SOFTWARE DESIGN DOCUMENT

GEN1

See Beyond the Fake, Trust What's Real!

GROUP MEMBERS

Muhammad Arsal - 2021355 | Haris Anjum - 2021205 | Hassan Ashfaq - 2021221

Supervisor - Engr. **Ahsan Shah**

Co-Supervisor - Dr. **Ali Imran Sandhu**

Revision History and Document Approval

Revision History:

Revision History	Date	Comments
1.00	9-10-2024	Presentation Layer not elaborated.
2.00	10-10-2024	Cover Page modified.

Document Approval:

The following document has been accepted and approved by the following:

Signature	Date	Name

Contents

List of Figures

List of Tables

1 Introduction

In recent years, the proliferation of AI-generated content, particularly deepfake videos, has raised significant concerns regarding misinformation, privacy, and security. These videos leverage advanced machine learning techniques to create highly realistic yet manipulated visual content, posing challenges across various domains including politics, social media, and law enforcement. The potential for misuse is vast, with deepfakes being employed in political smear campaigns, identity theft, and the dissemination of false information. As these technologies continue to evolve, the ability to differentiate between authentic and manipulated content becomes increasingly difficult. The primary objective of this Software Requirements Specification (SRS) document is to outline the development of a robust detection system specifically designed to identify AI-generated and deepfake videos. This system will utilize a combination of spatial and temporal analysis techniques to detect subtle anomalies that traditional detection methods often overlook. By focusing on both the visual inconsistencies within individual frames and the unnatural motion patterns across frames, our solution aims to provide a comprehensive tool for stakeholders who require reliable verification of video authenticity. This document will detail the purpose of the project, the scope of the product, and its intended functionalities. It will also address the user characteristics and constraints associated with its implementation. The ultimate goal is to equip cybersecurity professionals, digital forensics experts, content moderators, and law enforcement agencies with an effective means to combat the threats posed by AI-generated videos. Through detailed reporting mechanisms and real-time analysis capabilities, this detection system seeks not only to flag suspicious content but also to enhance the overall integrity of digital media in an era where trust is paramount.

1.1 Purpose

The purpose of this project is to address the emerging and increasingly complex challenge posed by AI-generated and deepfake videos. These videos are used in a variety of harmful ways—ranging from political misinformation campaigns to impersonation and identity theft—thereby affecting individuals, organizations, and governments alike. Current detection methods are lagging behind the technological advancements that make these videos highly realistic and difficult to detect with the naked eye. Our goal is to develop a detection system that accurately identifies AI-generated videos using a combination of spatial (image-level) and temporal (motion-level) analysis. This will allow for the detection of nuanced anomalies within video content that are often missed by traditional methods. The proposed model will be designed to handle the growing complexity and sophistication of deepfake generation, making it an invaluable tool for cybersecurity professionals, digital forensics experts, law enforcement agencies, and social media platforms. The project also aims to generate an end-of-analysis report, offering insights into the flagged videos, allowing for further verification or investigation.

1.2 Product Scope

The scope of this project extends to developing a robust detection tool that targets a wide variety of AI-generated and deepfake videos. The detection system will leverage both spatial and temporal features to flag videos as being potentially manipulated. Spatial features refer to visual inconsistencies like pixel artifacts, unrealistic textures, or color discrepancies that appear within individual frames of a video. Temporal features, on the other hand, focus on the motion patterns between frames, which may expose discrepancies in how objects or people move, revealing inconsistencies in the flow of time or physics-defying movements. In addition to simply flagging suspicious content, the detection system will offer comprehensive reporting functionalities. The final output will include a detailed report that not only flags the video as AI-generated or a deepfake but also explains the anomalies detected within both the spatial and temporal domains. This report will assist stakeholders in verifying the authenticity of video content, especially in environments where information credibility is crucial, such as journalism, legal proceedings, and content moderation.

Name	**Description**
SRS	Software Requirement Specifications
UC	Use case
SQ	Sequence Diagram
CNN	Convolutional Neural Network
GVD	Generated Video Dataset
HTTP	HyperText Markup Language
I2V	Image-to-Video
T2V	Text-to-Video
FR	Functional Requirements
MT-CNN	Multi-task Cascaded Convolutional Networks

Table 1.1: Terms used in this document and their description

2 Overview

2.1 The Overall Description

The detection system we are developing integrates two major detection components: a spatial analysis module and a temporal analysis module. The spatial module analyzes the visual content within individual frames of a video to detect irregularities that can arise from AI generation or manipulation. For example, inconsistencies in lighting, texture patterns, or pixel distribution are red flags for AI-generated content. Meanwhile, the temporal module analyzes the flow of motion across frames, identifying any unnatural movement patterns that could indicate manipulation. For instance, AI-generated videos may exhibit strange object deformations or inconsistent motion trajectories because the underlying algorithms fail to accurately replicate the natural laws of physics in moving scenes. These two analysis streams—spatial and temporal—are combined through a feature-stacking technique, which allows for the integration of both frame-level and motion-level anomalies. This layered approach increases the detection accuracy, making it harder for advanced AI technologies to bypass detection. The results from these analysis streams will be synthesized into a final verdict, with the system categorizing a video as either authentic or AI-manipulated, and providing detailed reasoning in a generated report.

