MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

for

Deep Fake Detection - A Comprehensive Approach using AI

Version 1.0

Prepared by

S. No	Roll Number	Student Name
1.	21N31A7221	GADADASU GANESH
2.	21N31A7211	CHEEDELLA MANIKANTA
3.	21N31A7229	KATLA SAI KUMAR

Supervisor:	MR. K. MAHESH BABU
Designation:	Assistant Professor
Department:	Computational Intelligence
Batch ID:	21CIMP2E06
Date:	
Supervisor Sign. & Date	

Department of Computational Intelligence

Title of the Project: Deep Fake Detection - A comprehensive approach

Content

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	G. Ganesh CH. Manikanta K.Sai Kumar	Deep Fake Detection -A comprehensive approach using AI	24/03/25

1 Introduction

Deepfake technology has emerged as a powerful yet controversial application of artificial intelligence, enabling the creation of highly realistic but manipulated visual content. Leveraging advancements in deep learning and generative adversarial networks (GANs), deepfakes can convincingly alter images and videos to misrepresent reality. This project, "Deepfake Detection and Prevention," addresses these challenges by developing advanced AI-based solutions to identify and mitigate the impact of manipulated images. By utilizing machine learning algorithms and neural networks, the project aims to build reliable detection systems capable of distinguishing authentic content from deepfakes with high accuracy.

1.1 Document Purpose

This document outlines the software requirements for the **Deepfake Detection System**, a project that aims to identify and mitigate deepfake content using AI-driven detection mechanisms. The system will leverage deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to analyze images and videos, identifying manipulations with high accuracy. Additionally, it will incorporate adversarial training to improve robustness against evolving deepfake generation techniques.

The purpose of this document is to provide a clear and comprehensive understanding of the system's functionalities, architecture, and development requirements. It will serve as a blueprint for stakeholders, including developers, researchers, and policymakers, ensuring alignment with project goals and compliance with industry standards.

1.2 Project/Product Scope

The Deepfake Detection System is designed to identify and analyze manipulated images and videos using advanced AI-driven techniques. It enhances the accuracy of detection by leveraging CNNs and RNNs to detect inconsistencies in media files. The system offers a scalable, real-time solution applicable to social media monitoring, law enforcement, and cybersecurity. It ensures rapid processing and automated deepfake detection with minimal human intervention. The system will also integrate with existing security frameworks, providing enhanced detection capabilities for digital forensic analysis. Designed with user-friendliness in mind, it will include an intuitive interface suitable for both technical and non-technical users. Additionally, the system will continuously learn from evolving deepfake techniques to improve its detection efficiency.

The Deepfake Detection System is designed to:

- Detect manipulated images and videos using deep learning models.
- Improve the accuracy of deepfake detection by leveraging CNNs and RNNs.
- Provide a scalable and real-time solution for social media and security applications

1.3 Existing System

Current deepfake detection systems use traditional forensic techniques and standalone AI models. However, they face several challenges, including:

- **Limited Scalability:** Many existing systems struggle to process large volumes of data in real time.
- **Low Robustness:** Some models fail to detect sophisticated deepfake techniques due to a lack of adaptability.
- **High Computational Costs:** Training and running deep learning models require significant computational resources.
- **Inconsistent Accuracy:** Many detection systems perform well on specific datasets but fail to generalize to new or unseen deepfake methods.
- **Lack of Real-Time Processing:** Many deepfake detection methods are too slow for real-time applications, making them impractical for live content monitoring.

1.4 Problems with Existing System

- **Adversarial Attacks**: Deepfake models are evolving rapidly, making detection harder.
- **Generalization Issues:** Models trained on one dataset may not perform well on new deepfake techniques.
- **Resource-Intensive:** High computational power is required for model training and real-time processing.
- **Privacy Concerns:** Need for secure data handling to prevent misuse.
- **Dataset Bias:** Many existing detection models are trained on biased datasets, leading to reduced effectiveness in detecting diverse deepfake manipulations.
- **Lack of Explainability:** AI models often operate as black boxes, making it difficult to interpret why a particular media file was classified as fake or real.
- **Real-Time Constraints:** Many deepfake detection models are too slow to be effectively deployed in live monitoring systems.
- **Generalization Issues**: Models trained on one dataset may not perform well on new deepfake techniques.
- **Resource-Intensive:** High computational power is required for model training and real-time processing.
- **Privacy Concerns:** Need for secure data handling to prevent misuse

1.5 Proposed System

The **AI-driven Deepfake Detection System** is designed to automate and enhance deepfake identification using state-of-the-art machine learning and deep learning models. The system follows a structured approach to analyze digital content effectively, ensuring accurate, scalable, and real-time detection of manipulated media. Below is a detailed breakdown of the system's key steps:

1. Data Input and Preprocessing

Users can submit images and videos for analysis. The system supports multiple input sources, including:

- Uploaded files (JPEG, PNG, MP4, AVI, etc.).
- Social media content via API integration.
- Live video streams for real-time monitoring.

