BSCS FINAL PROJECT

Lie Detector using facial analysis and physiological signals

Advisor: **Muhammad Basit Ali Gilani**

Presented by:

Group ID: F23CS094

Student Reg#	Student Name
L1F20BSCS0114	**ABDUL QADEER**
L1F20BSCS0660	**AREEBHASSAN**
L1F20BSCS0988	**MOHAMMAD ALI**

Faculty of Information Technology

University of Central Punjab

Deception Detection Using Facial Analysis and Physiological Signals

By

Abdul Qadeer
Mohammad Ali
Areeb Hassan

Project submitted to

Faculty of Information Technology,

University of Central Punjab,

Lahore, Pakistan.

in partial fulfillment of the requirements for the degree of

MASTER/ BACHELOR OF SCIENCE
IN
COMPUTER SCIENCE

Muhammad Basit Ali Gilani

______________________________	______________________________
Project Advisor	Manager Projects

Abstract

The detection of deception traditionally relies on polygraph tests, which are intrusive and require specialized equipment and trained personnel. This project addresses the challenge of developing a non-invasive deception detection system that utilizes video input to analyze facial expressions and physiological signals. The aim is to create a more accessible and scalable solution that can be used in various applications, such as security screenings and forensic investigations. Our approach leverages computer vision techniques and machine learning models to interpret subtle cues like facial micro-expressions and changes in heart rate, which are indicative of deceptive behavior. Our result demonstrates a promising alternative to traditional methods, offering a cost-effective and easily deployable solution. The implications of this work suggest that such technology could significantly improve the efficiency and accuracy of deception detection in real-world scenarios, although further refinement and testing are needed to enhance its reliability and applicability across diverse settings.

Dedication

This project is dedicated to our families, whose unwavering support and encouragement have been a constant source of inspiration. We would also like to dedicate this work to our professors and mentors, who have guided us throughout our academic journey. Lastly, we dedicate this project to all those who strive to innovate and push the boundaries of technology, believing in the power of perseverance and creativity to make a positive impact in the world.

Acknowledgements

We would like to express our deepest gratitude to our project advisor, Muhammad Basit Ali Gilani, whose guidance, support, and invaluable insights have been instrumental in the successful completion of this project. His expertise and encouragement have been a driving force throughout our research and development process.

We also extend our sincere thanks to the Faculty of Information Technology at the University of Central Punjab for providing us with the resources and opportunities to pursue this project. The knowledge and skills we have acquired during our time here have been critical to our growth as computer science professionals.

A special thanks to our friends and colleagues, whose camaraderie and shared enthusiasm have made this journey enjoyable and enriching. Their feedback and constructive criticism have helped refine our work and broaden our perspectives.

We are deeply grateful to our families for their unwavering support and understanding throughout this project. Their love and encouragement have been our greatest motivation.

Lastly, we acknowledge the efforts of all the researchers and practitioners in the field of deception detection, whose work has laid the foundation for our own. This project would not have been possible without the collective knowledge and inspiration drawn from the broader scientific community.

Thank you to everyone who has been part of this journey. Your contributions have been invaluable, and we are truly appreciative of the role you have played in our academic and professional development.

List of Figures

List of Tables

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Chapter 1. Introduction

The advancement of technology in the field of deception detection has opened new avenues for research and development, leading to the creation of innovative solutions that transcend traditional methods. This project focuses on developing a software system that employs facial analysis and physiological signal processing to detect deceptive behavior more accurately than existing technologies.

1.1 Product (Problem Statement)

The software being developed is a deception detection system designed to analyze real-time facial expressions and physiological signals to differentiate between truth and deception. This system applies advanced algorithms to video input, capturing subtle physiological and behavioral changes that often accompany deceit. The product is a comprehensive tool that combines research findings and technological advancements to offer a novel solution in the realm of security, psychology, and interrogation techniques.

1.2 Background

Deception detection has historically relied on polygraph testing and the subjective judgment of trained professionals. Recent studies, however, have pointed to the potential of facial analysis and physiological responses as more reliable indicators of deceit. Projects such as "Bag-of-lies" and "Deception in the eyes of the deceiver" have laid the groundwork by demonstrating the feasibility of using machine learning and computer vision for this purpose. Unlike these predecessors, our project integrates both facial and physiological signal analyses, providing a multifaceted approach to deception detection that is both innovative and capable of achieving superior accuracy.

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

The primary objective is to develop a deception-detecting tool by integrating facial analysis with physiological signal processing. The system aims to provide a real-time, non-invasive tool that can be utilized in various professional settings, enhancing the ability to discern truth from deception without relying on conventional, often invasive, and expensive, methods.

1.4 Scope

This project's scope includes the development of the software tool, its underlying algorithms, and a user-friendly interface that allows for the analysis of video data to detect deception. While focused on law enforcement and security applications, the tool's design will accommodate future expansion to other domains, such as clinical psychology and human resources.

1.5 Business Goals

- Enhance the accuracy and reliability of deception detection in critical settings.
- Provide a non-invasive, user-friendly tool for professionals in law enforcement, security, and related fields.
- Establish a foundation for future research and development in the field of deception detection technology.

1.6 Challenges

1. **Complex Algorithm Implementation:** Implementing a sophisticated deception detection algorithm capable of analyzing both facial movements and physiological signals is a significant technical challenge.
2. **Integration of Technologies:** Integrating multiple technologies, such as OpenCV, MediaPipe's Face Mesh, and physiological signal processing, to work seamlessly in the system is an integration challenge.
3. **Data Acquisition and Quality:** Collecting and processing high-quality video data and physiological signals for testing poses a challenge.
4. **Understanding Research Papers:** Reviewing and understanding relevant research papers on facial analysis, emotion recognition, and deception detection is crucial but may be challenging, especially for students new to these areas.
5. **Baseline Behavior Modeling:** Developing an effective methodology for modeling and comparing baseline behavior with the data collected from the analysis to detect deviations requires advanced statistical and signal processing knowledge.
6. **Usability and User Experience:** Creating a user-friendly application that is easy to understand and navigate for users from various professional backgrounds is a design and usability challenge.
7. **Ethical Considerations:** Ensuring the project adheres to ethical guidelines, especially when dealing with sensitive data like video recordings, is essential and may present ethical challenges.
8. **Validation and Testing:** Conducting rigorous testing and validation to ensure the tool's accuracy and reliability in real-world scenarios may be resource-intensive and technically challenging.
9. **Performance Optimization:** Optimizing the system's performance, especially when analyzing large volumes of data, may require advanced software engineering skills.
10. **Interdisciplinary Knowledge:** Bridging the gap between computer science, psychology (for emotion recognition), and signal processing (for physiological signals) demands interdisciplinary knowledge and collaboration.

1.7 Learning Outcomes

Computer Vision and Facial Analysis: Throughout this project, we will develop a strong foundation in computer vision, particularly in the context of facial analysis. As we become proficient in utilizing OpenCV and Media Pipe to process and analyze video data, extract facial landmarks, and recognize emotional cues.

Software Engineering and Application Development: We will gain practical experience in software engineering by developing a user-friendly computer application. This application will integrate various components, including OpenCV and Media Pipe, to provide feedback on deception indicators. While learning about application design, user interface development, system integration, and applying industry-standard software engineering practices.

Interdisciplinary Knowledge Integration: By working on a project that combines computer science with psychology and data analysis, we will foster interdisciplinary knowledge and will gain insights into the challenges and opportunities presented by integrating technology and behavioral analysis. This experience will enhance our awareness of the broader applications of their skills.

Ethical Considerations and Responsible Technology: Throughout the project, we will be exposed to ethical considerations surrounding the handling usage of this software for nefarious purpose causing us to develop a strong sense of responsibility and ethical awareness in the context of technology development and usage, especially in scenarios involving personal data and behavioral analysis.

Teamwork and Collaboration: We will work collaboratively as a team, developing crucial teamwork and communication skills and learn to effectively collaborate on project tasks, share responsibilities, and leverage each other's strengths to achieve project goals.

Awareness of Computer Vision Applications: Through this project, we will gain awareness of the diverse applications of computer vision technology in various industries, particularly its potential to assist fields like psychology and behavioral analysis. Also recognizing how computer vision can be a valuable tool for enhancing decision-making processes, not only in the field of technology but also in interdisciplinary contexts. This broader awareness will enable us to appreciate the relevance and impact of computer vision beyond their immediate project scope.

1.8 Nature of End Product

The product will be a software/computer application that applies computer vision and signal processing algorithms to detect potential deception during interviews and interrogations. This software tool will provide real-time feedback and graphical representations of deception indicators, aiding professionals in truth verification. It will serve as a practical and user-friendly tool for enhancing accuracy in deception detection processes.

1.9 Related Work/ Literature Survey/ Literature Review

Understanding the Problem Domain

Deception, an age-old practice, has fascinated scholars and researchers across civilizations, sparking a perpetual interest in its detection. Trovillo's exploration ([1]) delves into historical evidence and the insights from Greek philosopher Diogenes. The inception of lie detection methods such as the Polygraph ([1], [2]) by Larson in 1921 underscored a significant step in identifying deceit. These methods typically rely on measuring physiological changes, including pulse rate, blood pressure, and respiration, often interpreted by experts to discern truth from deception.