2.2 Problem Statement

The world is experiencing an unprecedented rise in the generation and dissemination of AI-generated videos, with deepfake technologies leading the charge. While these innovations hold potential for beneficial uses in fields such as entertainment, art, and education, they simultaneously open up a gateway for malicious uses. These AI-generated videos, especially deepfakes, are being used to create misleading and manipulative content, posing severe threats to political stability, social coherence, and individual reputations. For instance, deepfake videos can easily be used in political smear campaigns, social media disinformation, and even in legal settings to falsify evidence. Given that video manipulation techniques are becoming increasingly sophisticated, it is harder to distinguish real content from manipulated content. Current detection techniques primarily focus on AI-generated images or minor facial manipulations, often overlooking video-specific features. The dynamic nature of video content—how frames transition over time and how objects or individuals move—introduces an additional layer of complexity that requires robust analysis. Therefore, the need for a system that can accurately detect and flag AI-generated or deepfake videos is critical, especially when it comes to addressing the large-scale social, political, and economic consequences of their misuse. Our solution employs spatial and temporal analysis techniques and advanced feature-stacking mechanisms to detect these videos. The proposed system aims to provide an automated and accurate flagging mechanism to help identify such content before it causes widespread damage, while also generating comprehensive reports for further investigation.

2.3 Product Perspective

The AI video detection model will serve as a crucial tool for various stakeholders across industries, especially those concerned with media authenticity. By leveraging deep learning, our system will focus on identifying unique characteristics specific to AI-generated videos, such as pixel-level anomalies or motion patterns that deviate from physical laws. The system will be particularly valuable in the context of social media platforms, where misinformation can spread rapidly. It will allow content moderators to flag suspicious videos before they go viral, helping to prevent the dissemination of harmful content. Moreover, law enforcement agencies and digital forensics teams will benefit from the model's ability to verify video authenticity during investigations. This tool will also be useful for news organizations, allowing them to verify video submissions, especially in high-stakes political or social situations where fake content could lead to public unrest. Overall, the model will significantly contribute to the fight against misinformation and digital manipulation.

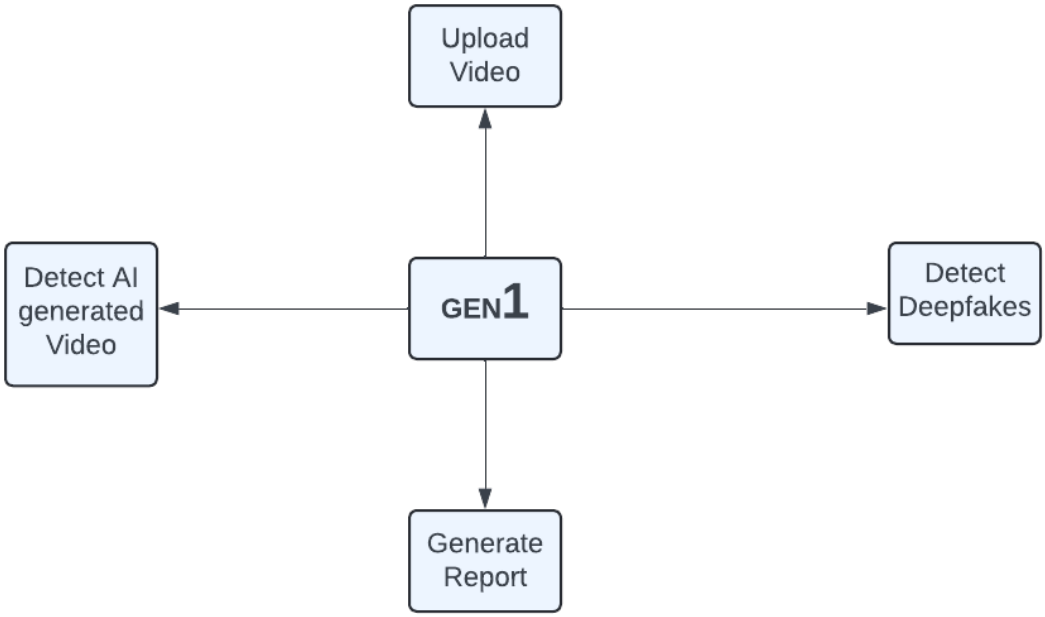

Figure 2.1: Product Perspective Diagram

3 Work Breakdown Structure

3.1 Project Initialization

3.1.1 Requirement Gathering

- Understand functional and non-functional requirements.
- Define detection goals (e.g., accuracy, video formats supported).
- Identify necessary technologies (e.g., MTCNN, Xception, EfficientNet, etc.).

3.1.2 Feasibility Study

- Evaluate hardware and software needs.
- Assess risks (e.g., real-time processing, large video files).

3.1.3 Team Setup

- Assign roles.
- Define responsibilities for each team member.

3.2 System Design

3.2.1 Architecture Design

- Design the Modular Layered Architecture.
- Define module interactions (e.g., User Interface, Video Processing, Detection Systems).

3.2.2 Module Identification

- Break the system into components (e.g., Video Processing Unit, Deepfake Detection System).
- Define data flows between modules.

3.2.3 Technology Stack Selection

- Choose programming languages, frameworks, and libraries.
- Choose cloud infrastructure or local server setup (if needed).

3.3 Front-end Development (Presentation Layer)

3.3.1 User Interface Development

- Design and develop the interface for video uploads.
- Create forms and reports to display detection results.

3.3.2 Integration with Backend

- Ensure smooth communication between UI and video processing systems.
- Implement input validation for video files.

