Description And Priority	16
4.2.2 Stimulus/Response Sequences	16
4.2.3 Functional Requirements	17
4.3 Other Nonfunctional Requirements.....	18
4.3.1 Performance Requirements	18
4.3.2 Security Requirements.....	18
4.3.3 Software Quality Attributes	18
5.Methodology	19
5.1 Data Collection.....	19
5.2 Eyeball Movement Analysis	20
5.3 Texture Analysis.....	20
5.4 Blood Flow Analysis	21

6.Detailed Design And Architecture	22
6.1 Software Design Model.....	22
6.2 System Overview	23
6.2.1 Architectural Design	23
6.3 Design Models.....	24
6.3.1 Sequence Diagram	24
6.3.2 Activity Diagram	26
6.3.3 Data Flow Diagram.....	27
6.4 Data Design	28
6.4.1 Data Dictionary	28
6.5 User Interface Design.....	29
6.5.1 Home Page:	29
6.5.2 Demonstration Page:	30
6.5.3 Services Page	31
6.5.4 Use Case Page.....	32
6.5.5 About Page.....	32
6.5.6 Contact Page	33
6.5.7 Screen Objects And Actions.....	33
7.Implementation And Testing.....	35
7.1 Implementation.....	35
7.2 Testing	36
8.Results And Discussion.....	37
9.Conclusion And Future Work	45
10.References.....	46

LIST OF FIGURES

1. Use Case Diagram	18
2. Data Collection Process	20
3. Eyeball Liveness Technique Process	21
4. Texture Analysis Process	21
5. Blood Flow Analysis Process	22
6. Software Design Model for Face liveness Detection System	23
7. System Architecture	24
8. Sequence Diagram	25
9. Activity Diagram	26
10. Data Flow Diagram	27
11. Home Page User Interface	28
12. Demonstration Page User Interface	29
13. Services Page User Interface	30
14. Use Case Page User Interface	31
15. About Page User Interface	31
16. Contact Page User Interface	32
17. Technologies Stack	34
18. User Feedback	36
19. Liveness Algorithms Results	42
20. Eyeball Detection Results	43

LIST OF TABLES

1. Categories of Liveness Detection Techniques	11
2. Eyeball Movement Applicability	35
3. Texture Algorithm – SVM Classifier	36
4. Texture Algorithm – KNN Classifier	36
5. Texture Algorithm – Decision Tree Classifier	37
6. Texture Algorithm – Random Forest Classifier	37
7. Texture Algorithm – Random Forest Classifier (Component Color Spaces) ..	38
8. Texture Algorithm –Finalized Model	38
9. Blood flow Algorithm	39

ABSTRACT

Face liveness detection is an essential task to combat spoofing attacks in biometric systems. In this project, three algorithms (i.e., eyeball movement, texture analysis and blood flow analysis) are used to develop a face liveness detection system. The system uses various patterns in a person's facial characteristics to distinguish between real and fake people. The system works by extracting frames from a real-time video. Eyeball movement analysis follows the face in real-time and locate the iris in relation to the shapes being presented. The texture analysis technique translates video into certain color spaces and generates local binary pattern histograms for classification. The blood flow analysis technique determines the heart rate by calculating the mean color values, applying a bandpass filter and extracting the amplitude spectrum. The system was created by following a waterfall design model. It has a user-friendly interface with numerous pages like 'Home', 'Services', 'Demonstration', 'Use cases', 'About', and 'Contact'. Real-time testing on various attack scenarios, such as replay videos and printed images, is done during the implementation and testing phases. The results demonstrate the extent to which the blood flow and texture analysis algorithms detect spoofing attempts. The suggested technique demonstrates promising results in detecting various types of spoofing attempts such as printed images, replay videos and digital screen images. The technique, however, has shortcomings when it comes to spotting 3D spoofing attempts. Future plans include increasing the dataset, taking into account local demographics, and investigating ANN approaches to improve the accuracy and reliability of the system. This Face liveness Detection System advances the area of biometric security and establishes the foundation for new developments in spoofing attack detection and prevention.

INTRODUCTION

Financial crimes have been around since the invention of currency. Criminal groups make billions of dollars off financial crimes. The international community has prioritized the battle against financial crime, paying particular attention to money laundering and the funding of terrorism, as a result of the effects that this type of crime has on the economy, governance, and society. [1] [2] The international financial regulations such as The Financial Action Task Force are implemented in preventive measures, that is KYC, Know Your Customer, a significant element in the fight against financial crimes. It is a process of identifying and verifying the customer's identity. The use of digital channels has remarkably increased post the COVID-19 pandemic. Due to this, businesses and banks have prioritized the provision of online services to customers. They use biometrics via online channels including video KYC (video identification), to adjust to changing client demands. [3] While proven to be beneficial, it also has a few drawbacks. Malicious invaders may gain access to the system through spoofing attacks and this can lead to leakage of private information. These spoofing attacks are mostly conducted using pictures, flat paper masks with holes, and video sequences with pictures and videos. To combat this risk of spoofing in biometric systems, liveness detection mechanism has been developed. [4] Liveness detection can be active or passive. Active detection usually requires the user to perform some simple tasks such as following a moving object on the screen. As the user performs the task, the detection algorithm analyzes the movement of user. Passive detection however, is seamless to the user. It runs in the background and does not give any indication to the user that they are being tested. Therefore, it is more difficult to spoof. Passive detection may analyze the skin texture or blood flow of a biometric sample. [5] This project focuses on providing a solution to spoofing attacks by implementing the active and passive liveness detection techniques, which differentiates a live person from photographs and videos. The focus will be on passive detection based on the blood flow of a person. This will contribute to enhancing enterprise security, decreasing the financial crime rate, and creating a safer environment for digital transactions.

In the table given below, the different types of liveness detection techniques are described. Liveness detection techniques can be categorized as active and passive. Active techniques require the user to perform specific actions while passive techniques do not require specific actions from the users. The actions can be predicted which makes it easy to spoof. The active aspect makes it inconvenient and non-user friendly. Spoof refers to the type of spoofing that can be done against the liveness detection techniques. Such as, the active techniques eye and face movement can be spoofed by pictures, and videos while the passive techniques can be spoofed using masks made of resin, silicon or wax heads.

	EYE MOVEMENT	FACE MOVEMENT	FACE TEXTURE	BLOODFLOW
TYPE	Active	Active	Passive	Passive
SPOOF	2D	2D	3D	3D

Table 1. Categories of liveness detection techniques.

LITERATURE REVIEW

The liveness detection systems are categorized as intrusive and non-intrusive methods. The techniques that require the clients to perform certain actions are intrusive such as head movement and blinking. On the other hand, non-intrusive methods do not require any client actions such as texture, color, and blood flow analysis. There has been rapid development in face liveness detection methods. The proposed solution was based on the color and texture of the face image. The aim of the solution is to differentiate between real and fake faces using the color and texture. This showed promising results on challenging face spoofing databases however it lacked face detection and pretreatment methods [6].

In order to mitigate the security risks caused by the face recognition systems, numerous face spoofing detection techniques based on video frames and video sequences have been proposed. In video sequence methods, the difference in liveness signals is utilized. They have greater detection accuracy but take more time to process. For instance, a video sequence of 4 seconds is required for a method based on eyeblink detection. In contrast, the video frames methods, the difference in texture and color is used. For the real-time applications, the frame-based techniques are better suited because it is difficult to capture long videos in real time. In this study, it was analyzed that different areas of a live face absorb visible light differently, and standard color cameras can capture changes in the visible light's absorption. Hence, the proposed solution was based on the blood vessel distribution analysis. This method showed promising results on printed images and masks. However, for the replayed video attacks, the result was not satisfactory and the actual blood distribution is different from the one obtained by the attention mechanism [7].

Biometric identification, such as face and iris recognition, has been popular over the past ten years. The liveness detection technique is required in the biometric systems to ensure the accuracy of identity authentication. There are three categories of attacks to facial recognition systems that are static and dynamic. The static or 2D attacks are

done using pictures and video of pictures while the dynamic or 3D attacks are done using masks. This study proposed a face liveness detection technique based on a short video. Firstly, the video is filtered using eulerian magnification then SIFT algorithm is applied to extract the matching pairs. The resultant feature histogram is fed into the SVM classifier. This method showed positive results [8].

A solution that utilizes texture features extracted from different color spaces was proposed to effectively detect and differentiate between genuine and spoofed faces. The challenges faced in face anti-spoofing, include the diverse nature of spoofing attacks such as print attacks, replay attacks, and 3D mask attacks. To address these challenges, a comprehensive framework that combines texture analysis algorithms with multiple color spaces, including RGB, and HSV was proposed. The proposed solution achieves promising results, outperforming existing methods on benchmark datasets such as the Replay-Attack and CASIA FASD datasets. However, there are some limitations in the proposed solution. One limitation lies in the computational complexity of the texture analysis algorithms, which may hinder real-time implementation on resource-constrained devices. Additionally, the authors note that challenging lighting conditions or sophisticated spoofing attacks may still pose difficulties for the proposed solution. The proposed solution offers promising results and insights for future research directions, aiming to enhance the robustness and effectiveness of face anti-spoofing systems. [9].