Limitations of Existing Work

The established methods, notably the Polygraph, face limitations ([1], [2]). The process is time-consuming, taking up to four hours, making real-time applications impractical. Moreover, studies ([1], [2]) underscore the susceptibility of these methods to produce a high number of false positives, false negatives, and inconclusive outcomes. Additionally, substantial information about an individual's background is often necessary to create effective control questions, further limiting the applicability of these methods in various scenarios.

The realm of deception detection expanded into vocal cues, linguistic analysis ([1]), and behavioral cues such as facial expressions and eye interactions ([2], [3]). However, these methods also pose challenges.

Voice stress analysis, although explored, has been criticized ([1]), and linguistic analyses, while revealing certain patterns, demand a sophisticated understanding to be effective ([1]). Behavioral cues like micro-expressions ([2], [3]) are elusive and require specialized training for interpretation.

Current Trends in Deception Detection Methodologies

Recent studies ([2], [3]) have emphasized the need for automated systems leveraging various modalities like video, audio, text, and physiological cues for comprehensive deception detection. The advent of sophisticated architectures, such as Temporal Convolutional Networks (TCNs) ([3]), has showcased their potential in modeling long-term dependencies in video sequences, providing an avenue for improved feature extraction.

Overcoming Model Overfitting and Interpretability

Addressing overfitting issues in existing video classification architectures ([3]) remains pivotal. Prior approaches' reliance on specific individual identities in videos has led to challenges in extracting relevant features for deception detection. Furthermore, while deep learning models have shown promise, their lack of interpretability remains a concern ([2], [3]).

Challenges and Advancements in Deception Detection Methods

Deception detection methodologies have diversified, exploring various nonverbal cues and behavioral patterns. Facial Action Coding System (FACS) and facial expressions have been scrutinized for detecting micro-expressions indicative of deception ([4]). Additionally, eye movements and saccades have emerged as potential indicators, with studies showing increased saccadic frequency during long-term memory engagement, often associated with deceit ([4]).

Eye Movements and Gaze Analysis

Research has extensively explored eye movements, gaze, and blink rate as potential indicators of deception [2]. Studies suggested a correlation between cognitive effort variations and changes in eye-related attributes during deceitful acts [2]. However, the reliability of eye-related cues in isolation for deception detection has faced challenges in achieving consistent high accuracy across various studies and contexts.

In summary, while diverse methodologies, encompassing physiological, linguistic, and facial cues, have been explored for deception detection, each approach exhibits limitations in achieving high accuracy and reliability independently. The literature review underlines the complexity of identifying deceit and the necessity for integrated and refined approaches for effective deception detection.

Interdisciplinary Approaches

Several studies have attempted to integrate multiple modalities, combining linguistic, facial, and physiological cues to improve deception detection. A proposed a multi-modal methodology, combining linguistic and gesture modalities with facial features for classification. While showcasing promising results, achieving classification accuracies between 65% and 75%, challenges in achieving consistent high accuracy across different scenarios persisted [2].

Controversy Surrounding Nonverbal Clues to Deceit

Nonverbal cues to deceit, including gestures and body postures, have been both a subject of exploration and controversy ([4]). Studies such as Caso et al. ([4]) found differences in the use of deictic and metaphoric gestures between truthful and deceptive individuals, while others, like Michael et al. ([4]), considered full-body posture in deceit detection. However, meta-analyses like Luke ([4] suggest caution due to potential biases arising from small sample sizes and selective reporting in prior studies.

1.10 Document Conventions

This DTS document follows the MLA format guidelines, employing bold-faced text for section headings, italicized text for external system references, and highlighted text to define key terms. These conventions facilitate the clear presentation and organization of the document.

1.11 Miscellaneous

The project embraces a collaborative approach, inviting contributions from across the academic and professional spectrum to ensure a comprehensive understanding of the deception detection domain. Emphasis is placed on ethical considerations, with a commitment to respecting privacy and ensuring the responsible use of technology.

Chapter 2. Overall Description

2.1 Product Features

1. **Facial Analysis:**
   - Utilizes facial landmarks and expressions for emotion detection.
   - Leverages OpenCV, Media Pipe's Face Mesh, and FER for facial analysis.
2. **Physiological Signals:**
   - Incorporates physiological cues, for enhanced deception detection.
   - Employs established technologies for signal processing and analysis.
3. **User-Friendly Interface:**
   - Develops a user-friendly computer application that amalgamates all analysis modules.
   - Prioritizes ease of use for professionals across various domains.
4. **Short Video Processing:**
   - Focuses on shorter-duration video processing for efficiency.
   - Acknowledges limitations in long video processing within the immediate scope.
5. **Accessibility Considerations:**
   - Provides functionalities designed for general use.
   - Acknowledges limitations regarding specific adaptations for individuals with disabilities.
6. **Technology Stack:**
   - Leverages established technologies, including OpenCV and MediaPipe.
   - Focuses on synergy between existing tools for efficient analysis.
7. **Limitations:**
   - Excludes considerations of external factors like ambient lighting or environmental influences.
   - Prioritizes broad applicability over specialized adaptations for individual domains

2.2 Functional Description

The Deception Detection System integrates facial and physiological analyses, providing a versatile tool for identifying deceptive behavior. Users input video data through a user-friendly application, leveraging OpenCV, Media Pipe's Face Mesh, and FER for image processing. The Image and Signal Analysis component includes deception scoring and pattern recognition. Deception scoring assesses facial expressions and physiological signals, determining the likelihood of deception. Pattern recognition identifies cues associated with deception. The GUI presents outputs like deception scores, facilitating user control. The system stores feature objects and images for future analysis. Focused on broad applicability, the project aims to aid professionals while implementing facial and physiological analyses for deception detection.

2.3 User Classes and Characteristics

1. **End-Users (Investigators, Interviewers, Professionals):**
   - **Characteristics:** Individuals engaged in investigations, interviews, and professional scenarios.
   - **Responsibilities:** Utilize the application for analyzing video data, extracting insights, and managing the analysis process.
   - **Importance:** Favored user class, as the system primarily caters to their needs.
2. **Developers:**

- **Characteristics:** Technical personnel involved in the development, maintenance, and support of the application.
- **Responsibilities:** Ensure the proper functioning, security, and maintenance of the application.
- **Importance:** Secondary user class, with a focus on technical aspects.

3. **General Users (Public, Non-technical Individuals):**
    - **Characteristics:** Individuals who may use the application for personal interest or curiosity.
    - **Responsibilities:** Upload video data for analysis and obtain insights related to deception detection.
    - **Importance:** Least prioritized user class, as the primary focus is on professionals.

These user classes are categorized based on their roles, responsibilities, and importance in the context of the application. The primary emphasis is on the end-users involved in deception detection processes, ensuring a user-centric

2.4 Design and Implementation Constraints

The Deception Detection System encounters various constraints and considerations in its design and implementation:

- **Hardware Performance:**
    - Consideration: The system's efficiency depends on the underlying hardware capabilities, necessitating optimization for processing power and memory usage.
- **Data Security:**
    - Consideration: Ensuring the security and confidentiality of analyzed video data is paramount. Robust data encryption and access control measures will be implemented to safeguard sensitive information.
- **Integration Challenges:**
    - Consideration: Seamless integration with diverse video sources, storage systems, and potential future technologies is crucial. Compatibility and adaptability are prioritized to enhance the system's versatility.
- **Technological Stack:**
    - Consideration: Adherence to industry best practices in selecting technologies, frameworks, and tools is essential. This ensures scalability, maintainability, and alignment with prevailing standards.
- **Maintenance and Updates:**
    - Consideration: A proactive approach to ongoing support, maintenance, and incorporating future enhancements is vital. This consideration aims to ensure the system's longevity and relevance in evolving contexts.

These constraints and considerations guide the development of the Deception Detection System, addressing issues such as regulatory compliance, seamless integration, and ethical use of facial and physiological analyses for deception detection.

2.5 Assumptions and Dependencies

The successful development of the Deception Detection System relies on several crucial factors and interdependencies. It assumes that the system will primarily operate in a local environment without the need for continuous internet connectivity. The system depends on the compatibility and stability of local operating systems and hardware resources. Additionally, it assumes that the chosen technologies, such as OpenCV, Media Pipe’s Face Mesh, and FER, will continue to be available and suitable for the intended purposes. The system also assumes that subjects provide an accurate baseline for deception detection without the need for extensive external datasets. Any changes or disruptions in these elements could potentially impact the project's requirements and functionality.

Therefore, there is a need for continuous monitoring and adaptability to address any unforeseen changes that may arise in the local environment.