3.4 Back-end Development (Business Logic Layer)

3.4.1 Video Processing Unit

- Implement video frame extraction logic.
- Process frames to prepare for feature extraction.

3.4.2 Deepfake Detection System

- MTCNN Model Implementation: Implement facial detection using MTCNN.
- Xception and EfficientNet Models: Integrate Xception and EfficientNet for feature extraction.
- Feature Selection and Stacking: Implement feature ranking and stacking logic.
- MLP Classifier: Implement classifier for deepfake detection.

3.4.3 AI Video Detection System

- Spatial Detector: Implement spatial analysis logic to detect frame anomalies.
- Temporal Detector: Develop temporal analysis logic for motion inconsistencies.

3.5 Data Access Layer

3.5.1 Feature Extraction Module

- Extract key features from video frames (e.g., facial landmarks, motion vectors).
- Implement feature selection algorithms.

3.5.2 Feature Stacking and Classification

- Stack selected features and pass them through the MLP classifier.

3.6 Report and Feedback Layer

3.6.1 Result Aggregation

- Combine results from spatial, temporal, and deepfake detection.

3.6.2 Report Generation

- Generate user-readable reports indicating AI-generated or deepfake classification.
- Provide confidence scores for the detection.

3.7 Testing and Quality Assurance

3.7.1 Unit Testing

- Test individual components like the video processing unit, detection systems, and classifiers.

3.7.2 Integration Testing

- Test the interaction between the front-end and back-end.
- Test how well the detection systems work with feature extraction and classification.

3.7.3 Performance Testing

- Test the system's scalability and ability to handle large video files.
- Check for latency or delays in generating detection results.

3.7.4 User Acceptance Testing (UAT)

- Present the system to end-users for feedback.
- Make adjustments based on usability and performance feedback.

3.8 Deployment

3.8.1 Prepare Production Environment

- Set up the production environment (servers, cloud, etc.).

3.8.2 Deploy System

- Deploy the system for use by users.

3.8.3 Post-Deployment Testing

- Ensure the deployed system works as expected in production.

3.9 Maintenance and Support

3.9.1 Bug Fixes and Updates

- Continuously monitor the system for bugs.
- Push updates and improvements as needed.

4 Design

4.1 Architectural Design

The architecture chosen for this project is a **Modular Layered Architecture:** This type of architecture is highly suitable for applications involving complex processing tasks, like AI-generated and deepfake detection, because it allows for clear separation of concerns, maintainability, and scalability. Below is a detailed explanation of the architecture and the reasoning behind its structure. The architecture consists of four layers:

- **Presentation Layer**: Interacts with the user and handles video uploads and results display.
- **Business Logic Layer:**Executes the core detection logic, including video processing, feature extraction, and AI detection algorithms.
- **Data Access Layer:** Manages the extraction, combination, and storage of video features.
- **Report and Feedback Layer:** Generates and presents detection reports.

4.2 Why Modular Layered Architecture?

- **Scalability**: Each layer and its corresponding modules can be scaled independently. For example, if more computational power is needed for deepfake detection, the business logic layer can be scaled without altering the user interface or data access layer.
- **Maintainability**: Since different layers handle distinct tasks, it's easier to maintain and update specific components (e.g., upgrading deepfake detection models without affecting the entire system).
- **Separation of Concerns:** The architecture clearly separates video input/output, processing logic, and data handling, making it easier for developers to focus on one aspect at a time. This also leads to fewer conflicts between different system components.
- **Extensibility:** New detection algorithms or feature extraction methods can be added without impacting other parts of the system. This is critical because AI technologies evolve rapidly, and new models (e.g., for detecting AI-generated content) can be integrated as needed.
- **Re-usability**: Components like feature extraction, video frame handling, or report generation can be reused across different projects or extended to handle additional types of video manipulation detection.

4.3 Detailed Breakdown of Modules (Module Identification)

4.3.1 Presentation Layer (User Interface Module)

- **Description:** The User Interface module allows users to upload videos and provides them with the detection results. The UI will also allow user a detailed report generated from the detection process.
- Key Responsibilities:
  - Video upload functionality.
  - Display of detection results and reports.
  - Interaction with the video processing pipeline by sending video input and receiving results.
- **Why this Module?**: This module ensures a user-friendly way for users to interact with the system, submit videos for analysis, and view the detection outcome.

4.3.2 Business Logic Layer (Core Processing and Detection Modules)

The core detection happens in this layer, where videos are analyzed and features are extracted for AI and deepfake detection.

Video Processing Unit

Description: Preprocesses the video for frame extraction and prepares it for further analysis.

Key Responsibilities:

- Extracts frames from the video based on its resolution and frame rate.
- Ensures each frame is ready for spatial and temporal analysis.

Deepfake Detection System

Description: Contains models to detect deepfakes using facial features extracted from video frames.

Key Responsibilities:

- Uses models like Xception and EfficientNet-B7 to analyze frames and detect tampering.
- Runs facial detection, feature extraction, and classification processes.

AI Video Detection System

Description: Detects AI-generated content using spatial and temporal anomaly detection methods.

Key Responsibilities:

- **Spatial Detector**: Analyzes static frame content to identify irregularities.
- **Temporal Detector**: Analyzes motion anomalies between frames to detect AI-generated behaviors.