A technique known as photoplethysmography (PPG) uses light to monitor a number of vital indicators such as pulse rate, respiratory rate and blood oxygenation. Due to the changes in blood volume during the cardiac cycle, PPG can detect differences in the light absorption of human skin. These variations can also be measured at a distance resulting in remote photoplethysmography (RPPG). The dichromatic reflection model states that light reflected by skin has two components, diffuse reflection component and specular reflection component. The diffuse reflection component changes color in response to variations in blood volume while the specular reflection component displays the color of light source and is unaffected by blood volume variations. These components depend on the angle between camera, skin and light

source. To address this challenge, a chrominance-based RPPG method is introduced that utilizes the normalized color difference signals derived from RGB color channels and eliminates the specular reflection component. The algorithm's performance is analyzed by comparing the results with the benchmark algorithms based on blind source separation techniques using independent component analysis and principal component analysis. The chrominance based RPPG algorithm showed promising results [10].

PROBLEM DEFINITION

As the world is moving towards digitizing all the sectors, the financial and enterprise security are the most crucial ones. Digital banks allow customers to create accounts without visiting the branch. Similarly, other institutions offer identical facilities to the customers. This requires KYC verification and authentication. KYC is a process to identify and verify who the customer is and who he claims to be. Initially, biometric data was used. Facial recognition is a convenient method, however, it is also vulnerable to spoofing. To tackle this issue, liveness detection came into light. The technological advances have a negative side to it as well, that is they can be exploited by the fraudsters. The fraudsters use 2D flat masks with holes, 3D prints, wax heads and masks, silicon, and resin masks to conduct facial spoofing attacks. Thus, it is a challenge to imply what is real and what is fake. The liveness detection techniques are categorized as active and passive. In today's world, it is possible to dodge the active liveness detection techniques such as face movements, pupil dilation, blinking. As it requires user's direct involvement, it results in higher abandonment rate and poor user experience. In contrast, the passive liveness detection technique such as texture and blood flow analysis do not require specific user actions, it results in frictionless user experience.

3.1 SOLUTION STATEMENT

The aim of this project is to develop an active and passive liveness detection system using image processing and deep learning techniques through eyeball movement, color-texture and blood flow analysis.

3.1.1 Who Needs It: This system can be used in financial and enterprise security sectors to overcome the challenge of spoofing.

SOFTWARE REQUIREMENTS SPECIFICATIONS

SYSTEM FEATURES

4.1 Face Detection and Feature Extraction

4.1.1 Description and priority:

It is a high priority feature. The system shall detect the face of the user in real-time, capture pictures and then extract features from it.

4.1.2 Stimulus/response sequences:

As the user comes in the frame of the camera, the system should detect the face and capture pictures.

4.1.3 Functional requirements:

REQ-1: The system should detect the face correctly and ignore the background.

REQ-2: The system should alert the user if the system is unable to detect the face due to noisy background.

REQ-3: There should be no object present between the camera and user for successful face detection.

REQ-4: The system should detect faces and capture pictures with minimum time delay.

REQ-5: The system should extract the features efficiently.

REQ-6: The system should notify the user when the input process is done.

4.2 Process the input and generate result

4.2.1 Description and priority:

It is a high priority feature. The system should process the extracted face features and classify it correctly.

4.2.2 Stimulus/response sequences:

As the features are extracted, the system should perform this operation.

4.2.3 Functional Requirements:

REQ-1: The system should classify the face correctly as a real or fake person.

REQ-2: The system should display the result after classification.

REQ-3: The system should perform this task with minimum time delay.

The use case diagram of the liveness detection system involves the interaction between the user and the system. The user, represented as an actor, presents their face to the system to start the procedure. The system begins by performing face detection to identify and locate the face within the input video stream. If a face is recognized correctly, the algorithm moves on to feature extraction, which involves extracting relevant features from the recognized face. These features are then passed through liveness detection algorithms. The algorithms apply various techniques, such as eyeball movement analysis, texture analysis, and blood flow analysis, to determine the authenticity of the presented face. Finally, the system generates the results, indicating whether the presented face is deemed live or not. However, if the system is unable to detect a face in the input, it alerts the user to the lack of a detected face, giving the user the option to adjust or re-present their face for processing and assessment.

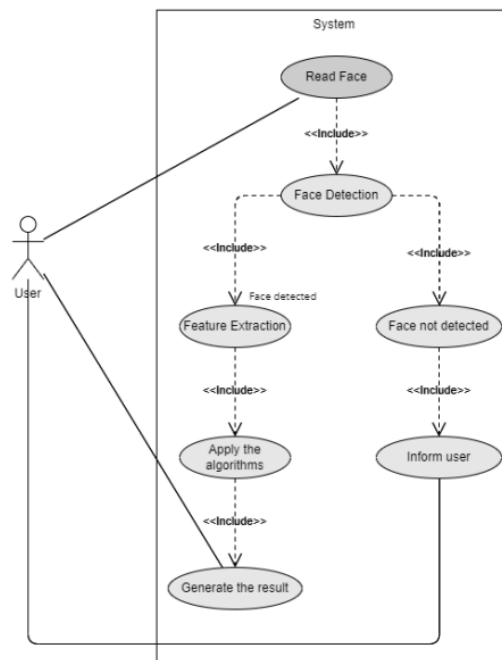


Figure 1: Use case diagram

4.3 OTHER NONFUNCTIONAL REQUIREMENTS:

4.3.1 PERFORMANCE REQUIREMENTS:

1. The system should detect the face and capture pictures within 10 seconds.
2. The system should extract features and process it within 30 seconds.
3. The website should take 1-5 ms to load.

4.3.2 SECURITY REQUIREMENTS:

1. The pictures of the users will not be visible to anyone except the ML models.
2. The pictures of the users will not stay on the cloud in case they have to be saved for processing.

4.3.3 SOFTWARE QUALITY ATTRIBUTES:

- 1. Availability:** The system should be available 24/7.
- 2. Correctness:** The system should perform its intended tasks and generate correct results.
- 3. Reusability:** The system should be reusable, that is it can be used as a preliminary task for further implementation of other innovative technologies.
- 4. Robustness:** The system should be able to handle unexpected inputs, that is frames with no user or frames with noisy background.
- 5. Testability:** The features of the system should be testable and verify whether they meet the project goals.
- 6. Usability:** The system should be easy-to-use for the users. The website should constantly display messages to show progress.

METHODOLOGY

The methodology is explained below;

5.1 Data Collection:

For data collection, users were instructed to upload videos in daylight (outdoor and indoor) for real data and videos displaying printed and screen pictures for fake(negative) data. The videos were then processed to extract frames and detect face in them. Once the face was detected, the frames were cropped around the face region to obtain frames of size 112x112. The cropped images are labelled as 0 for real data and 1 for negative data. The data collection was done using google forms.

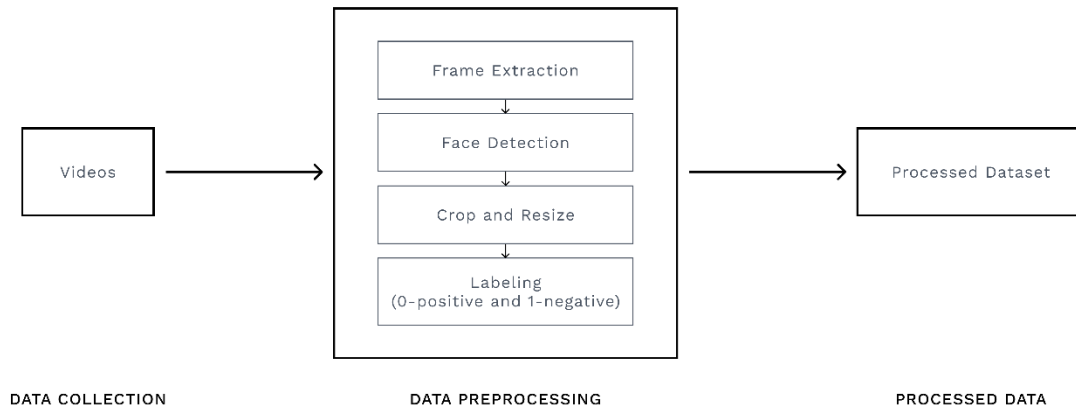


Figure 2. Process of data collection

5.2 Eyeball Movement Analysis:

In the eyeball movement, face is detected and tracked in real-time and then landmarks are detected. After the face and landmark detection, the iris position is determined. A circular shape is displayed on the screen five times at random positions and its' position is recorded. The iris position and position of the shape is compared and output is displayed accordingly.

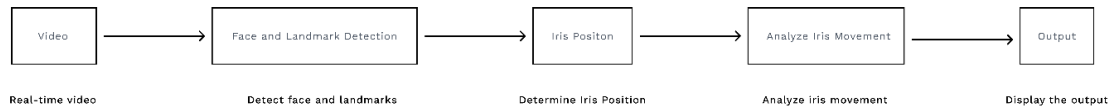


Figure 3. Process of eyeball detection

5.3 Texture Analysis:

For liveness detection using the color-texture analysis, frames are extracted from the video and fed to the algorithm. The color space of the image is converted to HSV and RGB color spaces. After this, the local binary pattern histograms of the images are created and concatenated. The histogram is then fed to the classifier which classifies the images as real or fake.



