National University of Sciences & Technology (NUST) School of Electrical Engineering and Computer Science (SEECS)
Department of Computing

Final Year Project

Software Requirements Specification

For

Fortify – Face Liveness Detection for Spoofing

Version 1.0

Arooba Siddiqi
Emaan Bashir
Rimsha Mirza

Advisor: Dr.Seemab Latif

December 1st, 2022

Table of Contents

Table of Contents 2
Revision History 2
1. Introduction 3
1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Product Scope 3
1.5 References 3
2. Overall Description 4
2.1 Product Perspective 4
2.2 Product Functions 4
2.3 User Classes and Characteristics 4
2.4 Operating Environment 4
2.5 Design and Implementation Constraints 5
2.6 User Documentation 5
2.7 Assumptions and Dependencies 5
2.8 External Interface Requirements 5
3.1 User Interfaces 5
3.2 Hardware Interfaces 7
3.3 Software Interfaces 7
3.4 Communications Interfaces 8
4 System Features 8
4.1 System Feature 1 8
4.2 System Feature 2 (and so on) 9
5. Other Nonfunctional Requirements 10
5.1 Performance Requirements 10
5.2 Safety Requirements 10
5.3 Security Requirements 10
5.4 Software Quality Attributes 10
5.5 Business Rules 10
6. Other Requirements 11
Appendix A: Glossary 11
Appendix B: To Be Determined List 11

1. Introduction

1.1 Purpose

This project aims to detect face liveness and differentiate between a live person and a spoofing attack using a paper mask, photograph, or video sequence. This will help reduce the risk of financial and enterprise security breach and will provide the users with a safer access to confidential information. The goal is to improve user authentication in security systems and create a secure environment for organizations with critical data.

1.2 Document Conventions

TBD

1.3 Intended Audience and Reading Suggestions

This document is meant for developers, testers, and advisors to understand exactly what the system intends to do.

1.4 Product Scope

This project is meant to be used in security systems to prevent unauthorized access. Organizations can use this software to improve their facial recognition systems by detecting spoofing attacks (using paper masks, photographs, or video sequences) and allowing the systems to be accessed only by certain authorized people.

1.5 References

NA

2. Overall Description

2.1 Product Perspective

This project takes inspiration from already existing face liveness detection systems, which detect liveness through techniques such as texture analysis and color distortion analysis. These systems can still be spoofed by creating masks having texture like human skin. We aim to detect liveness using blood flow analysis which is more difficult to spoof than the previously existing techniques. Thus, our system will increase the security of existing facial recognition systems.

2.2 Product Functions

The system shall perform the following functions:

- Capture picture of a person
- Locate the face in a visual scene (Face Detection)
- Process the image and differentiate between a fake and live person (Liveness Detection)
- Combine responses of different models
- Display result of liveness detection

2.3 User Classes and Characteristics

The following two kinds of users will be using the system:

1. Security in charge of organizations with facial security systems
2. Employees of such organizations

The security in charge can use the system to improve the security of facial recognition systems. It can be used to allow only the authenticated employees to access certain systems.

2.4 Operating Environment

The model will work on mobile devices, tablets, and laptops, through the website.

2.5 Design and Implementation Constraints

As the system uses Machine Learning model at the backend, it requires considerable processing power and memory but that shall be handled using cloud integration and GPUs.

2.6 User Documentation

The user interface is just to present the model, and is extremely easy to understand. Hence user documentation is deemed not necessary.

2.7 Assumptions and Dependencies

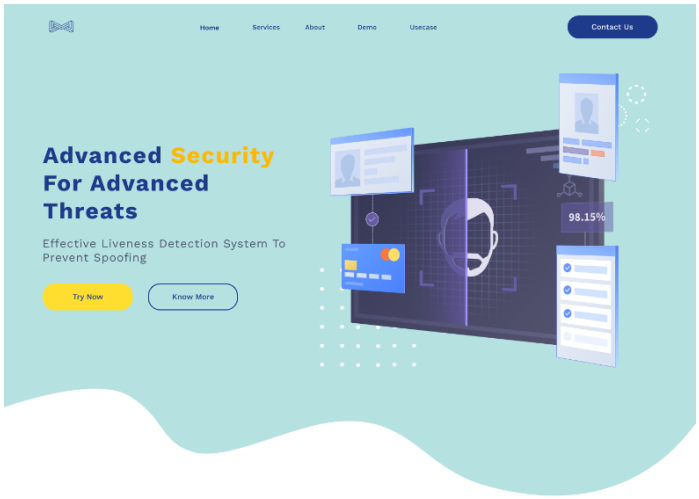
The system depends on face-api.js for face detection and tracking, it uses the ssdmobilenetv1 model. The feature extraction step uses the faceLandmark68Net model.