Preprocessing includes:

- Frame Extraction: Videos are split into frames for detailed analysis.
- Noise Reduction: Enhancing image/video clarity before deepfake evaluation.
- Face Detection: Identifying faces using OpenCV and DLIB.

2. Feature Extraction and Analysis

The system employs AI-driven models to extract and analyze key features:

- CNN-based Image Processing: Detects inconsistencies in lighting, facial structure, and texture.
- RNN-based Motion Analysis: Identifies unnatural movements in videos.
- Deepfake Pattern Detection: Trained on large datasets of real vs. fake content.

3. Deepfake Classification

Once features are extracted, the system classifies media as real or fake using:

- Pre-trained AI Models: Leveraging deep learning models such as XceptionNet and EfficientNet.
- Adversarial Training: Strengthening detection models against adversarial attacks.
- Confidence Scoring: Assigning probability scores to detected manipulations.

4. Report Generation and Alerts

The system provides detailed reports on detection results, including:

- Confidence Levels: Probability of content being deepfake.
- Manipulation Markers: Highlighting detected anomalies.
- Automated Alerts: Notifications to stakeholders when deepfake content is identified.

5. Continuous Learning and Updates

To adapt to evolving deepfake techniques, the system integrates:

- Self-Improving AI Models: Learning from newly detected deepfake patterns.
- Crowdsourced Verification: Incorporating user feedback for better accuracy.
- Periodic Model Retraining: Updating detection algorithms with new data.

6. Deployment and Scalability

The system is designed for large-scale implementation, supporting:

- Cloud-Based Processing: Utilizing AWS, GCP, or Azure for real-time scalability.
- Edge AI Integration: Running lightweight models on mobile and embedded devices.
- Multi-Platform Support: Web-based, desktop, and mobile application compatibility.

By implementing these advanced AI-driven techniques, the Deepfake Detection System ensures accurate, scalable, and efficient deepfake identification across multiple industries, from cybersecurity to media integrity protection.

1.6Advantages of Proposed Systems

- **High Accuracy:** Implements deep learning models for precise deepfake detection.

- **Real-Time Processing:** Optimized algorithms enable fast analysis of media content.

- **Scalability:** Designed to handle large-scale detection tasks across various platforms.

- **Robust Security:** Ensures secure data handling and compliance with global privacy laws.

- **Automated Detection:** Reduces human intervention with AI-powered workflows.

- **Continuous Learning:** Adapts to evolving deepfake generation techniques for enhanced detection.

2 Overall Description

2.1 Feasibility Study

A comprehensive feasibility study was conducted to assess the viability of the Deepfake Detection System. This evaluation covered four key areas: **technical, economic, operational, and scheduling feasibility.**

- **Technical Feasibility**

Uses established AI frameworks such as TensorFlow, PyTorch, and OpenCV.

Integrates seamlessly with cloud-based solutions for scalable deepfake detection.

Supports real-time processing for live content analysis.

- **Economic Feasibility**

Leverages open-source AI tools to minimize development costs.

Monetization opportunities include subscription services, API licensing, and enterprise solutions.

Reduces the cost of manual video authentication, providing a high return on investment (ROI).

- **Operational Feasibility**

Automates deepfake detection, reducing human effort in forensic analysis.

Designed with a user-friendly interface for accessibility across different sectors.

Can be easily integrated with social media platforms and security systems..

- **Scheduling Feasibility**

A structured development roadmap is established, covering model training, testing, and deployment.

Available AI tools and cloud resources support timely project completion.

Includes risk management strategies for model inaccuracies and technology shifts.

The feasibility study confirms that the Deepfake Detection System is **technically, economically, and operationally viable**, offering a scalable and efficient solution to combat deepfake threats.

2.2 Product Functionality

The **Deepfake Detection System** offers a comprehensive set of features to streamline the process of detecting and analyzing deepfake content:

- **Automated Media Processing:** Analyzes images and videos using deep learning models to identify manipulated content.
- **Feature Extraction & Analysis:** Utilizes CNNs and RNNs to detect facial inconsistencies, unnatural movements, and texture anomalies.
- **Deepfake Classification:** Employs AI-powered classification models such as XceptionNet and EfficientNet to determine whether content is real or fake.
- **Confidence Scoring:** Assigns probability scores to detected deepfakes, providing insights into the likelihood of manipulation.
- **Real-Time Detection:** Supports live content monitoring by processing video streams in real-time to identify deepfake elements instantly.
- **Automated Report Generation:** Generates detailed reports highlighting anomalies and potential deepfake markers for forensic analysis.
- **Multi-Platform Integration:** Compatible with social media platforms, security agencies, and forensic investigation systems.