MT-CNN Feature Stacking

Description: Stacks and ranks extracted features to enhance the detection performance.
Key Responsibilities:

- Feature extraction from different models (spatial, temporal, deepfake).
- Combining and ranking features for use in the final detection classification.

Why this Layer?: This layer handles the essential logic for detecting AI-generated and deepfake videos, providing the core functionality of the system.

4.3.3 Data Access Layer (Feature Management and Storage Modules)

Feature Extraction and Selection Module

Description: Extracts features from each frame (e.g., facial landmarks, motion vectors) and selects the most relevant features for the classification process.
Key Responsibilities:

- Extract features using models such as MTCNN for facial recognition and deep learning models for other aspects.
- Perform feature selection to reduce noise and improve detection accuracy.

Feature Stacking and MLP Classification Module

Description: After feature extraction, this module stacks and processes the selected features using an MLP (Multi-Layer Perceptron) classifier to determine if the video is a deepfake or AI-generated.
Key Responsibilities:

- Stack relevant features into a unified set.
- Classify videos as AI-generated or deepfake with a confidence score.

Why this Layer?: It allows efficient extraction, combination, and classification of video features, which is critical for accurate detection results.

4.3.4 Report and Feedback Layer (Report Aggregation Module)

Result Aggregation and Report Generation Module

Description: Aggregates the results from the detection systems and generates a comprehensive report for the user.
Key Responsibilities:

- Combine results from the spatial, temporal, and deepfake detectors.
- Generate and present the final report, including a confidence score for whether the video is AI-generated or a deepfake.

Why this Layer?: This layer ties together all results and provides the final user-facing output. Without a clear report, the system's utility would be diminished.