Figure 4. Process of texture analysis

5.4 Blood flow analysis:

From the real-time video, the face is tracked and detected. The face region is cropped from the frame and mean value of colors in red, green and blue channels of the face is calculated. These mean values are then resampled and processed using the bandpass filter. The chrominance components are determined and amplitude spectrum is extracted. From which, the estimated beats per minute (BPM) and signal-to-noise ratio (SNR) is calculated and processed to determine real or fake.

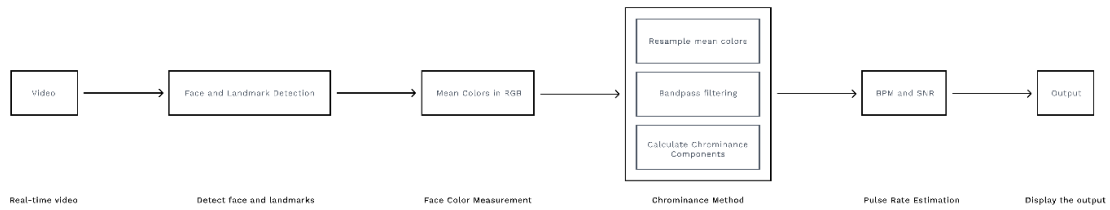


Figure 5. Blood flow Analysis Process

DETAILED DESIGN AND ARCHITECTURE

6.1 SOFTWARE DESIGN MODEL

The waterfall model is a linear and sequential process for software development. For Face liveness Detection System, it would include requirements analysis to define the necessary features and functionalities of the system. The next steps are website design and development to create a user interface of the system. Next, camera integration is implemented then the algorithms are implemented. The algorithms are then integrated in the system and tested to ensure they are functioning correctly. This model ensures that each step is completed before moving on to the next, which helps to minimize the errors and ensure the final product meets the specified requirements.

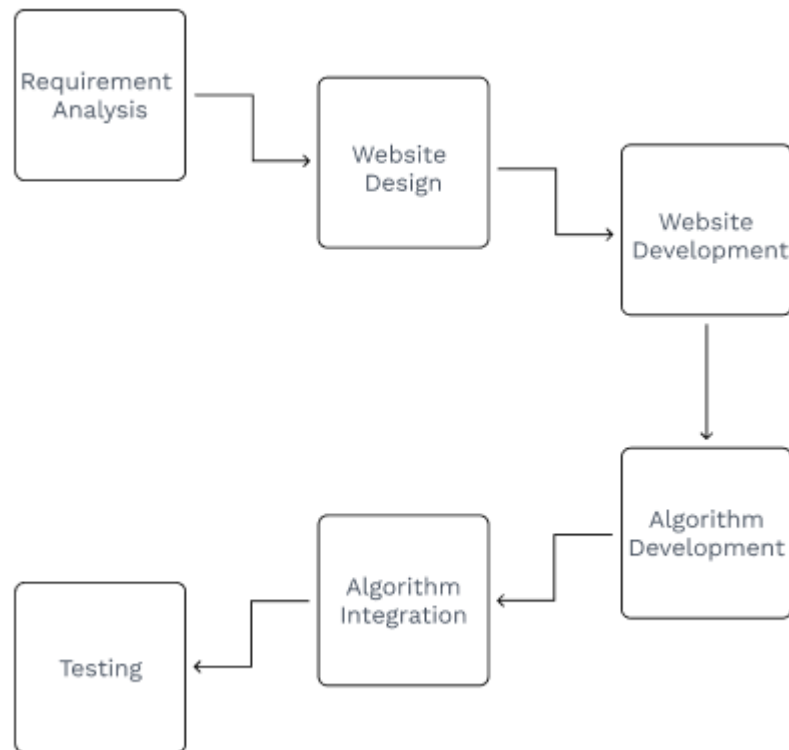


Figure 6. Software Design Model for Face liveness Detection System

6.2 SYSTEM OVERVIEW

Face liveness Detection is a system designed to differentiate between a real and fake person against the spoofing attacks by photographs or masks. The system works by detecting the face and landmarks of the user then recording a video of a few seconds of the user. The video is then processed by extracting frames from it. The extracted frames are then fed to the algorithms. There are two algorithms, texture algorithm and blood flow analysis algorithm.

6.2.1 Architectural Design

The system architecture of the system a high-level outline of the project requirements that introduces the different system components and subsystems. The system is composed of different modules with a specific purpose that interact with each other to provide an efficient liveness detection system. The main components include, user interface module, video processing, feature extraction and liveness detection algorithms module.

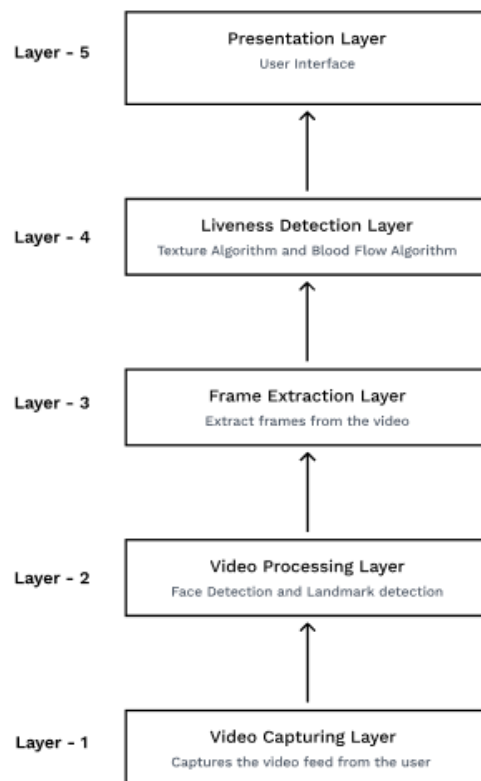


Figure 7. System Architecture

6.3 Design Models

6.3.1 Sequence Diagram

In the sequence diagram from a user's perspective, the sequence starts when the user clicks the liveness button in the UI to initiate the liveness detection process. The UI sends a request to the camera to start capturing live video. The camera begins capturing the live video feed and continuously sends frames to the UI for display. The UI displays the live video feed from the camera to the user. The system checks if a face is detected in the captured video frames. If a face is not captured, the system sends feedback to the user, indicating that no face was detected. This feedback could be displayed on the UI or communicated through a notification. If a face is successfully captured, the system sends the captured face data to the liveness detection model for further processing. The liveness detection model applies the required algorithms and techniques to analyze the captured face and determine its liveness. The model sends the liveness detection result back to the system. The system receives the result and sends it to the UI. The UI displays the liveness detection result to the user, indicating whether the presented face is deemed live or not. The system ends the video feed from the camera. The UI returns to the initial state, waiting for further user interactions. The user views the results displayed on the UI.

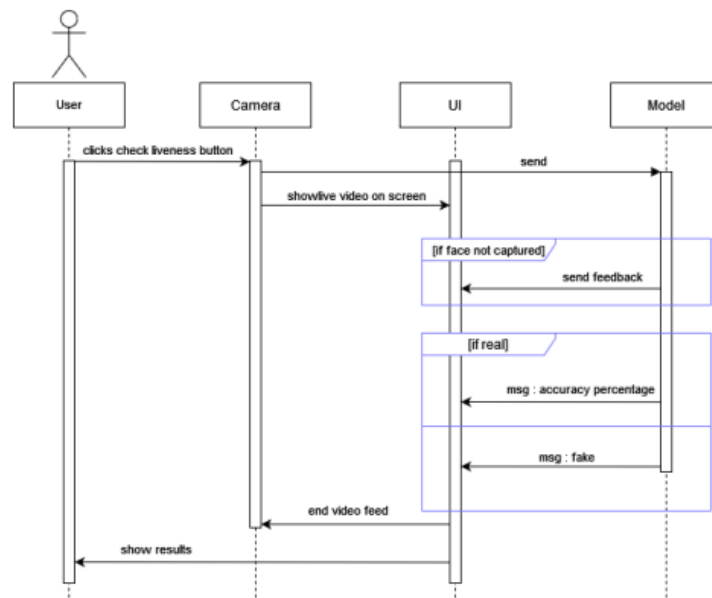


Figure 8. Sequence diagram

6.3.2 Activity diagram

The activity diagram for the liveness detection process begins by showing live video on the screen. The system then moves to the face detection activity, where it locates the faces within the video frames and extract relevant facial features from the detected face. Next, the system checks the number of faces detected. If only one face is detected, the frames are extracted from the video. These frames are then preprocessed to enhance the quality and prepare them for further analysis. The preprocessed frames are fed into the ML model, which applies liveness detection algorithms and techniques. Finally, the result of the liveness detection is obtained, indicating the authenticity of the presented face. This result concludes the activity diagram.