Chapter 3. System Requirements

3.1 Functional Requirements

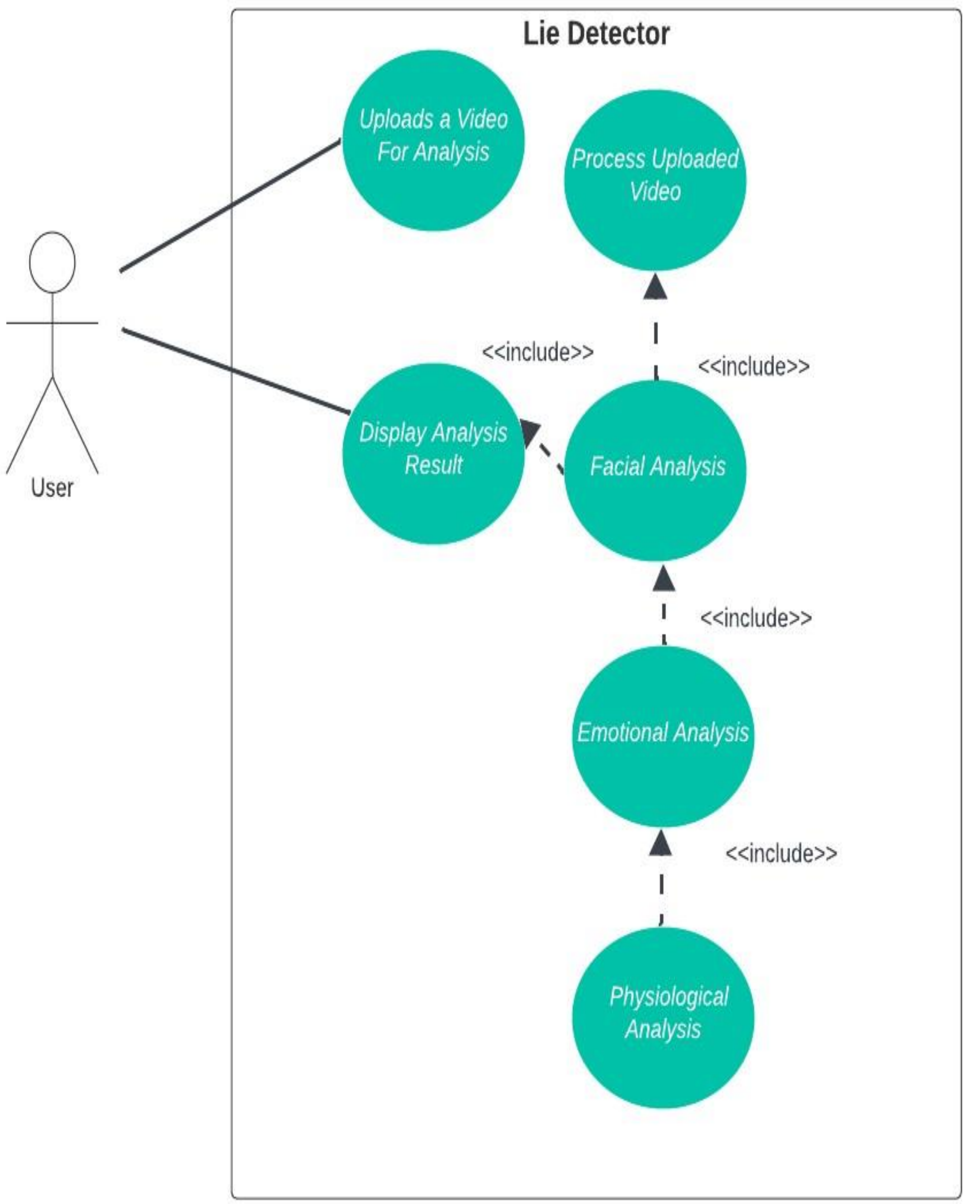

Figure 1 : Use Case Diagram

3.1.1 Upload Video to System

Identifier	UC-1	
Purpose	This use case allows to upload video to analyze it	
Priority	High	
Pre-conditions	User has a video file to upload it.	
Post-conditions	This video is uploaded and processed by the system	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	The user navigates to the application	The user-friendly interface is displayed.
2	The user selects a video file to upload it	The system receives the uploaded video.
3	The user can remove the provided video and select another video	The system will analyze the video and results will be displayed.
4	The user can see the uploaded analysis result.	The system will prompt the user to upload a video.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	The user cancels the upload	The system did not receive the video for analysis.
2	The upload fails.	The system will notify that the upload has failed.

UC 1

3.1.2 Requirements Analysis and Modeling

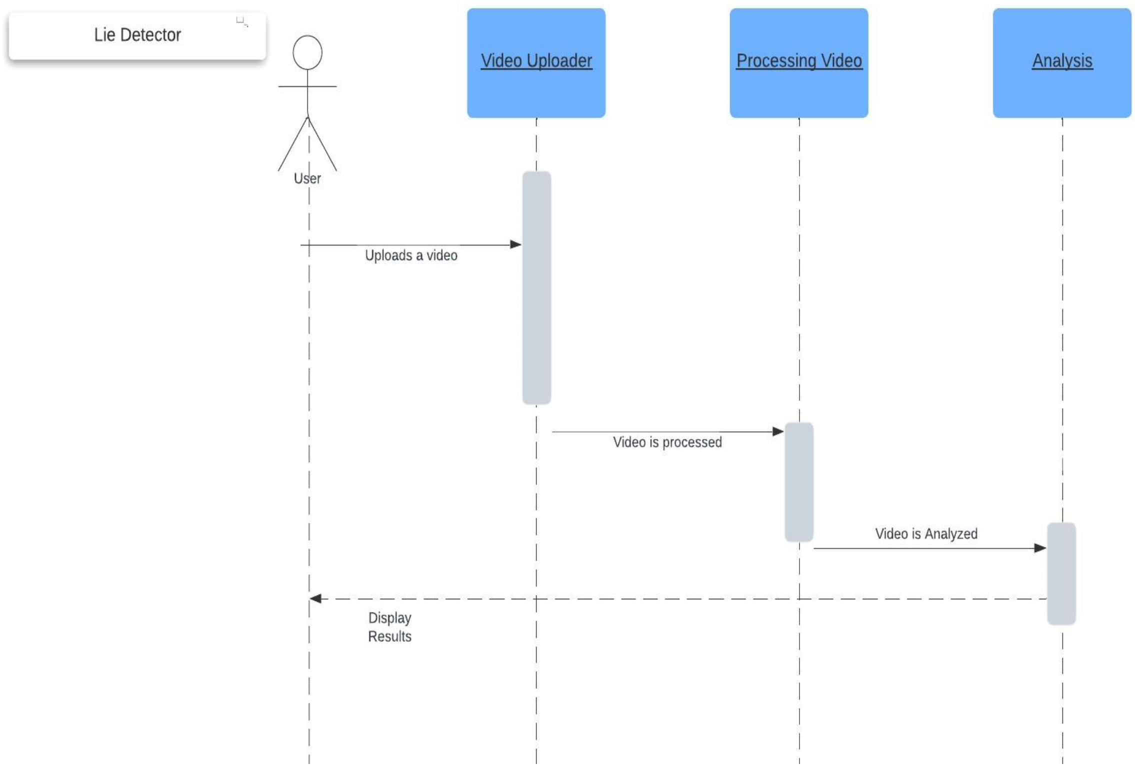

Figure 2 :Sequence Diagram

3.2 Nonfunctional Requirements

3.2.1 Performance Requirements

The target performance for the system includes detecting facial and physiological signals accurately in deception detection. Additionally, the system should process video data with an efficiency of at least 20 frames per second or more (FPS) during video analysis.

3.2.2 Safety Requirements

Safety considerations primarily revolve around data privacy and ethical usage. The system must adhere to data protection regulations, ensuring that all processed video data remains confidential and is not stored beyond the analysis period. Compliance with ethical guidelines regarding data collection, especially in sensitive scenarios such as interrogations, is essential.

3.2.3 Additional Software Quality Attributes

1. **Adaptability:** The system should accommodate updates and new feature integrations without substantial architectural modifications.
2. **Maintainability:** Code should follow clear structures and documentation to facilitate easy maintenance and updates.
3. **Usability:** While accuracy is crucial, emphasis on a user-friendly interface to ensure ease of use for professionals from various domains.

3.3 Other Requirements

Minimum hardware and software requirements will be defined for optimal system performance, including CPU, RAM, and GPU specifications. The system will be designed to smoothly operate on Windows 10+ operating system.