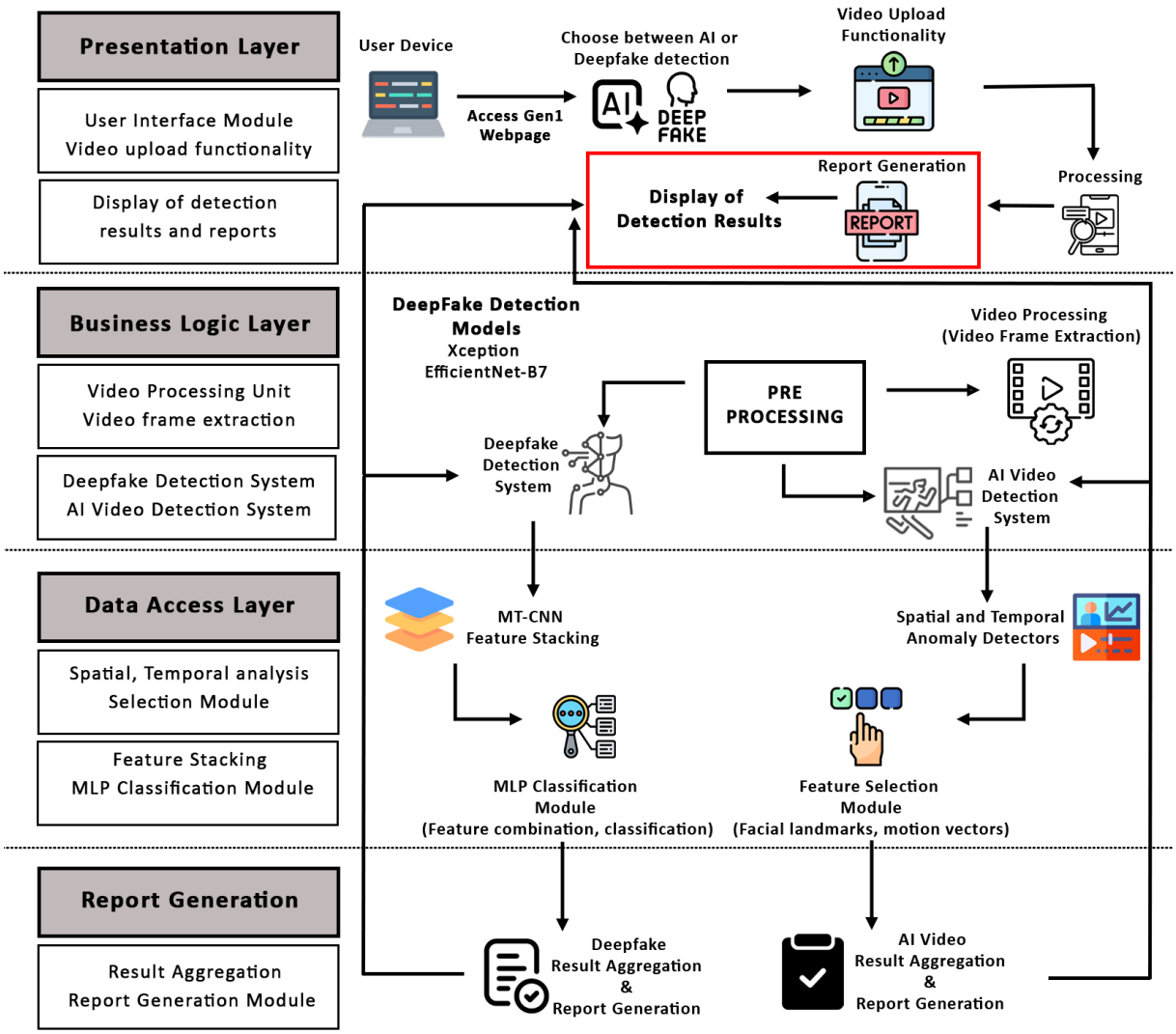

Figure 4.1: Architecture Diagram

5 4+1 Architecture View Model

5.1 Use Case View

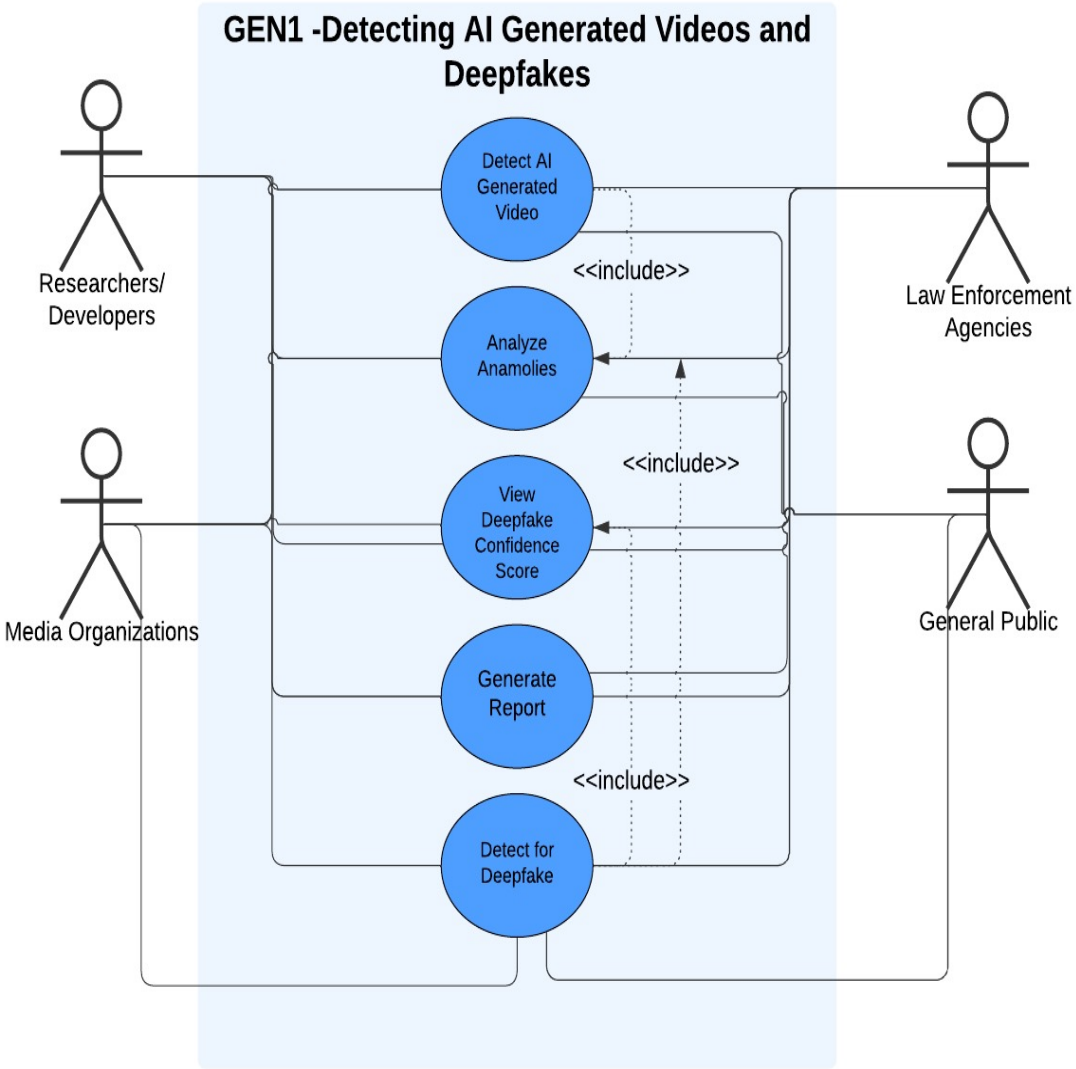

Figure 5.1: Use Case Diagram

5.2 Logical View

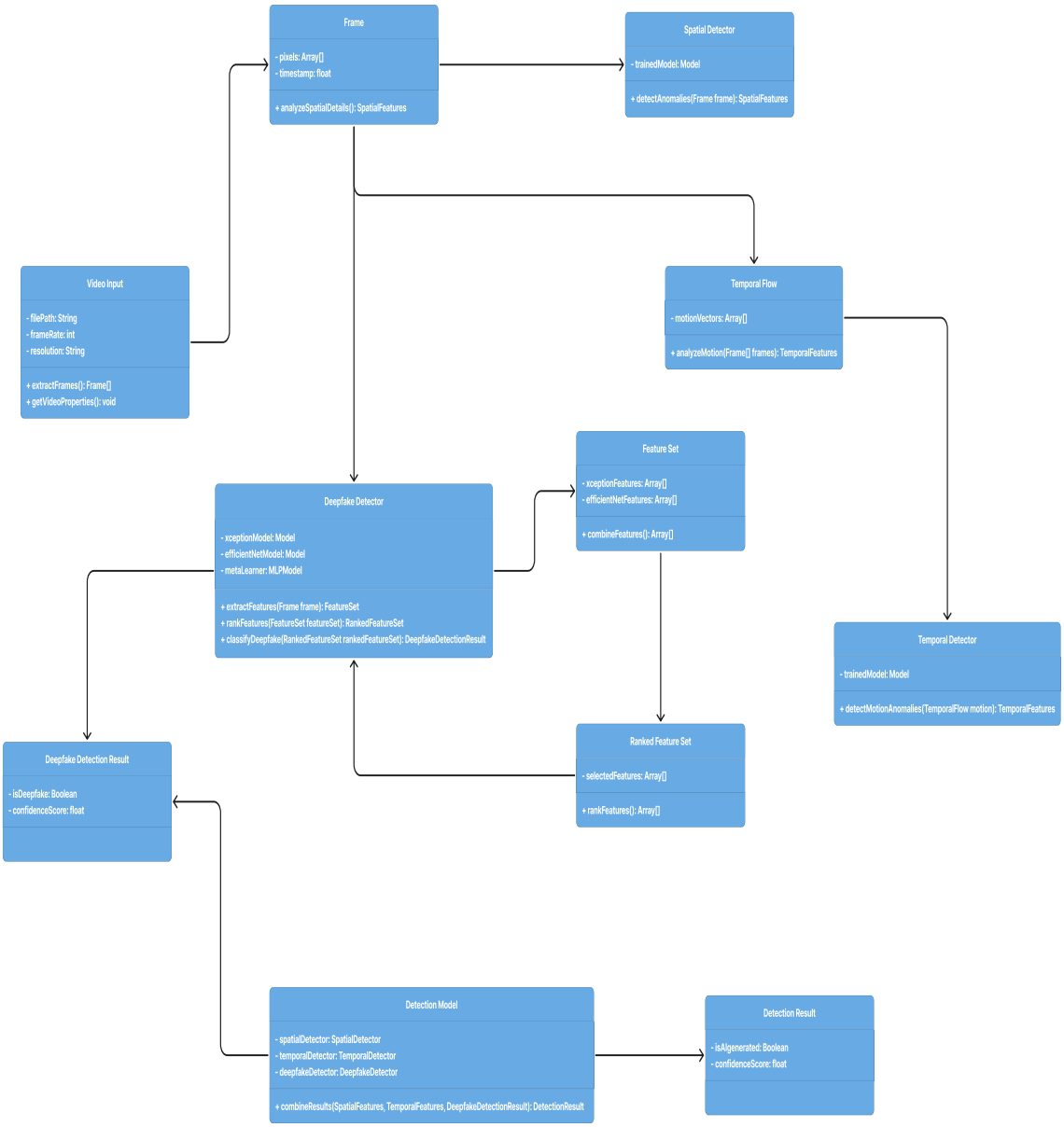

Figure 5.2: Class Diagram

The class diagram for the AI-generated video and deepfake detection project outlines several core components:

- Video Input: This class captures video properties such as file path, frame rate, and resolution. It includes methods to extract frames from the video and retrieve its properties.

- Frame: Each video frame consists of pixel data and a timestamp. This class has a method to analyze spatial details, providing spatial features for further analysis.
- Spatial Detector: A trained model is used to detect spatial anomalies within individual frames. It identifies inconsistencies in spatial features.
- Temporal Flow: This class tracks the motion vectors between frames and analyzes temporal characteristics.
- Temporal Detector: It uses a trained model to detect anomalies in motion vectors, identifying irregularities in the temporal flow of the video.
- Deepfake Detector: This component uses multiple models (Xception, EfficientNet, and a Meta-Learner MLP model) to extract and rank features from video frames. It then classifies whether the video is a deepfake.
- Feature Set and Ranked Feature Set: These classes handle the combination and ranking of features extracted from frames using different models to provide a unified set of data.
- Detection Model: This model integrates results from the spatial, temporal, and deepfake detectors, combining their outputs to generate an overall detection result.
- Detection Results: This class captures the final detection results, indicating whether a video is AI-generated or a deepfake, along with a confidence score.