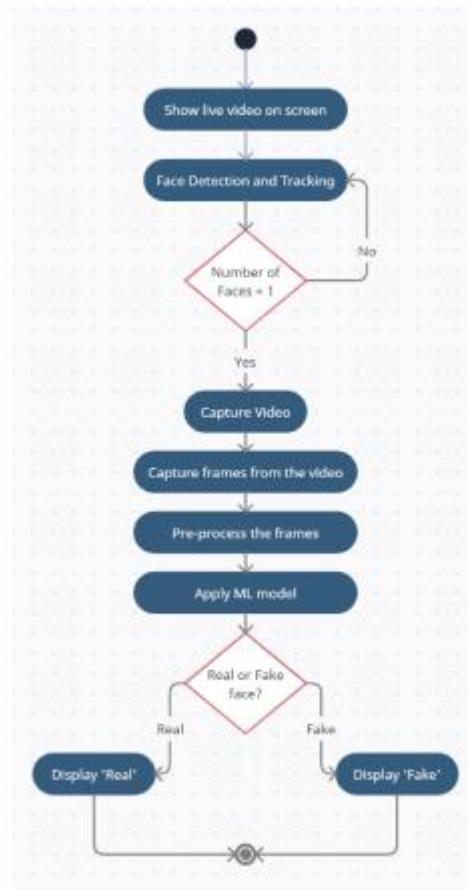


Figure 9. Activity diagram

6.3.3 Data flow diagram

The data flow diagram for the liveness detection system begins with the user accessing the system by pressing a button. This action triggers the flow of data within the system. The system then proceeds to display the live video feed to the user, providing real-time visual information. Simultaneously, the system performs face detection to determine the number of faces present in the video feed. If only one face is detected, the system captures the video and extracts frames from it. These frames are then passed through liveness detection algorithms, eyeball movement analysis, texture analysis, and blood flow analysis. The algorithms analyze the frames and determine the liveness of the captured face. Finally, the system displays the output of the liveness detection process, indicating whether the presented face is considered live or not.

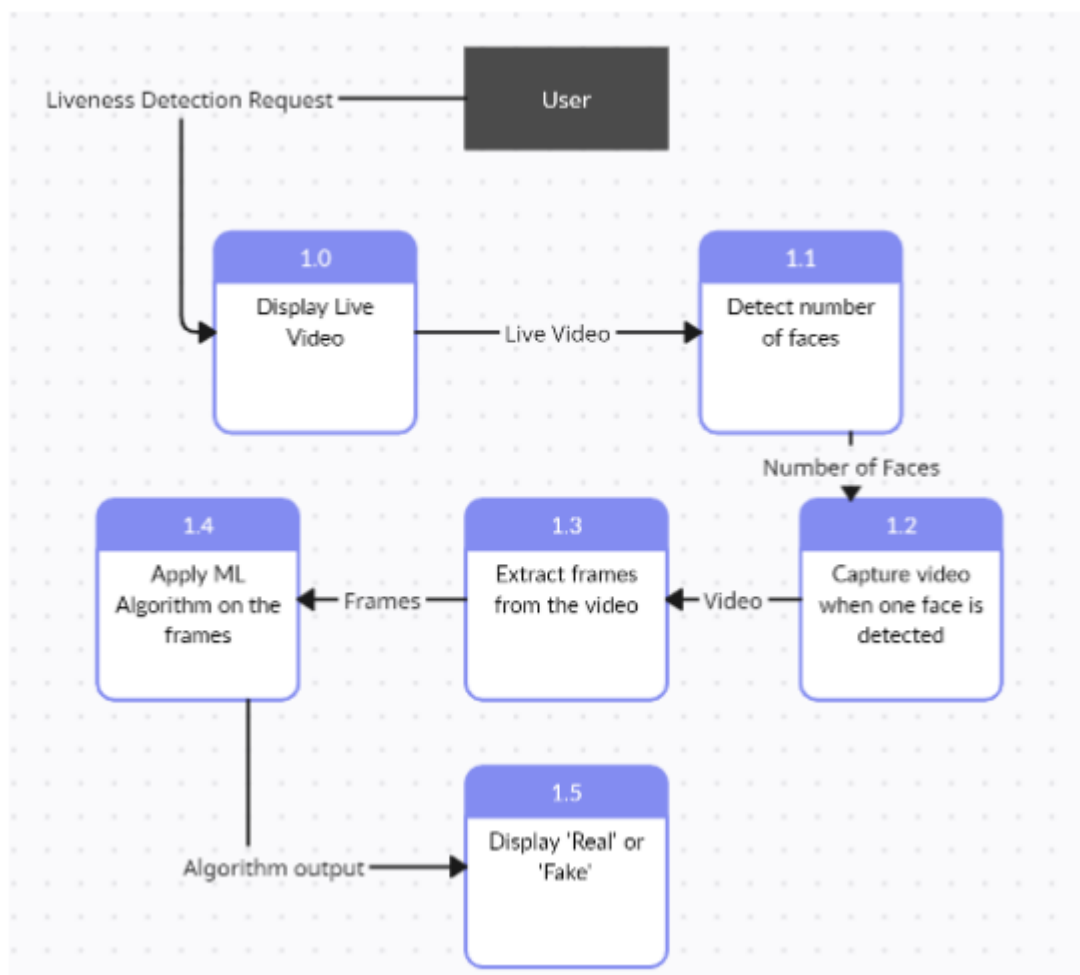


Figure 10. Data flow diagram

6.4 DATA DESIGN

The system uses a video clip of the user as an input to the detection algorithm in order to differentiate between a live and fake person. However, due to privacy and security reasons, it does not store any information about the user. Therefore, a database is not required.

6.4.1 Data Dictionary

6.4.1.1 Video clip: (type: video)

5 seconds video clip of the user will be captured. From this video, several frames will be extracted and used as inputs for the face liveness detection algorithm

6.4.1.2 Detection result: (type: string)

The output of the liveness detection algorithm will be displayed on the screen in the form of a string

6.5 USER INTERFACE DESIGN

6.5.1 Home Page:

The homepage provides the basic information about the system and its features. It has a navbar that allows the user to access different sections of the website such as services, use case, demo and about etc. It has a call-to-action button as well to directly use the key feature of the system. Moreover, it highlights the benefits and applications of the system.



Figure 11. Home Page

6.5.2 Demonstration Page:

This page allows the user to utilize the system's functionality of liveness detection. The camera will detect the face and 68-landmark points of the user, and record a video. This video will then be preprocessed by frame extraction then it will be processed by the texture and blood flow algorithms. The output will be displayed to the user on the screen.

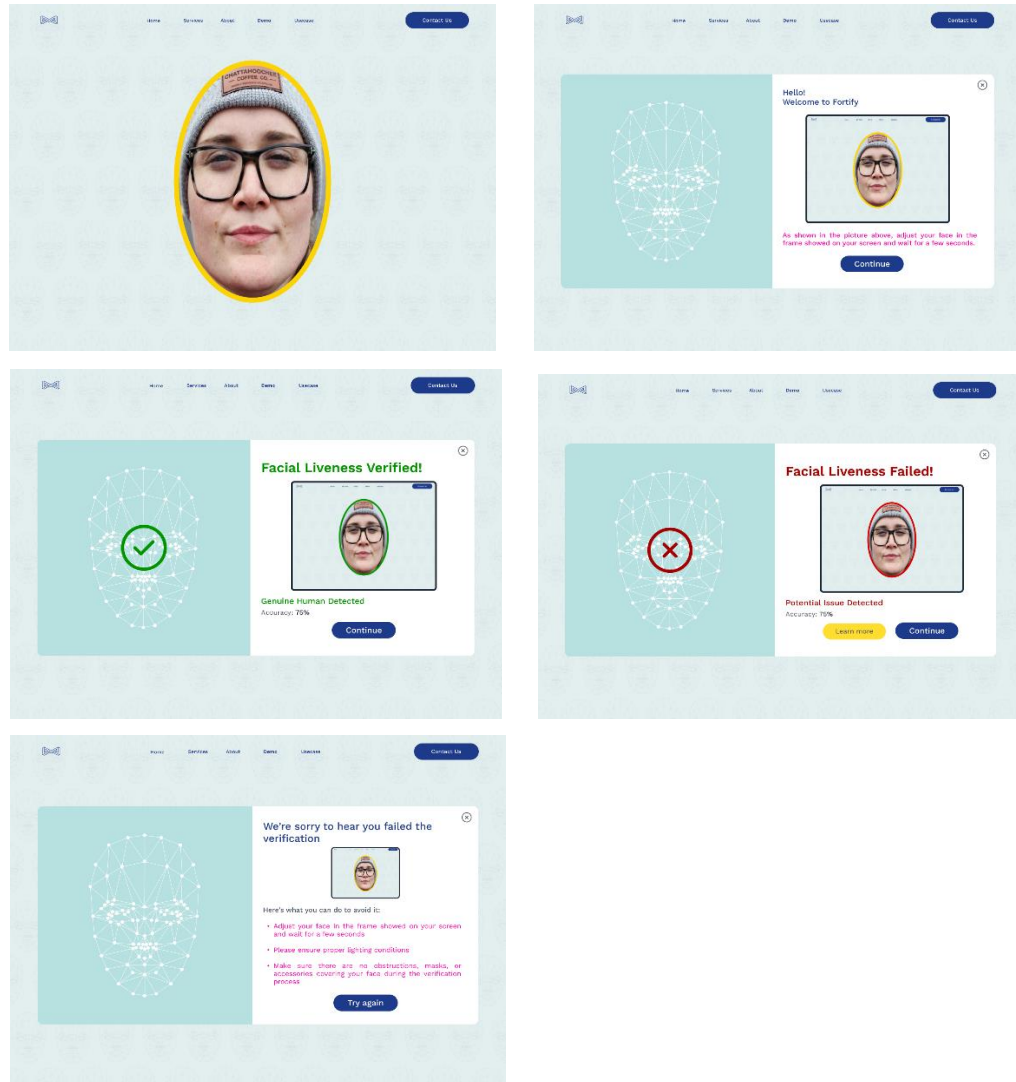


Figure 12. Demonstration Page User Interface

6.5.3 Services Page:

The services page highlights the features of the system and provides a step-by-step description of the process. It is designed to provide the users with a comprehensive understanding of the system's services and how they can be used to meet their specific needs.