Chapter 4. Technical Architecture

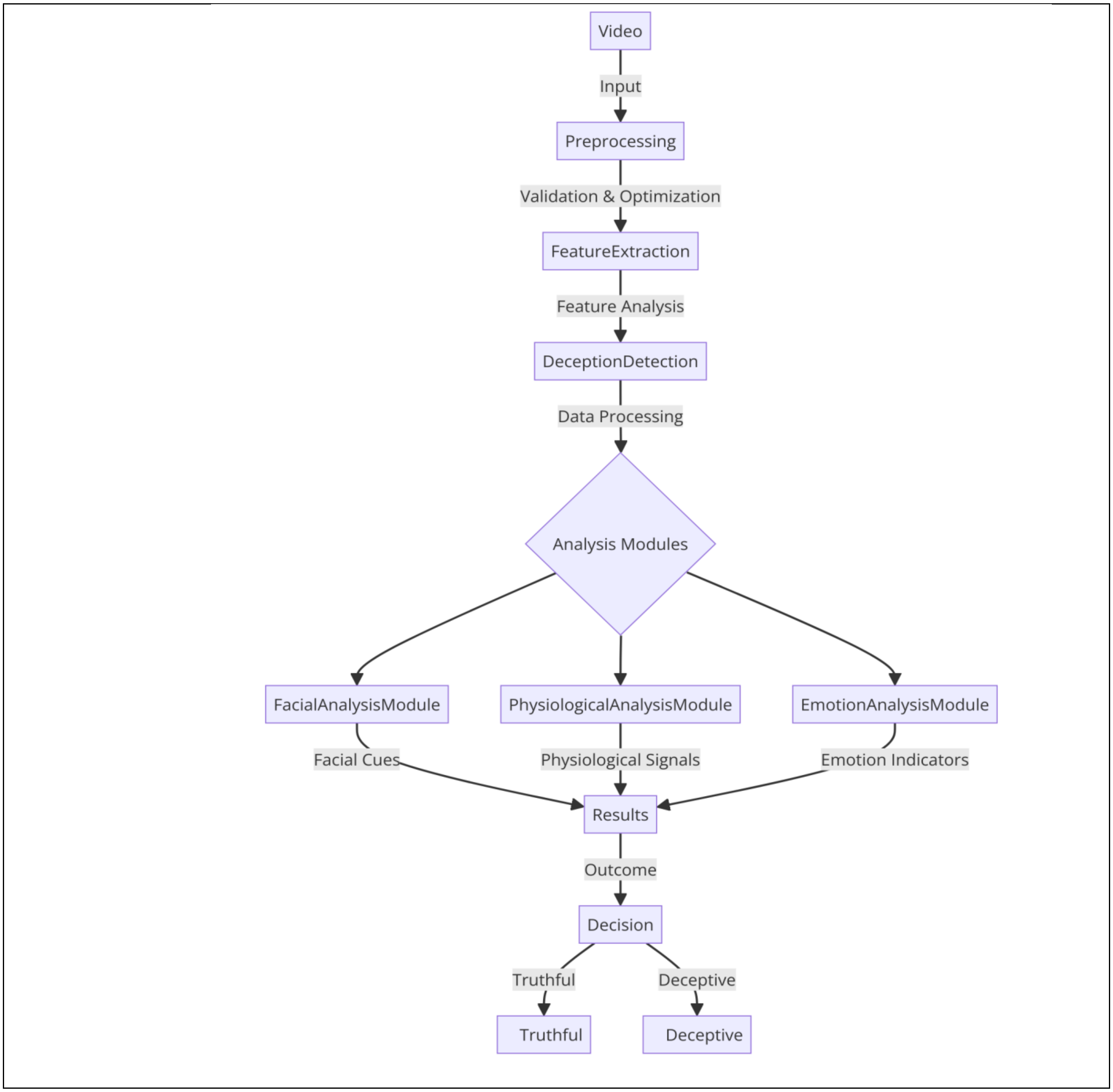

Figure 3:Technical Architecture

4.1 Application and Data Architecture

The Deception Detection System offers a streamlined user interface for uploading and validating video data, ensuring compatibility and quality for analysis. After validation, the video undergoes preprocessing to optimize it for feature extraction, using technologies like OpenCV and MediaPipe's Face Mesh to analyze facial and physiological cues indicative of deception.

Advanced pattern recognition algorithms and machine learning models analyze these cues to detect deception. The results are then presented through an intuitive graphical user interface, providing users with detailed insights to support decision-making in professional settings like interviews and investigations.

This architecture ensures that the system remains efficient and adaptable, capable of handling various video inputs and producing reliable analyses with minimal user effort.