5.3 Development View

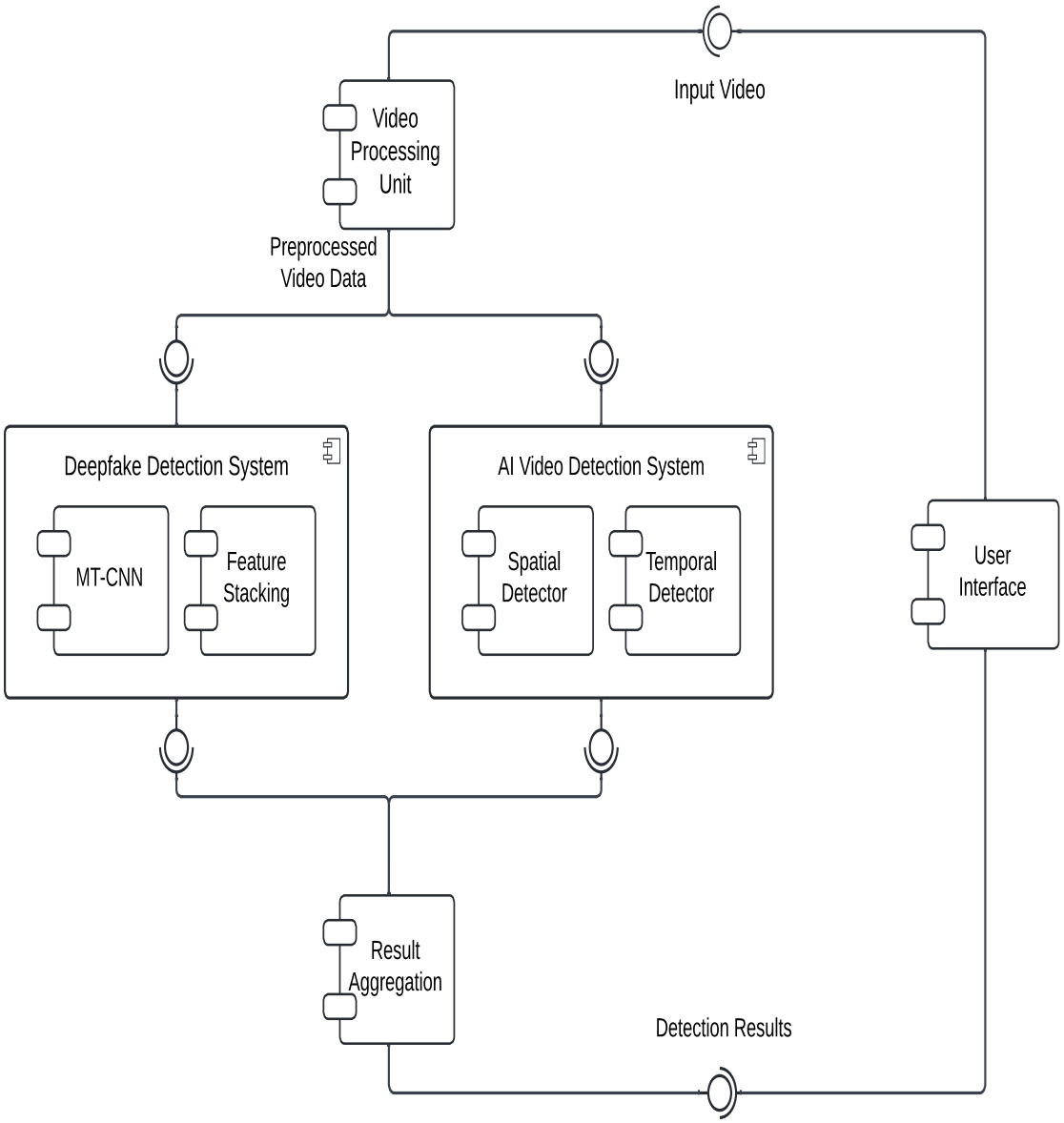

Figure 5.3: Component Diagram

The component diagram for the AI-generated video and deepfake detection project outlines a system that processes input videos through a series of detection stages. The User Interface enables video submission and displays results, while the Video Processing Unit handles video preprocessing. The system leverages MT-CNN Feature Stacking to combine spatial, temporal, and deepfake-related features, feeding into both the Deepfake Detection System, which uses spatial and temporal detectors, and the AI Video Detection System

for AI-generated content identification. Finally, the Result Aggregation component consolidates detection outcomes to provide a comprehensive report to the user.

5.4 Process View

AI Content Sequence Diagram

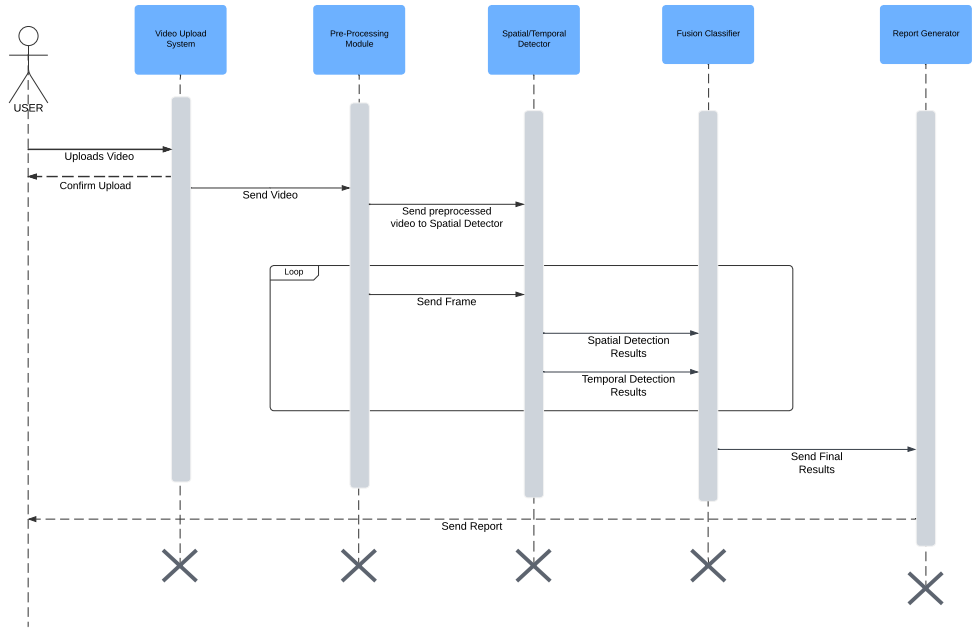

Figure 5.4: Sequence Diagram

The sequence diagram illustrates the process of detecting AI-generated and deepfake videos. It begins with a user uploading a video, which is confirmed by the system. The video is first sent to a pre-processing module, which prepares the video for analysis. The pre-processed video is then sent to both spatial and temporal detectors. The results from these detectors are processed and passed on to a fusion classifier. After the final detection results are compiled, they are sent to a report generator, which creates and sends a detailed report back to the user.