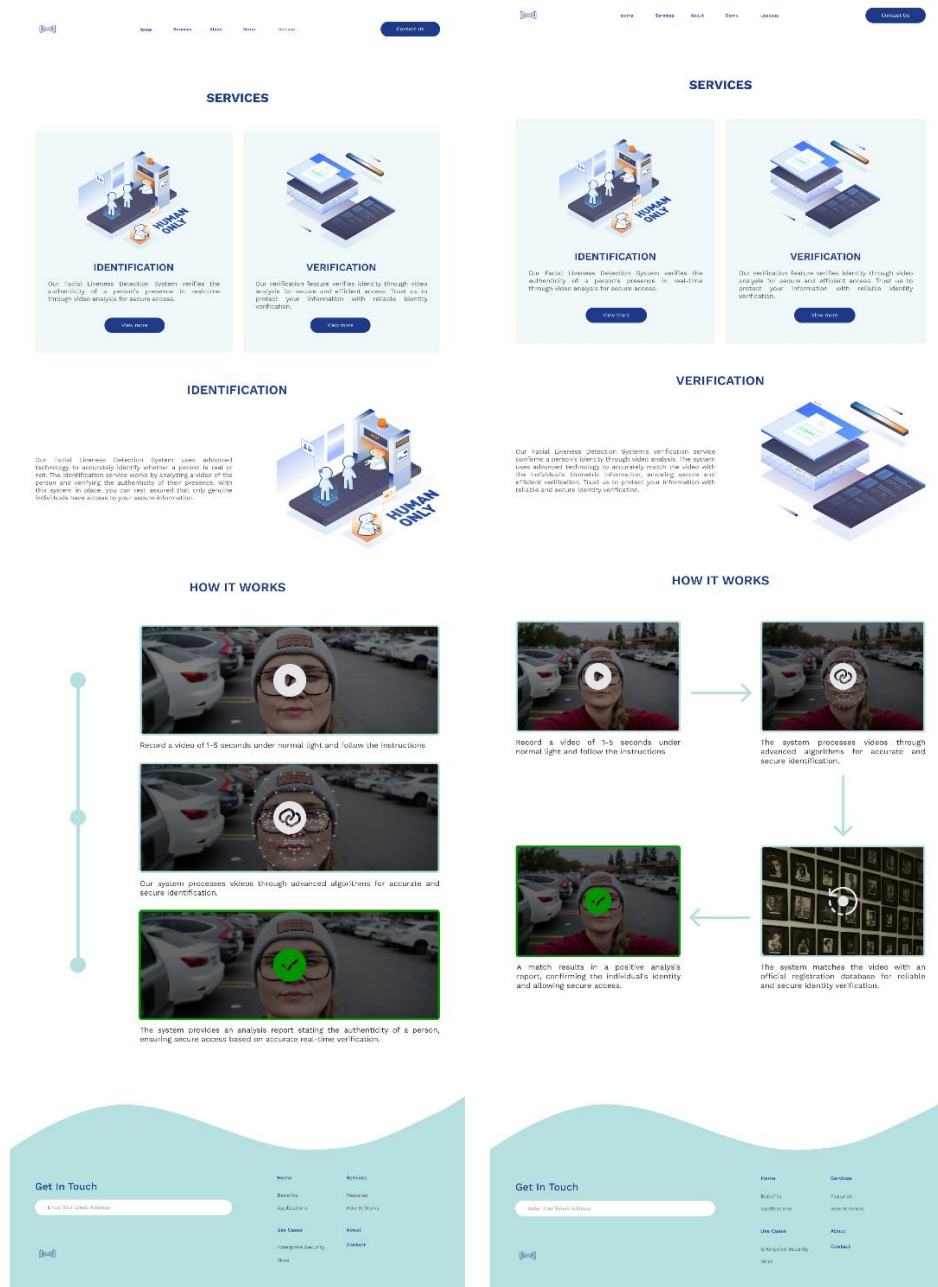


Figure 13. Services pages

6.5.4 Use case Page:

This page includes a video depicting the shortcomings of the available facial authentication systems and how face liveness detection systems can play its role in providing a robust and efficient solution.

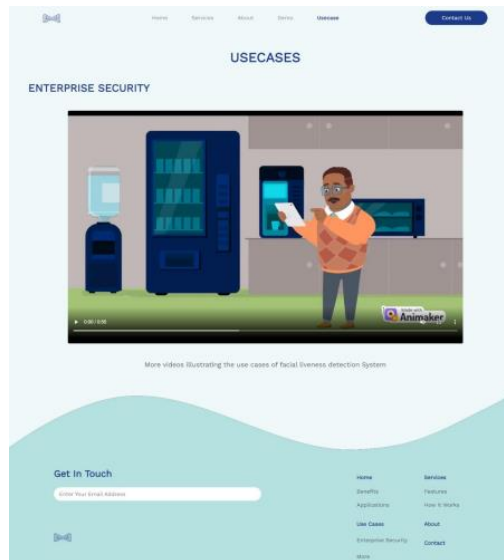


Figure 14. Use Case Page

6.5.5 About Page:

The about page provides a brief description of the system and the team. It includes an overview of the system's needs, capabilities and benefits.

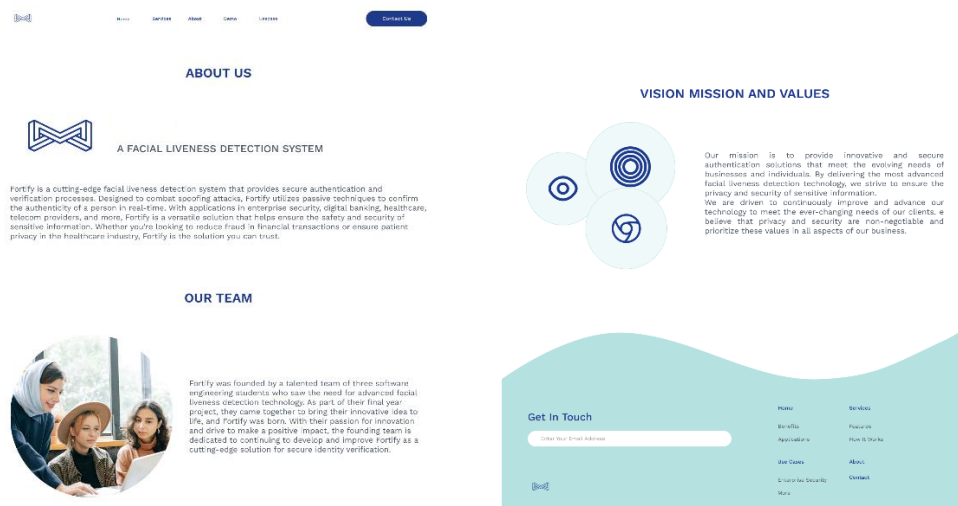


Figure 15. About Page

6.5.6 Contact Page:

The contact us page of a face liveness detection system provides users with a way to get in touch with the team.

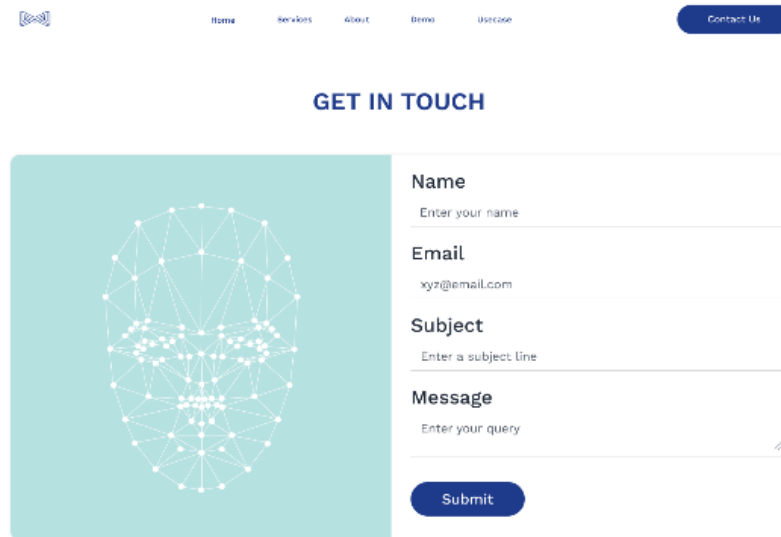
The image shows a web page layout for a contact form. At the top, there is a navigation bar with a logo on the left and links for 'Home', 'Services', 'About', 'Demo', 'Usecase', and a 'Contact Us' button on the right. Below the navigation bar, the heading 'GET IN TOUCH' is centered. The main content area is divided into two sections. The left section features a teal square with a white geometric pattern of dots and lines. The right section is a white form with fields for 'Name' (placeholder: 'Enter your name'), 'Email' (placeholder: 'xyz@email.com'), 'Subject' (placeholder: 'Enter a subject line'), and 'Message' (placeholder: 'Enter your query'). A blue 'Submit' button is located at the bottom of the form.