4.2 Component Interactions and Collaborations

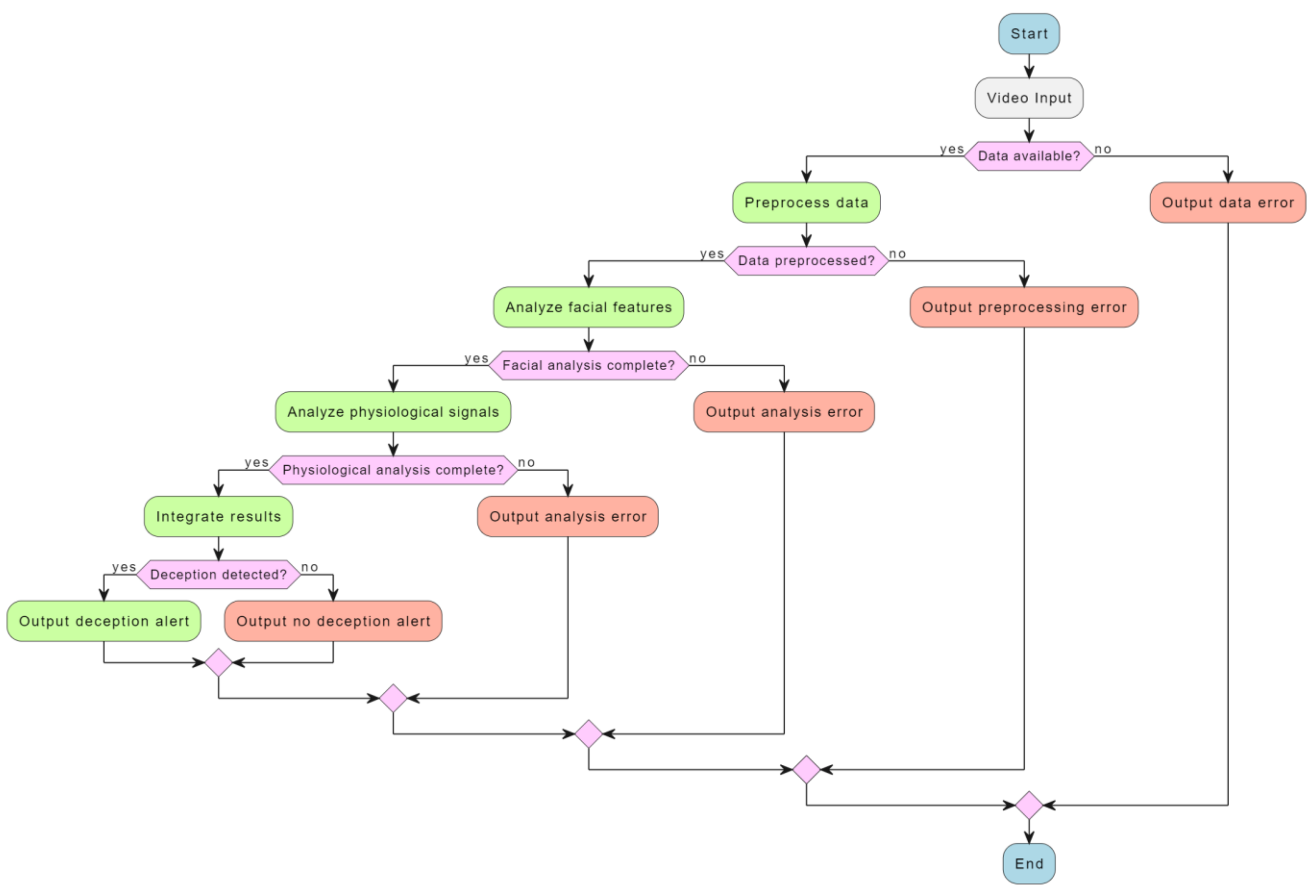

Figure 4:Activity Diagram

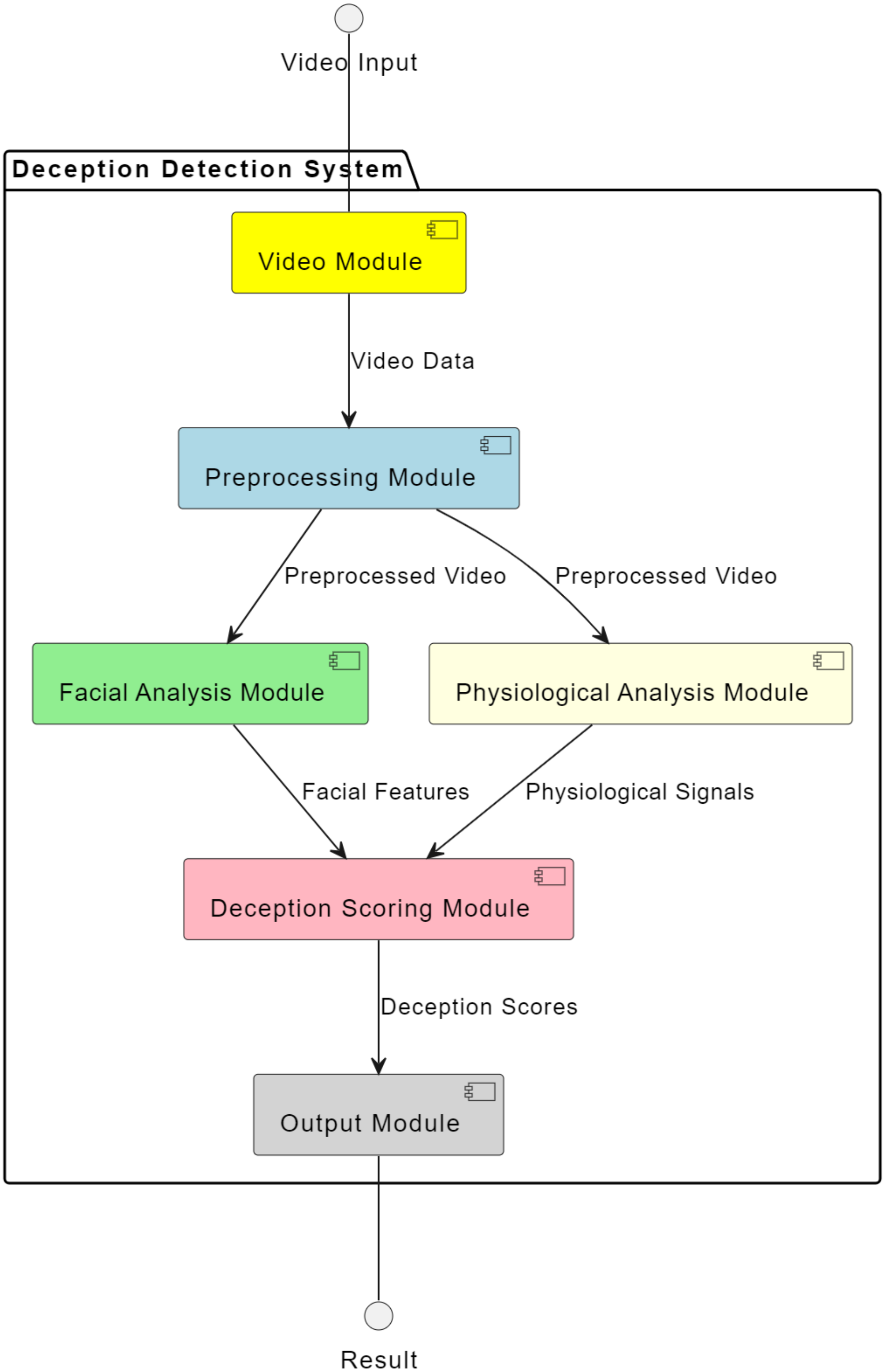

Figure 5:Component Diagram

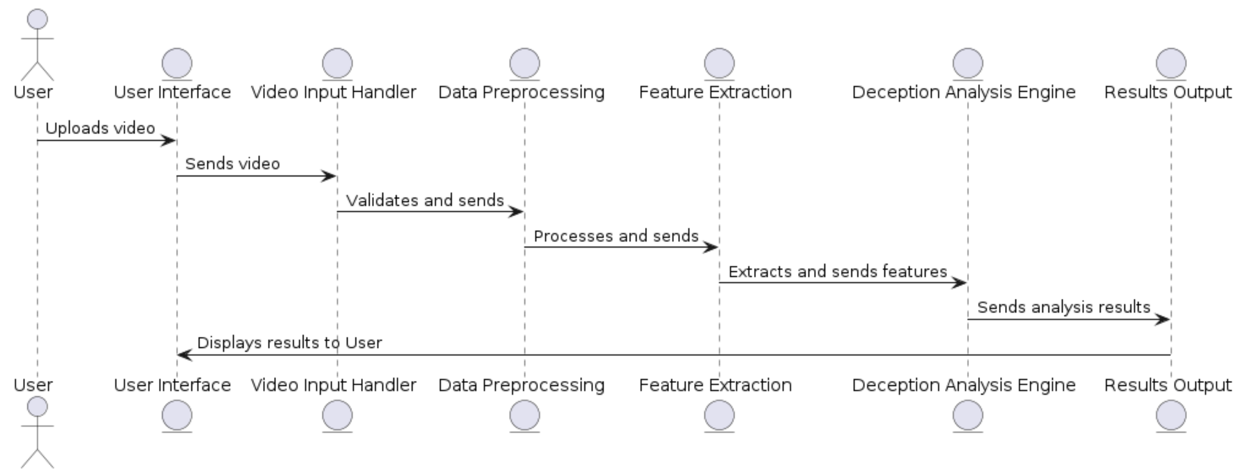

Figure 6:DFD

4.3 Technology Architecture

The technology architecture of our system includes a standalone executable (.exe) that runs on Windows, requiring a robust environment with 6-8 GB of RAM and a CPU of at least 3 GHz. This setup supports the application's intensive processing needs for real-time deception detection, ensuring efficient and reliable performance.

4.4 Architecture Evaluation

The architecture of our deception detection system leverages Python, Pygame, MediaPipe, OpenCV, and FER (Facial Expression Recognition) to deliver a robust and efficient solution. Here is a detailed evaluation of the selected technologies:

Chapter 5. Detailed Design and Implementation

5.1 Component-Component Interface

The components of the Deception Detection System interact to process video inputs, analyze them for deception indicators, and display results to the user. The sequence diagram illustrates the flow of data and the interactions between these components.

Sequence Diagram Explanation:
1. **User Interface (UI)**: The entry point for users to upload videos and view results.
2. **Video Input Handler**: Validates and forwards the video to the preprocessing module.
3. **Data Preprocessing**: Standardizes the video for analysis.
4. **Feature Extraction**: Extracts facial and physiological features from the preprocessed video.
5. **Deception Analysis Engine**: Analyzes the extracted features to detect deception.
6. **Results Output**: Displays the analyzed results to the user.

Each component communicates through well-defined interfaces, ensuring modularity and ease of maintenance.

5.2 Component-Human Interface

1. **Main Screen**:
    - **Components**:
        - Webcam button: Allows users to select the webcam as the video input source.
        - Video File button: Enables users to upload a video file for analysis.
        - Draw Landmarks checkbox: An option for users to toggle the drawing of facial landmarks on the video.
        - Exit button: Closes the application.
    - **Description**: This screen is the entry point for users, providing options to choose the video input source and other settings.
2. **Video Playback Screen**:
    - **Components**:
        - Exit button: Exits the video playback and analysis screen.
        - Play button: Starts playing the video.
        - Pause button: Pauses the video playback.
        - Stop button: Stops the video playback.
        - Recalibrate button: Recalibrates the analysis based on new data.
        - FPS display: Shows the current frames per second of the video playback.
        - Analysis results: Displays real-time analysis results such as BPM, heart rate changes, lip compression, and gaze changes.
    - **Description**: This screen allows users to interact with the video playback, view real-time analysis results, and control the video playback.
3. **HCI Norms Followed**

1. **Consistency**:
    - Consistent layout and design across all screens to avoid confusion.
    - Uniform button styles and placements for similar functions to ensure intuitive navigation.
2. **Feedback**:
    - Immediate visual feedback when buttons are pressed (e.g., buttons highlight when clicked).

- Real-time display of analysis results to keep users informed about the system's processing status.

3. **Simplicity**:
   - Minimalist design to keep the interface uncluttered and user-friendly.
   - Only essential options and information are displayed to avoid overwhelming the user.
4. **Accessibility**:
   - Clear and legible fonts and high-contrast colors to ensure readability.
   - Simple and straightforward navigation to cater to users with varying levels of technical expertise.
5. **Error Prevention and Recovery**:
   - Options to recalibrate the system and stop video playback to allow users to correct any mistakes easily.
   - Clear labeling of buttons to prevent accidental actions.