Different Python libraries are also used for pre and post processing (section 3.3).

3. External Interface Requirements

3.1 User Interfaces

The website will include: the landing page, camera page, the usecase page, and about page. The landing page will include some basic information about the services regarding the benefits of the product. The camera page will be a live testing page to try out the model. The usercase page will include a video explaining the different usecases of the product and the about page will include basic details about product, team and methodologies used.



BENEFITS

APPLICATIONS

Enterprise Security

Text explaining how our facial liveness detection system works in enterprise security. Please add something meaningful here. Add something add something add something here. Think. Facial liveness detection system against spoofing

Learn more

Digital Banking

Text explaining how our facial liveness detection system works in digital banking. Please add something meaningful here. Add something add something add something here. Think. Facial liveness detection system against spoofing

Learn more

3.2 Hardware Interfaces

The web app shall be able to run on the browsers mentioned in section 2.4 with similar efficiency whether they are running on PCs, laptops, tablets, or mobile devices.

3.3 Software Interfaces

The Web Development Framework used will be Django.

The system will communicate with a pre-built model (for color-texture analysis) and a custom-built model (for blood-flow analysis). These will then be combined intelligently (combining method TBD) to provide most accurate results.

The following additional Python libraries will also be used for feature detection, feature extraction, data collection and formatting, model building, model training :

- pytorch

- cv2
- tensorflow
- Mtcnn
- Skimage
- Numpy
- Sklearn
- Metric
- Matplotlib
- math
- Glob
- Sys
- Os
- Torch
- Torchvision
- Pil
- Data
- Nets
- statistics

3.4 Communications Interfaces

NA

4. 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:

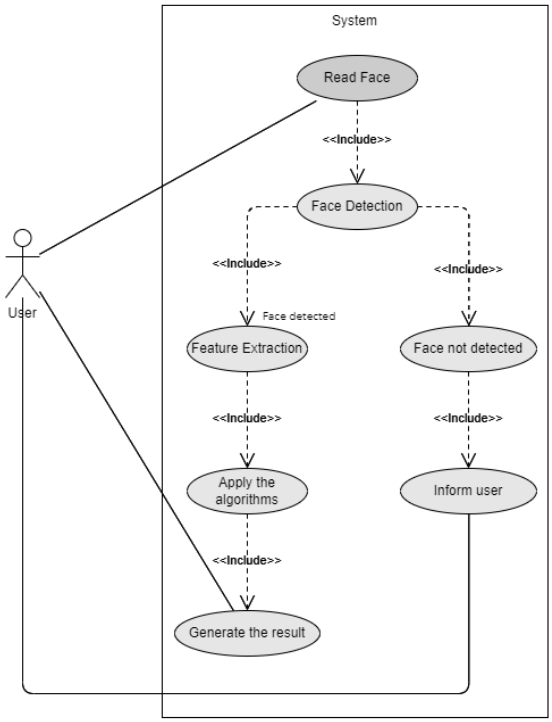
As the features are extracted, the system should perform this operation.

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.



5. OTHER NONFUNCTIONAL REQUIREMENTS:

5.1 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.

5.2 SAFETY REQUIREMENTS:

NA

5.3 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.

5.4 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.

6. BUSINESS RULES:

TBD

7. OTHER REQUIREMENTS:

The system's accuracy should be constantly improved until it reaches an accuracy of >80%.

The user interface should be improved based on the user feedback.

APPENDIX A: GLOSSARY

NA

APPENDIX B: TO BE DETERMINED LIST

The document conventions and business rules are to be determined.