Figure 16. Contact Page

Figma Link:

<https://www.figma.com/file/OldLwWKHJbD6fx19Esqumy/FYP?nodeid=0%3A1&t=NexhH4ea2QRg0Zm5-1>

6.5.7 Screen Objects and Actions

A discussion of screen objects and actions associated with those objects

6.5.7.1 Navbar:

The navbar allows the user to navigate between the pages of the website including home, services, about, demo and use case pages. It also has a link to the contact us page.

6.5.7.2 Logo:

The logo on the top of the page allows the user to return to the homepage of the website.

6.5.7.3 Footer buttons:

The buttons on the buttons allow navigation between different sections of a webpage.

6.5.7.4 Try now:

The 'try now' button takes the user to the demonstration page, which allows him or her to test the liveness detection functionality.

6.5.7.5 Continue Demo:

The continue button on the demonstration page will allow the website to start capturing a live video of the user by the webcam or a connected camera. This will be used to detect and track the face of the user and provide input to the liveness detection algorithm

6.5.7.6 Services buttons:

The services buttons, i.e., identification (view more) and verification (view more), will allow the user to view the identification services or the verification services of the system.

6.5.7.7 Video play button:

Clicking on the use case video play button, will allow the user to start a video that shows the need of this system through a simple use case scenario.

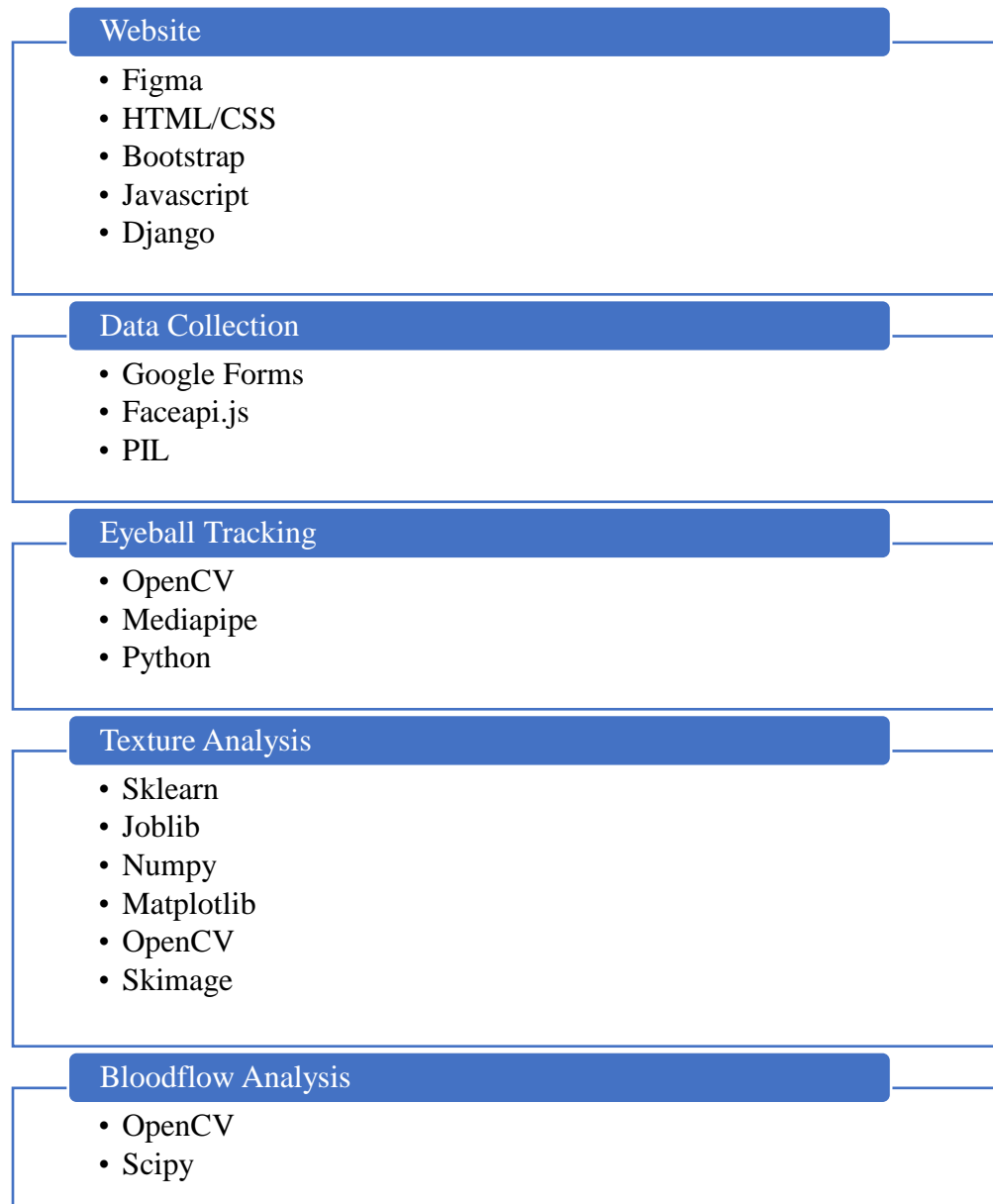
6.5.7.8 Contact form:

The contact form will allow the users to send a message to the owners of the website. The owners might contact the user to address the query through the provided email address of the user

IMPLEMENTATION AND TESTING

7.1 IMPLEMENTATION

The technologies used for the implementation of the system are shown in the figure below;



7.2 TESTING

The testing techniques include the following;

1. Real-time Testing:

This involved testing the system in real-time scenarios using a variety of inputs. It included verifying the accuracy of the algorithms individually by assessing their performance on real and fake samples, such as printed pictures, videos, and digital images. This helped to evaluate the system's ability to distinguish between real and fake inputs accurately.

2. Test Dataset Evaluation:

For the texture algorithm, testing was performed using the test dataset that has been collected. This dataset consisted of real and fake samples including printed pictures and digital images. This allowed for the comprehensive testing of the performance of algorithm and assessing its accuracy.

3. User Feedback Testing:

After providing liveness detection result to the user, an additional testing technique involves obtaining feedback from the user. Users are asked to confirm whether the output of the system matches the reality of the input they provided. This feedback helps evaluate the effectiveness of the system in accurately identifying real and fake inputs.

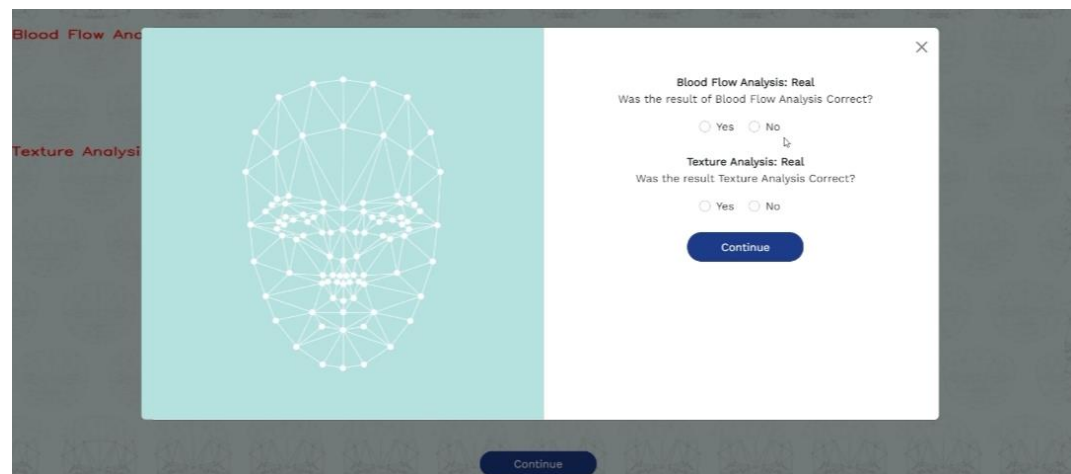


Figure 18. User Feedback

RESULTS AND DISCUSSION

The accuracy of the **eyeball movement technique** in liveness detection mainly depends on user interaction and cooperation. The user's ability to accurately track the required eye movements is a key factor in this technique's efficiency. This technique yields satisfactory results in individuals without significant eye-related conditions. The certain eye-related conditions such as nystagmus (involuntary eye movements), strabismus (misalignment of the eyes) and prosthetic or artificial eyes, might make it difficult for a person to perform the necessary eye movements, which could result in inaccurate liveness detection. Moreover, the environmental factors such as poor lighting conditions, may impact the visibility of eye and result in inaccurate results.

USER GROUP	APPLICABILITY OF EYE MOVEMENT TECHNIQUE
Normal Individuals	Yes
Individuals with Spectacles	Yes
Individuals with Nystagmus	No
Individuals with Strabismus	No
Individuals with Prosthetic Eye	No

Table 2. Eyeball Movement Applicability

The **texture algorithm** was initially trained on the NUAA dataset, that consists of real face images and spoofing face images including printed pictures, and, tested on the NUAA dataset and collected dataset. The color spaces used are YCrCb, HSV and RGB. Initially, the **SVM classifier**, a supervised machine learning classifier was used. SVM classifier achieved a training accuracy of 66.3894%, percentage error of 1% and

percentage half total error rate of 0.5% on NUAA dataset across three different color spaces. The test accuracy on the NUAA dataset was 66.3614% and HTER of 50%.