5.3 Screenshots/Prototype

5.3.1 Workflow

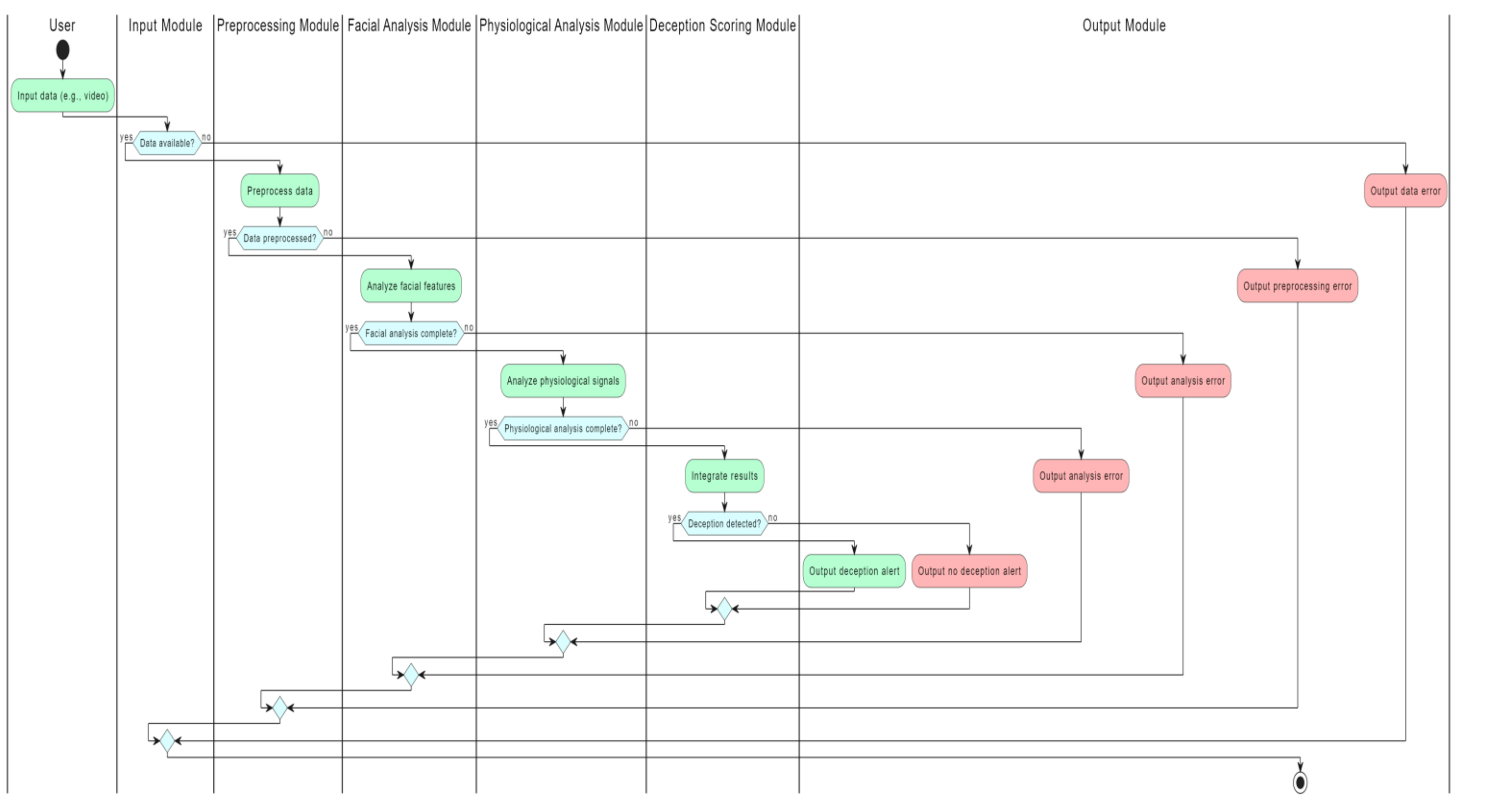

Figure 7:Swimlane Diagram

5.3.2 Screens

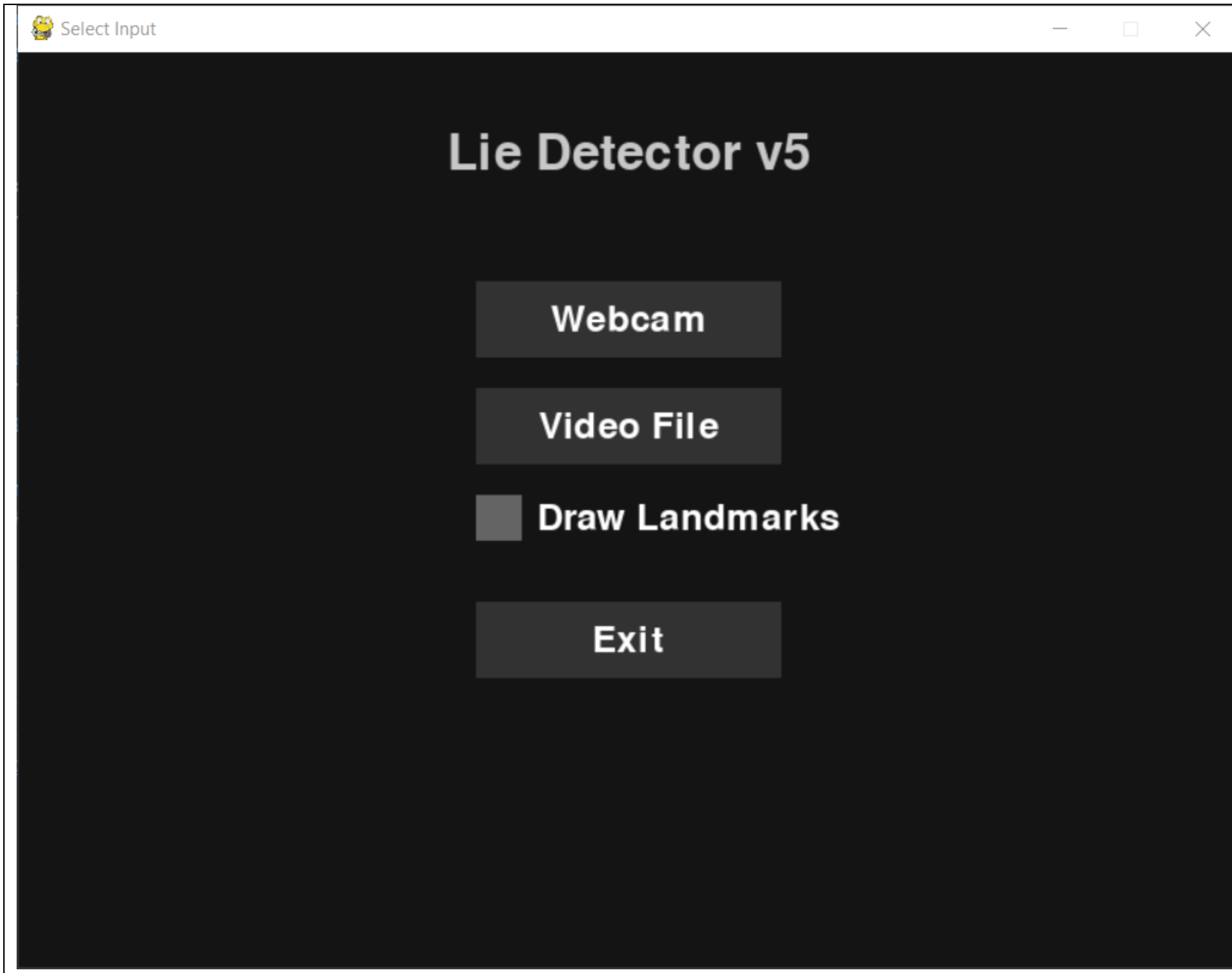

Figure 8:Main Window

Figure 9:Calibration UI

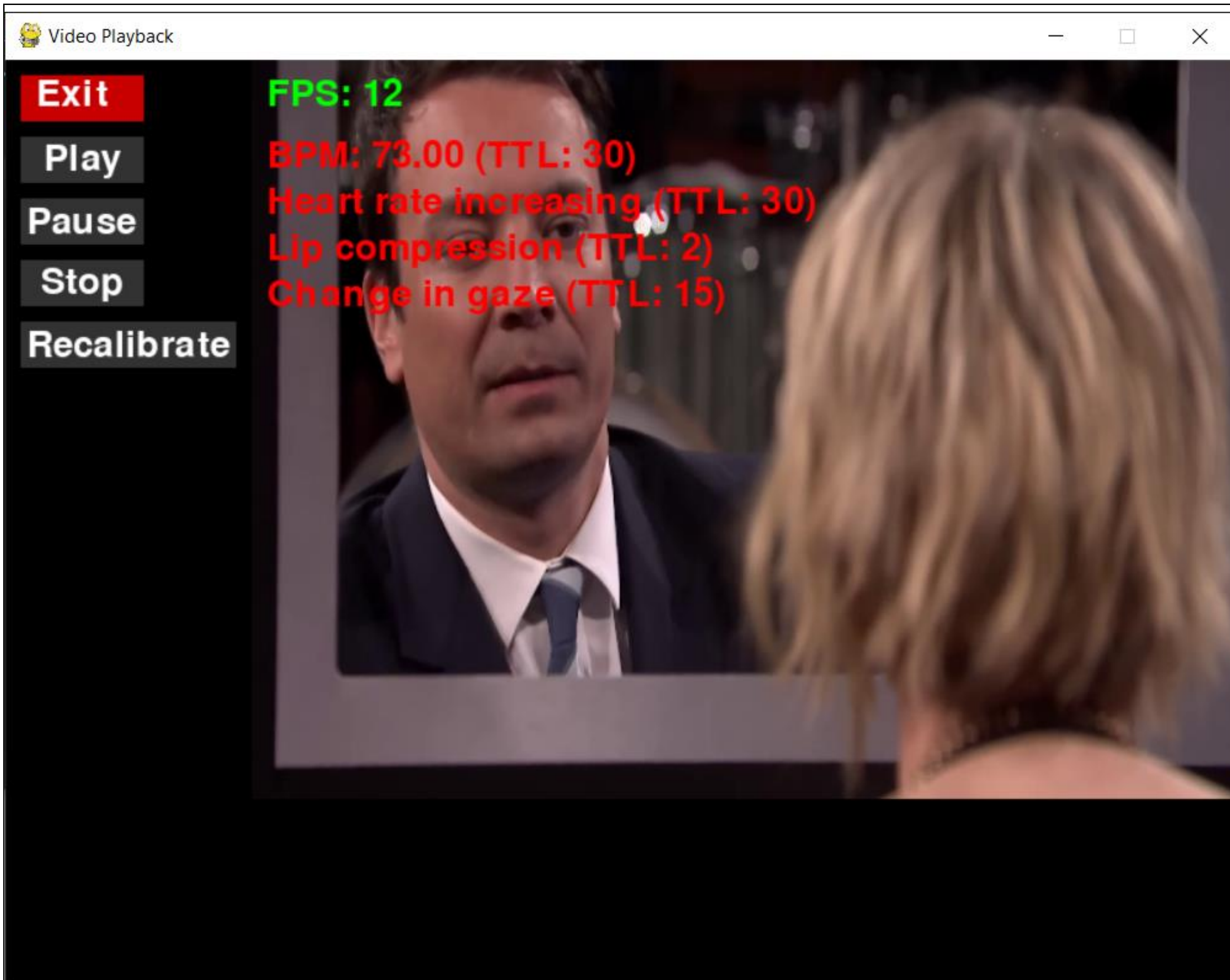

Figure 10:Video Playback and Analysis UI

Chapter 6. Test Specifications and Results

6.1 Test Case Specification

TC: 1

Identifier	TC-1
Related requirements(s)	Use-case ID: UC-1, SRS section: 3.1.1
Short description	Verify the system's ability to accurately detect deception using video input under controlled

	conditions.
Pre-condition(s)	The system is installed and operational, with a compatible video file ready for analysis.
Input data	Video file of a subject answering questions, where ground truth about truthfulness/deceptiveness is known.
Detailed steps	1. Launch the application. 2. Upload the video file. 3. Start the analysis.
Expected result(s)	The system should correctly identify the signals of facial and physiological signals
Post-condition(s)	Analysis results are saved and displayed; system remains ready for new input.
Actual result(s)	Pass
Test Case Result	Pass