Deepfake Sequence Diagram

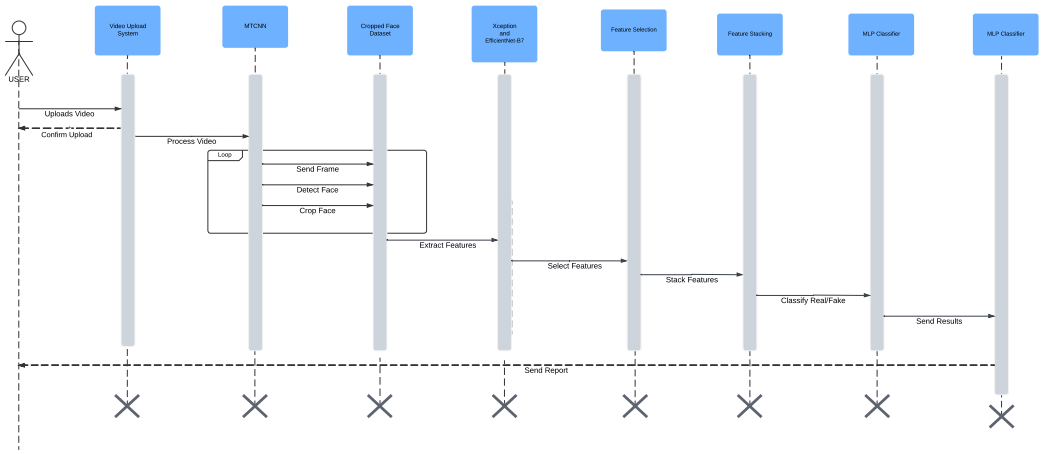

Figure 5.5: Deepfake Sequence Diagram

The sequence diagram for the deepfake detection process illustrates the steps involved from video upload to final classification. The User uploads a video via the Video Upload System, which triggers the processing of the video. Faces are detected and cropped using MTCNN, followed by feature extraction using Xception and EfficientNet-B7 models. These extracted features are then passed through a Feature Selection and Feature Stacking process, before being classified as real or fake using an MLP Classifier. The system then sends the detection results and generates a report for the user.

5.5 Physical View

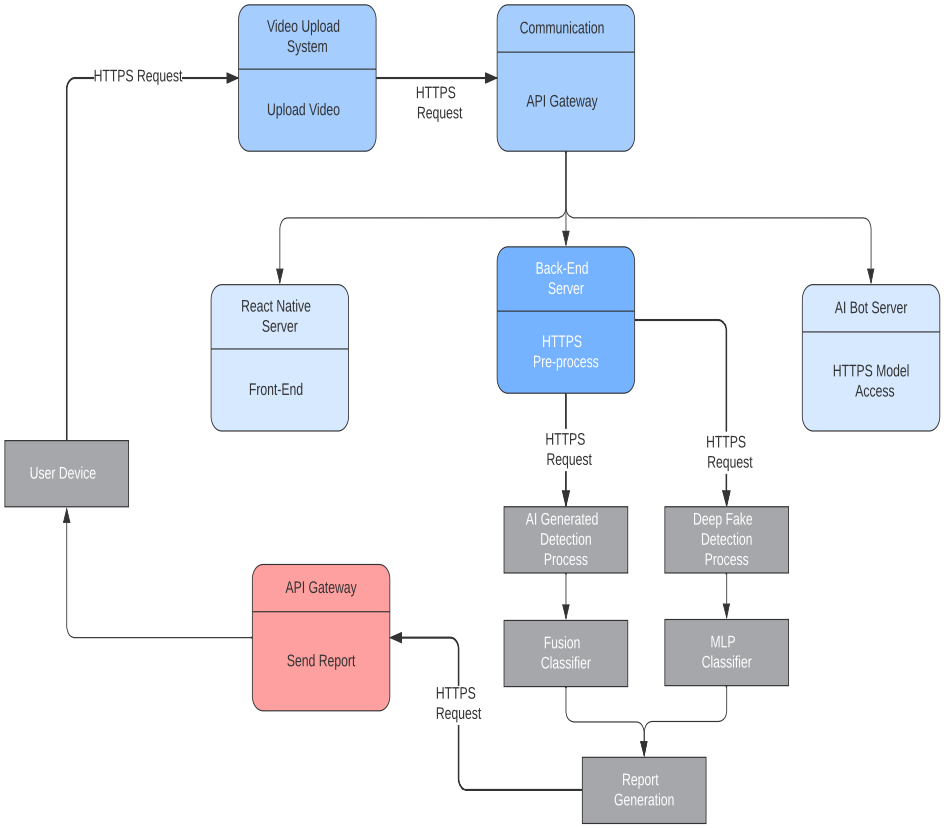

Figure 5.6: Physical View Diagram

The project involves detecting AI-generated and deepfake videos through a structured system. A user uploads a video via a front-end interface connected to a back-end server. The system sends the video for processing, where it passes through two detection processes: AI-generated detection and deepfake detection. Both processes use machine learning models (including MLP and Fusion classifiers) to classify the video content. Once analyzed, the results are communicated via HTTPS through an API Gateway, and a report is generated and sent back to the user, summarizing the detection outcome.