SUPPORT VECTOR MACHINE						
Colorspace	Train (NUAA)			Test (NUAA)		
	Acc	EER	HTER	Acc	EER	HTER
YCbCr, HSV, RGB	66.3894	100	50	66.3614	0	50

Table 3. Texture algorithm – SVM classifier

As the train and test accuracy in three color spaces was same, and half total error rate was 50% in the SVM classifier, other classifiers were considered.

K-Nearest Neighbor classifier is another simple and effect supervised learning algorithm. The training accuracy in three color spaces was 1% and percentage error was 0%. In **YCrCb** color space, this classifier gave the test accuracy and percentage error of 99.8554% and 28.65% on NUAA dataset, while on the collected dataset, it was 55.7364% and 44.3706% respectively. Using **HSV** and **RGB** color spaces, the test accuracy and percentage error were approximately equal. However, on the collected dataset, the test accuracy using HSV color space was 66.0886% and 62.2199% using the RGB color space.

K NEAREST NEIGHBOURS									
Colorspace	Train (NUAA)			Test (NUAA)			Test on Unknown		
	ACC	EER	HTER	ACC	EER	HTER	ACC	EER	HTER
YCbCr	100.00	0.00	0.00	99.86	0.29	0.18	55.74	44.37	44.56
HSV	100.00	0.00	0.00	99.90	0.14	0.14	66.09	35.18	36.21
RGB	100.00	0.00	0.00	99.81	0.14	0.25	62.22	37.80	39.18

Table 4. Texture algorithm – KNN Classifier

Another classifier used was the **Decision Tree Classifier**. When applied on the NUAA dataset, it achieved the training accuracy of 1% across the three different color spaces. On the NUAA dataset, this classifier gave the test accuracy of 98.1205% with the percentage error of 24.355% in YCrCb color space. On the other hand, considering the collected dataset, the classifier achieved the highest test accuracy of 61.6329% in RGB color space with the percentage error of 34.7279%.

DECISION TREE CLASSIFIER									
Colorspace	Train (NUAA)			Test (NUAA)			Test on Unknown		
	ACC	EER	HTER	ACC	EER	HTER	ACC	EER	HTER
YCbCr	100.00	0.00	0.00	98.12	2.44	2.02	53.18	51.21	44.30
HSV	100.00	0.00	0.00	96.96	4.01	3.28	56.06	52.81	38.84
RGB	100.00	0.00	0.00	97.59	4.15	2.84	61.63	34.79	40.42

Table 5. Texture Algorithm – Decision Tree Classifier

Another popular classifier is **Random Forest Classifier** that achieved the training accuracy of 1% with the percentage error of 0% in all three different color spaces. On testing it on NUAA dataset, this classifier gave the same test accuracy in three color spaces, however the percentage errors were different that is 14.33% in YCrCb, 28.65% in HSV and 57.31% in RGB. Comparing the performance on collected dataset, the highest test accuracy and lowest percentage error was 76.6275% and 30.3753% respectively in **HSV** color space followed by 63.9808% test accuracy and 38.6239% percentage error in RGB color space.

RANDOM FOREST CLASSIFIER									
Colorspace	Train (NUAA)			Test (NUAA)			Test on Unknown		
	ACC	EER	HTER	ACC	EER	HTER	ACC	EER	HTER
YCbCr	100.00	0.00	0.00	99.57	0.14	0.64	58.11	41.44	41.36
HSV	100.00	0.00	0.00	99.57	0.29	0.61	76.63	30.38	28.25
RGB	100.00	0.00	0.00	99.57	0.57	0.64	63.98	38.62	40.12

Table 6. Texture algorithm – Random Forest Classifier

Based on the analysis using three color spaces and different classifiers, it was found that better results were achieved using the Random Forest Classifier.

Next, how different colorspace performed on the random forest classified model was determined. For this, we used the collected dataset to train the model and YCrCb, RGB and HSV color space was used. We checked the impact of each component of each color space individually to see the impact of each component. Then, the combinations of two the color species to see which give us the highest accuracy and the lowest false positive rates and false negative rate. False negative rates or more important in our case because it is more important to detect all images and insure that not of them are classified as real images. If any real images are classified as spam images it may decrease the user experience but will be less costly in terms of security. (All gave 100% accuracy on train dataset so it is not mentioned.)

RANDOM FOREST CLASSIFIER									
Colorspace	Test (Collected)			NUAA			LCC FASD		
	ACC	FPR	FNR	ACC	FPR	FNR	ACC	FPR	FNR
YCrCb	99.82	0.37	0.09	63.36	32.96	38.51	67.37	28.76	33.36
Y	99.15	1.39	0.60	65.58	57.34	22.81	69.12	47.72	27.72
C _r	98.50	2.88	0.86	41.17	19.59	78.70	65.04	49.65	32.20
C _b	98.68	1.95	1.03	41.73	33.36	70.89	48.21	30.40	55.80
RGB	99.35	1.39	0.30	62.81	67.36	21.92	76.37	53.68	17.98
R	98.71	2.79	0.60	55.27	72.38	30.72	75.51	59.28	17.95
G	98.68	2.41	0.82	67.93	66.32	14.73	74.03	58.93	19.78
R-G	93.26	13.56	3.58	63.57	57.83	25.60	74.67	49.82	20.73
B	98.26	4.46	0.47	63.38	73.44	17.98	76.87	60.85	16.05
HSV	99.94	0.09	0.04	67.41	96.70	0.13	76.93	35.30	20.78
H	96.94	6.22	1.59	67.07	91.05	3.50	69.93	56.65	25.07
S	97.62	5.11	1.12	67.61	88.50	3.98	76.12	59.45	17.19
V	98.91	2.51	0.43	51.90	75.33	34.31	76.92	58.52	16.43
HSV+RGB	99.74	0.46	0.17	68.35	88.04	3.09	77.02	39.61	19.86
HSV+YCrCb	99.65	0.46	0.30	70.74	61.62	12.87	72.09	28.53	27.79
RGB+YCrCb	100.00	0.00	0.00	99.71	0.46	0.22	60.23	51.26	33.95

Table 8. Texture algorithm – Finalized Model

HSV+YCrCb were chosen as they demonstrated the best results among all datasets. Lastly, we aimed to optimize the RandomForest classifier's hyperparameters using grid search. First, we loaded the feature and label data from files. We defined a parameter grid to explore different values for 'n_estimators', 'criterion', 'max_depth', 'min_samples_split', and 'min_samples_leaf'. We generated all possible combinations of these parameters. To expedite the search process, we selected half of the combinations for grid search. Using a fixed random state, we instantiated the random

forest classifier. We performed grid search with the selected parameter combinations and employed 5-fold cross-validation. The search yielded the best hyperparameters, determining the minimum samples required for a split as 10, while it gave default values for the other parameters. Finally, we evaluated the best model obtained from grid search on an independent test set and reported its accuracy.

The final model was trained on collected dataset, and used HSV and YCrCb colourspaces with ‘min_samples_split’ as 10.

When tested real-time and collected dataset, the **blood flow analysis algorithm** achieved the accuracy of 97.50% for real people in daylight, 96.66% for real people in indoor lighting, 73.33% for digital spoof images and 70.37% for fake print images.

BLOOD FLOW ANALYSIS	ACCURACY
Real – daylight	97.50%
Real – indoor	95.66%
Fake – Digital Screen	78.33%
Fake – Print	75.37%