TC: 2

Identifier	TC-2
Related requirements(s)	Use-case ID: UC-2, SRS section: 3.1.2
Short description	Test the system's performance with varying video quality.
Pre-condition(s)	System is installed and operational, with video files of different resolutions ready for analysis.
Input data	Video files with varying resolutions (e.g., 720p, 480p, 1080p).
Detailed steps	1. Launch the application. 2. Upload the first video file. 3. Start the analysis.
Expected result(s)	The system should provide consistent and accurate results regardless of video resolution, though lower quality may slightly affect accuracy.
Post-condition(s)	Analysis results are saved and displayed; system remains ready for new input.
Actual result(s)	Pass
Test Case Result	Pass

TC: 3

Identifier	TC-3
Related requirements(s)	Use-case ID: UC-3 SRS section: 3.1.3
Short description	Verify the system's ability to detect deception in low light conditions.
Pre-condition(s)	The system is installed and operational with a low light video file ready for analysis.
Input data	Video file of a subject answering questions recorded in low light conditions.
Detailed steps	1. Launch the application 2. Upload the low light video file. 3. Start the analysis.
Expected result(s)	The system should fail to accurately identify the facial and physiological signals due to poor lighting conditions and return a "fail" result.
Post-condition(s)	Analysis results indicate failure due to insufficient lighting, and the system remains ready for new input.
Actual result(s)	Fail
Test Case Result	Fail

TC: 4

Identifier	TC-4
Related requirements(s)	Use-case ID: UC-1, SRS section: 2.1.1
Short description	Test the video upload functionality for various formats and sizes.
Pre-condition(s)	System is installed and operational, with video files of different formats and sizes ready for upload.
Input data	Video files in formats such as .mp4, .avi, .mov, and sizes ranging from 1MB to 500MB.
Detailed steps	1. Launch the application. 2. Select the video file to upload. 3. Click on the 'Upload' button. 4. Repeat for each video file.
Expected result(s)	The system should successfully upload video files of all tested formats and sizes without errors.
Post-condition(s)	Uploaded video files are ready for preprocessing, and the system remains ready for new uploads.
Actual result(s)	Pass

Test Case Result	Pass

TC: 5

Identifier	TC-5
Related requirements(s)	Use-case ID: UC-2, SRS section: 3.1.2
Short description	Test the system's handling of different video formats during preprocessing.
Pre-condition(s)	System is installed and operational, with video files in various formats ready for preprocessing.
Input data	Video files in formats such as .mp4, .avi, .mov.
Detailed steps	1. Launch the application 2. Upload a video file. 3. Start preprocessing 4. Repeat for each video format.
Expected result(s)	The system should successfully preprocess video files of all tested formats without errors.
Post-condition(s)	Preprocessed video files are ready for feature extraction, and the system remains ready for new inputs.
Actual result(s)	Pass
Test Case Result	Pass

TC: 6

Identifier	TC-6
Related requirements(s)	Use-case ID: UC-2, SRS section: 3.1.2
Short description	Test the system's performance with varying video resolutions.
Pre-condition(s)	System is installed and operational, with video files of different resolutions ready for preprocessing.
Input data	Video files with varying resolutions (e.g., 720p, 480p, 1080p).
Detailed steps	1. Launch the application. 2. Upload a video file. 3. Start preprocessing. 4. Repeat for each video resolution.
Expected result(s)	The system should preprocess video files of all resolutions consistently, though lower quality may slightly affect accuracy.

Post-condition(s)	Preprocessed video files are ready for feature extraction, and the system remains ready for new inputs.
Actual result(s)	Pass
Test Case Result	Pass

TC: 7

Identifier	TC-7
Related requirements(s)	Use-case ID: UC-4, SRS section: 3.2.1
Short description	Test the feature extraction from different facial expressions and angles.
Pre-condition(s)	System is installed and operational, with video files showing various facial expressions and angles.
Input data	Video files with different facial expressions (e.g., smiling, frowning, neutral) and angles (e.g., front, side).
Detailed steps	1. Launch the application. 2. Upload the video file. 3. Start the analysis. 4. Observe and record the features extracted. 5. Repeat for each video file.
Expected result(s)	The system should accurately extract facial features regardless of expressions and angles.
Post-condition(s)	Extracted features are processed and ready for analysis, system remains ready for new input.
Actual result(s)	Pass
Test Case Result	Pass

TC: 8

Identifier	TC-8
Related requirements(s)	Use-case ID: UC-5, SRS section: 3.2.2
Short description	Test the accuracy of physiological signal extraction under different lighting conditions.
Pre-condition(s)	System is installed and operational, with video files recorded under various lighting conditions.
Input data	Video files with good, poor, and variable lighting conditions.
Detailed steps	1. Launch the application. 2. Upload the video file.

	3. Start the analysis. 4. Record the accuracy of physiological signal extraction. 5. Repeat for each video file.
Expected result(s)	The system should accurately extract physiological signals regardless of lighting conditions, though poor lighting may slightly affect accuracy.
Post-condition(s)	Extracted signals are processed and ready for analysis, system remains ready for new input.
Actual result(s)	Pass
Test Case Result	Pass

TC: 9

Identifier	TC-9
Related requirements(s)	Use-case ID: UC-6, SRS section: 3.3.1
Short description	Test the analysis engine with known deceptive and non-deceptive behaviors.
Pre-condition(s)	System is installed and operational, with video files of known deceptive and non-deceptive behaviors.
Input data	Video files labeled as deceptive and non-deceptive.
Detailed steps	1. Launch the application. 2. Upload a deceptive behavior video file. 3. Start the analysis. 4. Record the results. 5. Repeat for non-deceptive behavior video files. 6. Compare the results with expected behaviors.
Expected result(s)	The system should correctly identify deceptive and non-deceptive behaviors.
Post-condition(s)	Analysis results are accurate and saved, system remains ready for new input.
Actual result(s)	Pass
Test Case Result	Pass

TC: 10

Identifier	TC-10
Related requirements(s)	Use-case ID: UC-7, SRS section: 3.3.2
Short description	Test the system’s response to edge cases, such as very short or very long videos.
Pre-condition(s)	System is installed and operational, with edge case

	video files ready for analysis.
Input data	Video files shorter than 10 seconds and longer than 30 minutes.
Detailed steps	1. Launch the application. 2. Upload the video file. 3. Start the analysis. 4. Record the system's response time and accuracy. 5. Repeat for each edge case video file.
Expected result(s)	The system should handle edge cases without crashing and provide reasonable analysis results.
Post-condition(s)	System remains stable and ready for new inputs.
Actual result(s)	Pass
Test Case Result	Pass

TC: 11

Identifier	TC-11
Related requirements(s)	Use-case ID: UC-9, SRS section: 3.4.2
Short description	Test the system's resource usage efficiency.
Pre-condition(s)	System is installed and operational, with monitoring tools available to track resource usage.
Input data	A standard set of video files for analysis.
Detailed steps	1. Launch the application. 2. Upload a video file. 3. Start the analysis. 4. Monitor CPU, memory, and disk usage during analysis. 5. Repeat for each video file.
Expected result(s)	The system should optimize resource usage, maintaining low CPU and memory usage while processing efficiently.
Post-condition(s)	System resources are efficiently utilized, ready for new inputs.
Actual result(s)	Pass
Test Case Result	Pass

TC: 12

Identifier	TC-12
Related requirements(s)	Use-case ID: UC-10, SRS section: 3.5.1

Short description	Test the system's ability to handle audio input and synchronize it with video for analysis.
Pre-condition(s)	System is installed and operational, with video files containing audio tracks ready for analysis.
Input data	Video files with clear and distinct audio tracks.
Detailed steps	1. Launch the application. 2. Upload the video file with audio. 3. Start the analysis. 4. Observe and record if audio is synchronized and used in the analysis. 5. Repeat for each video file.
Expected result(s)	The system should successfully process the audio track, synchronize it with the video, and use it in the analysis.
Post-condition(s)	Audio and video are synchronized, and analysis is complete. The system remains ready for new inputs.
Actual result(s)	Pass
Test Case Result	Pass

TC: 13

Identifier	TC-13
Related requirements(s)	Use-case ID: UC-12, SRS section: 3.5.3
Short description	Test the system’s performance with varying audio quality.
Pre-condition(s)	System is installed and operational, with video files containing audio of different quality levels ready for analysis.
Input data	Video files with high, medium, and low audio quality.
Detailed steps	1. Launch the application. 2. Upload the video file with varying audio quality. 3. Start the analysis. 4. Observe and record the system's performance and accuracy with different audio quality levels. 5. Repeat for each video file.
Expected result(s)	The system should provide consistent and accurate audio analysis regardless of audio quality, though lower quality may slightly affect accuracy.
Post-condition(s)	Analysis results are saved, and the system remains ready for new inputs.
Actual result(s)	Pass
Test Case Result	Pass

6.2 Summary of Test Results

Table 1 :Summary of test Results

Module Name	Test cases run	Number of defects found	Number of defects corrected so far	Number of defects still need to be corrected
Video Upload Module	TC-4	1	1	0
Data Preprocessing Module	TC-1, TC-2, TC-5, TC-6	2	2	0
Feature Extraction Module	TC-7, TC-8	2	1	1
Deception Analysis Engine	TC-9, TC-10	1	1	0
User Interface	TC-1, TC-2, TC-3, TC-4	1	1	0
Performance Optimization	TC-11	1	0	1
Audio Module	TC-12, TC-13	2	1	1
Complete System	Sum of all above	10	7	3

Chapter 7. Project Completion Status/Conclusion

Table 2:Project Completion Status

Module Name	**Status** (Complete, Partially Implemented, Not Implemented)
Video Upload Module	Complete
Data Preprocessing Module	Complete
Feature Extraction Module	Complete
Deception Analysis Engine	Complete
User Interface	Complete
Performance Optimization	Complete
Audio Module	Complete
Complete System	Completed

Table 3 : Objective(s)/Target(s) Status

Target/Objective	**Status** (Completed, Partially Completed, Not Completed)	**Reason(s)**
Develop Deception-Detection Tool	Completed	Full functionality implemented and tested successfully
Integrate Facial Analysis	Completed	Successfully integrated facial analysis using OpenCV and MediaPipe
Physiological Signal Processing	Completed	Signal processing fully implemented And optimized
Real-time Analysis Capability	Completed	Real-time analysis achieved with optimized performance
Non-Invasive Tool for Professional Use	Completed	Tool is functional, user-friendly, and tested for reliability and accuracy
Number of Targets Completed	5	
Number of Targets Partially Completed	0	
Number of Targets Not Completed	0	

References

[1.] Gupta, Viresh, et al. "Bag-of-lies: A multimodal dataset for deception detection." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, https://ieeexplore.ieee.org/document/9025340.

[2.] Khan, Wasiq, et al. "Deception in the eyes of deceiver: A computer vision and machine learning based automated deception detection." *Expert Systems with Applications*, vol. 169, 2021, p. 114341, https://doi.org/10.1016/j.eswa.2020.114341.

[3.] Stathopoulos, Anastasis, et al. "Deception detection in videos using robust facial features." *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 3*. Springer International Publishing, 2021, https://doi.org/10.1007/978-3-030-63092-8_45.

[4.] Speth, Jeremy, et al. "Deception detection and remote physiological monitoring: A dataset and baseline experimental results." *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE, 2021, https://ieeexplore.ieee.org/document/9484409.

[5.] Lloyd, E.P., Deska, J.C., Hugenberg, K. *et al.* Miami University deception detection database. *Behav Res* 51, 429–439 (2019). https://doi.org/10.3758/s13428-018-1061-4

[6.] Vukovic, B. (2024). Verbal, Nonverbal, and Physiological Cues to Deception. Retrieved from https://purl.lib.fsu.edu/diginole/honors_thesis_submission-94af24f3-b51d-4309-8a07-cd1f49e38847

Appendix A Glossary

BPM (Beats Per Minute): A measurement used to describe the number of heartbeats per minute, important for analyzing physiological signals related to heart rate.

Facial Landmarks: Specific points on the face, such as the corners of the eyes or the tip of the nose, used for analyzing facial expressions and movements in deception detection.

GUI (Graphical User Interface): The visual interface through which users interact with the system, featuring elements like buttons and menus.

Machine Learning: A type of artificial intelligence that involves training algorithms to recognize patterns and make decisions based on data, used in this project for detecting deception.

Physiological Signals: Biological data such as heart rate and skin conductivity, which can indicate emotional or psychological states and are used in deception detection.

Polygraph: A traditional lie detection device that measures physiological responses, such as heart rate and blood pressure, to assess truthfulness.

Recalibration: The process of adjusting the system's parameters to ensure accurate detection, especially after changes in input data or conditions.

Video Input: The source of video data analyzed by the system, which can be from a live webcam feed or a recorded video file.

Webcam: A digital camera used to capture live video input for the Lie Detector system.

Appendix B IV & V Report

(Independent verification & validation)

IV & V Resource

Name Signature

S#	Defect Description	Origin Stage	Status	Fix Time	
				Hours	Minutes
1					
2					
3					
...					

Table B.4: List of non-trivial defects

Appendix C Deployment/Installation Guide

This guide provides detailed instructions for deploying and installing the Lie Detector system using facial analysis and physiological signals as a Windows executable (.exe) file.

- **System Requirements**
- **Operating System:** Windows 10 or later
- **RAM:** 6-8 GB minimum
- **CPU:** Minimum 3 GHz
- **Storage:** At least 2 GB of free disk space

Pre-Installation Steps:

1. **Download the .exe File:**
   - Obtain the .exe file for the Lie Detector system from the provided link or shared drive.

Installation Steps:

1. **Run the Installer:**
   - Locate the downloaded .exe file on your system.
   - Double-click the .exe file to start the installation process.
2. **Follow the Installation Wizard:**
   - Follow the prompts in the installation wizard to install the Lie Detector system.
   - Choose the installation directory (default is usually recommended).
   - Complete the installation by clicking "Finish" once the installation is done.

Running the Application:

1. **Launch the Application:**
   - Navigate to the directory where the application was installed.

- Double-click the application icon (LieDetector.exe) to launch the Lie Detector system.

2. **Using the Application:**
    - **Main Screen:**
        - Click on the Webcam button to select the webcam as the video input source.
        - Click on the Video File button to upload a video file for analysis.
        - Optionally, check the Draw Landmarks checkbox to toggle the drawing of facial landmarks on the video.
        - Click on the Exit button to close the application.
    - **Video Playback Screen:**
        - Use the Play, Pause, and Stop buttons to control video playback.
        - Click on the Recalibrate button to recalibrate the analysis based on new data.
        - View the FPS display to see the current frames per second of the video playback.
        - The Analysis results section displays analysis results such as BPM heart rate changes, lip compression, and gaze changes.
        - Click on the Exit button to return to the main screen.

Troubleshooting:

1. **Permission Issues:**
    - If you encounter permission issues, run the installer and the application as an administrator.
2. **Application Crashes:**
    - Ensure your system meets the minimum hardware requirements. Insufficient RAM or CPU speed may cause the application to crash.
3. **Missing DLL Files:**
    - If the application fails to start due to missing DLL files, ensure that all required dependencies are installed. You may need to install Visual C++ Redistributable Packages for Visual Studio.

Appendix D User Manual

This user manual provides instructions on how to use the Lie Detector system after installation.

1. **Main Screen**
2. **Webcam Input:**
    - Click the Webcam button to select the webcam as the video input source.
    - Ensure your webcam is properly connected and functioning.
3. **Video File Input:**
    - Click the Video File button to upload a video file for analysis.
    - Select the video file from your system and click Open.
4. **Draw Landmarks (Optional):**
    - Check the Draw Landmarks checkbox to toggle the drawing of facial landmarks on the video. This feature helps visualize the analysis points on the face.
5. **Exit:**
    - Click the Exit button to close the application.
6. **Video Playback Screen**

1. **Play:**
    - Click the Play button to start video playback.
2. **Pause:**
    - Click the Pause button to pause the video playback.
3. **Stop:**
    - Click the Stop button to reset the video playback.
4. **Recalibrate:**
    - Click the Recalibrate button to recalibrate the analysis based on new data. This is useful if you want to reset the analysis parameters for a new session.
5. **FPS Display:**

- View the FPS (Frames Per Second) display to see the current frames per second of the video playback. This helps in monitoring the performance of the system.

6. **Analysis Results:**
    - The Analysis results section displays analysis results, including:
        - **BPM (Heart Rate):** Shows changes in heart rate detected through facial color analysis.
        - **Lip Compression:** Indicates instances of lip compression, a potential sign of deception.
        - **Gaze Changes:** Tracks changes in gaze direction, which can be an indicator of deceptive behavior.
7. **Exit:**
    - Click the Exit button to return to the main screen.

8. Troubleshooting

1. **Webcam Issues:**
    - Ensure your webcam is properly connected and not being used by another application.
    - Restart the application if the webcam feed does not appear.
2. **Video Playback Issues:**
    - Ensure the video file format is supported by the application. Commonly supported formats include MP4, AVI, and MKV.
    - Check your system's performance if the video playback is lagging.
3. **Analysis Accuracy:**
    - Ensure good lighting conditions for the webcam input to improve the accuracy of facial analysis.
    - Make sure the subject's face is clearly visible and unobstructed during video recording.