Table 9. Blood Flow Algorithm

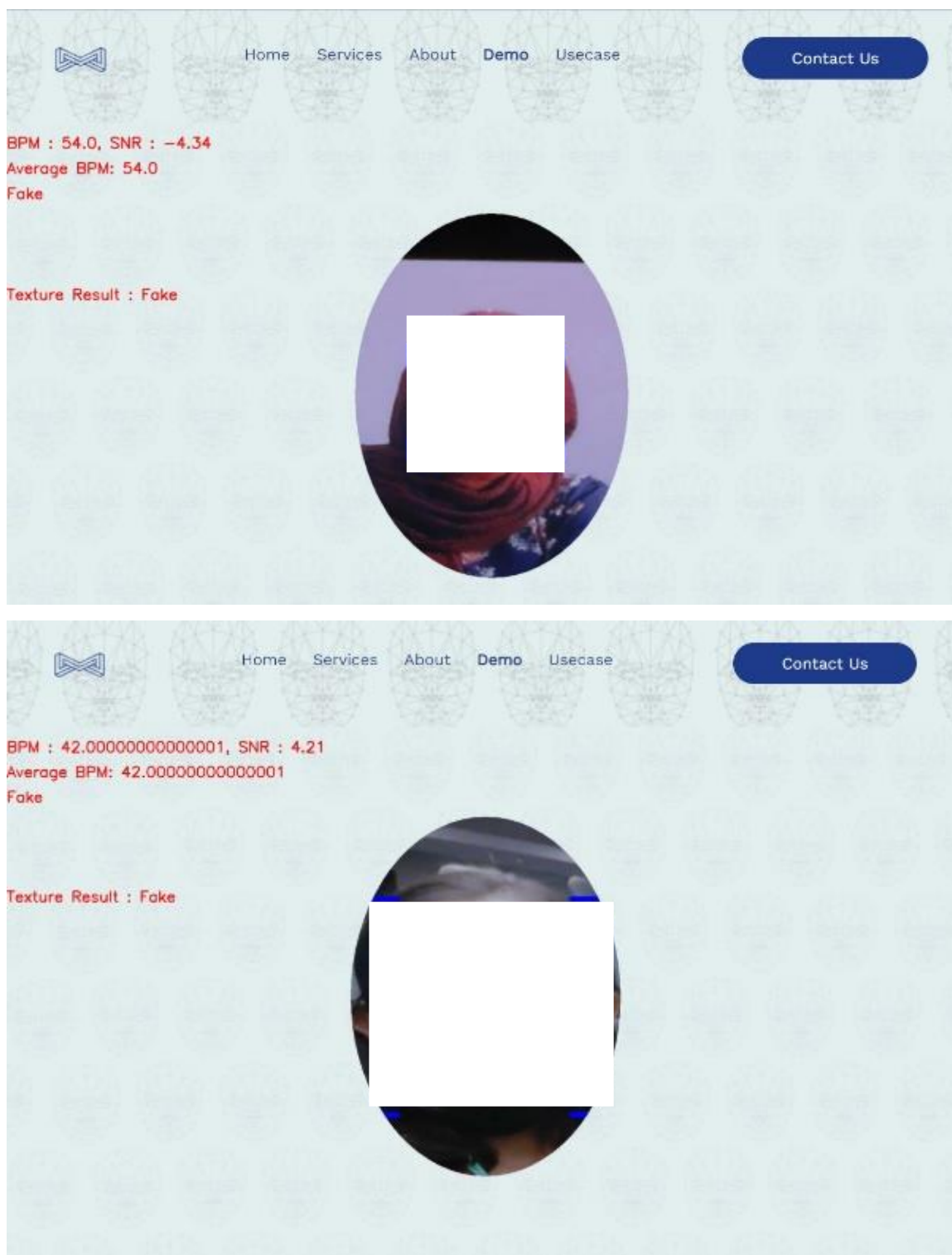


Figure 19. Liveness Algorithms Results

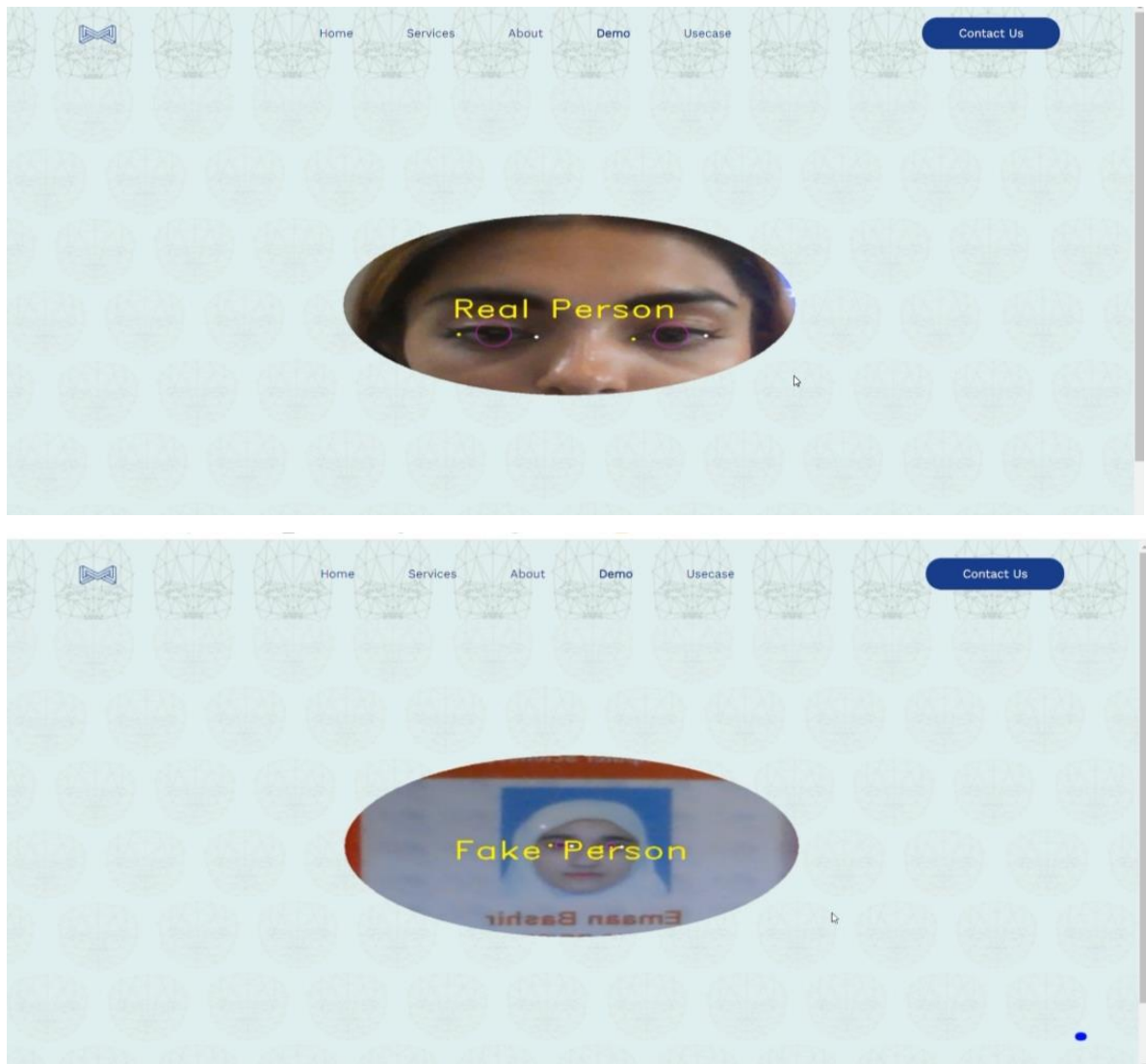


Figure 20. Eyeball Detection Results

CONCLUSION AND FUTURE WORK

In conclusion, the goal of this project was to develop a face liveness detection system incorporating three algorithms that is eyeball, texture analysis and blood flow analysis. By implementing the active and passive liveness detection techniques, the system's ability to detect spoofing attacks was enhanced. The system has demonstrated promising results in detecting spoofing attacks including printed pictures, digital screen images and videos. However, it is crucial to note that the system has some limitations related to lighting conditions. For future, work can be done in several areas. The dataset can be expanded to incorporate a more diverse and extensive data to improve the reliability of the system. Different types of spoofing techniques can be considered such as 3D masks. Moreover, considering the regional demographical characteristics also play an important role in dataset. The other advanced machine learning and deep learning techniques such as neural networks can be considered for face liveness detection.

REFERENCES

1. *What is Financial Crime? | Dow Jones Professional*. (n.d.). Retrieved October 6, 2022, from <https://www.dowjones.com/professional/risk/glossary/financial-crime/>
2. *The History of Financial Crime: Exploring Examples of Modern Money Crimes - Enterprise Risk Management Software | BusinessForensics*. (n.d.). Retrieved October 6, 2022, from <https://businessforensics.nl/financial-crime-history/>
3. *What is KYC in Banking? (Updated)*. (n.d.). Retrieved October 6, 2022, from <https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/issuance/id-verification/know-your-customer>
4. *Liveness detection - face & fingerprint (anti-spoofing) | Thales*. (n.d.). Retrieved October 6, 2022, from <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/liveness-detection>
5. *Liveness Detection - Definition, FAQs - Innovatrics*. (n.d.). Retrieved October 6, 2022, from <https://www.innovatrics.com/glossary/liveness-detection/>
6. Sichuan Institute of electronics, & Institute of Electrical and Electronics Engineers. (n.d.). *ICCCBDA 2019: 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analytics : April 12-15, 2019, Chengdu, China*.
7. Li, L., Yao, Z., Zhou, S., Ma, Y., Wu, J., & Xia, Z. (2022). *Image Analysis of Facial Blood Vessels for Anti-Spoofing of Printed Image and 3D Mask Attacks*. 68–72. <https://doi.org/10.1109/icipmc55686.2022.00021>
8. Kollreider, K., Fronthaler, H., & Bigun, J. (2005). Evaluating liveness by face images and the structure tensor. *Proceedings - Fourth IEEE Workshop on Automatic Identification Advanced Technologies, AUTO ID 2005, 2005*, 75–80. <https://doi.org/10.1109/AUTOID.2005.20>
9. Boulkenafet, Z., Komulainen, J., & Hadid, A. (2015). *face anti-spoofing based on color texture analysis*. <http://arxiv.org/abs/1511.06316>

10. De Haan, G., & Jeanne, V. (2013). Robust pulse rate from chrominance-based rPPG. *IEEE Transactions on Biomedical Engineering*, 60(10), 2878–2886. <https://doi.org/10.1109/TBME.2013.2266196>