The system is designed to be **efficient, secure, and adaptable** to evolving deepfake threats, making it an essential tool in combating misinformation and digital fraud.

2.3Design and Implementation Constraints

The development of the **Deepfake Detection System A comprehensive approach** must adhere to several key constraints:

1. **Hardware & Performance Limitations**
    - The system must support high-performance deepfake detection while ensuring accessibility for users with limited hardware resources.
    - Cloud-based processing will be leveraged for scalability and to reduce computational load on local devices.

2 **Software Platform**

- The project will be developed using Python, integrating AI libraries such as TensorFlow, PyTorch, and OpenCV.
- The system must follow best practices in AI model deployment and optimization to enhance efficiency.

3. Modular Design

- The system must follow a modular architecture, ensuring easy integration of new AI models and features.
- Individual components (image preprocessing, feature extraction, classification, and reporting) should be interchangeable for flexibility and scalability.

4. User Interface (UI) and User Experience (UX)

- The UI must be intuitive and accessible, catering to users without technical expertise in AI or deepfake detection.
- Features such as drag-and-drop media upload, real-time detection previews, and interactive report visualization must be integrated

5. Data Management

- Efficient storage and retrieval mechanisms must be in place for storing detection logs and metadata.
- The system must provide interactive data visualization for forensic and analytical insights.

6. Security Considerations

- User data privacy must be ensured when processing and storing images and videos.
- The system should implement access control measures to prevent unauthorized use and content tampering.

7. Compliance with Industry Standards

- The system must comply with data protection regulations such as GDPR and CCPA.
- AI-driven detection results should align with ethical AI guidelines to avoid bias and misinformation.

By adhering to these design and implementation constraints, the **Deepfake Detection System** ensures efficiency, security, and adaptability in detecting and preventing deepfake content across multiple applications.

2.4 Assumptions and Dependencies

The following assumptions and dependencies are critical to the successful development and implementation of the **Deepfake Detection System**:

Assumptions

- **Availability of AI Models:** The system assumes continued support and updates from AI frameworks such as TensorFlow, PyTorch, and OpenCV.
- **Cloud Processing Capabilities:** Users are expected to have access to cloud-based resources for high-performance deepfake detection.
- **User Interest in Deepfake Detection:** The demand for detecting and mitigating deepfake content is expected to grow across industries.
- **Sufficient Development Resources:** The project is expected to be completed with available AI expertise and computational infrastructure.

Dependencies

- **AI Framework Stability:** The system relies on pre-trained AI models, and any major changes or deprecations could impact performance.
- **Third-Party APIs and Tools:** Dependencies include OpenCV for image processing, cloud services for storage, and social media APIs for integration.
- **Regulatory Compliance:** The system must adhere to privacy laws and ethical AI guidelines when handling user-generated content.

Conclusion

By addressing these assumptions and dependencies, the **Deepfake Detection System** ensures a scalable, secure, and reliable approach to deepfake identification and prevention.

3 Functional Requirements

3.1 Software Requirement Specifications

The **Deepfake Detection System** relies on a robust software infrastructure for efficient deepfake analysis and detection:

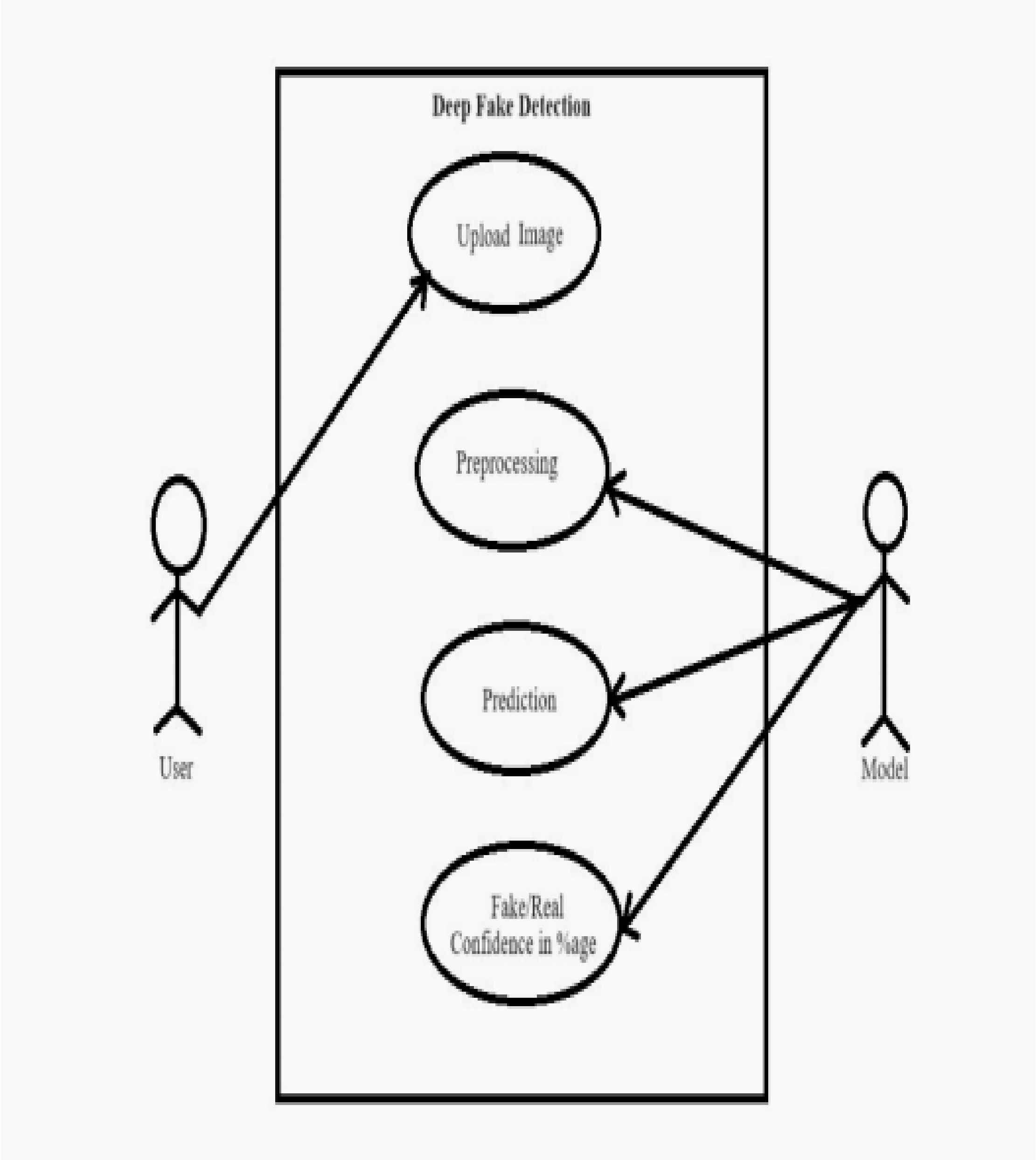
- **Operating system :** Windows 10 or above
- **Programming language :** Python 3.8 or later
- **ML and DLFrameworks :** Tensorflow, keras
- **Image processing :** OpenCV, Numpy
- **Web application framework :** streamlit
- **Pretrained model :** MobileNet

3.2 Hardware Requirements Specifications

The Deep Fake Detection require:

- **Processor :** intel i5 or i7
- **RAM :** 8GB
- **GPU :** NVIDIA RTX 3060
- **Storage :** 128GB

3.3 Use Case Model



3.3.1 Use Case #1 (Deepfake Detection – U1)

Author – Team Deepfake AI Detection

Purpose – This Use Case provides a high-level overview of how a user interacts with the platform to analyze and detect manipulated images or videos using AI-based models.

Requirements Traceability:

- **R1**: Media upload and preprocessing
- **R4**: Feature extraction and motion analysis
- **R7**: Deepfake classification using AI models
- **R10**: Report generation with confidence scoring
- **R12**: Integration with real-time monitoring systems

Priority – High

Preconditions – User has uploaded an image or video file for analysis.

Postconditions – The system generates a detailed report indicating whether the media is real or manipulated (deepfake).

Actors – User (Forensic Analyst, Security Officer, Content Moderator)

Extends – N/A

Flow of Events

1. **Basic Flow**
   - **User logs in to the detection platform.**
   - **User uploads a media file (image/video) or inputs a social media link.**
   - **System extracts frames (for videos) and preprocesses content.**
   - **CNN-based models analyze image textures and inconsistencies.**
   - **RNN-based models analyze motion patterns (for videos).**
   - **AI classifier (e.g., XceptionNet) determines authenticity.**
   - **A report with confidence scores and deepfake indicators is generated.**
   - **User views/downloads the analysis report.**

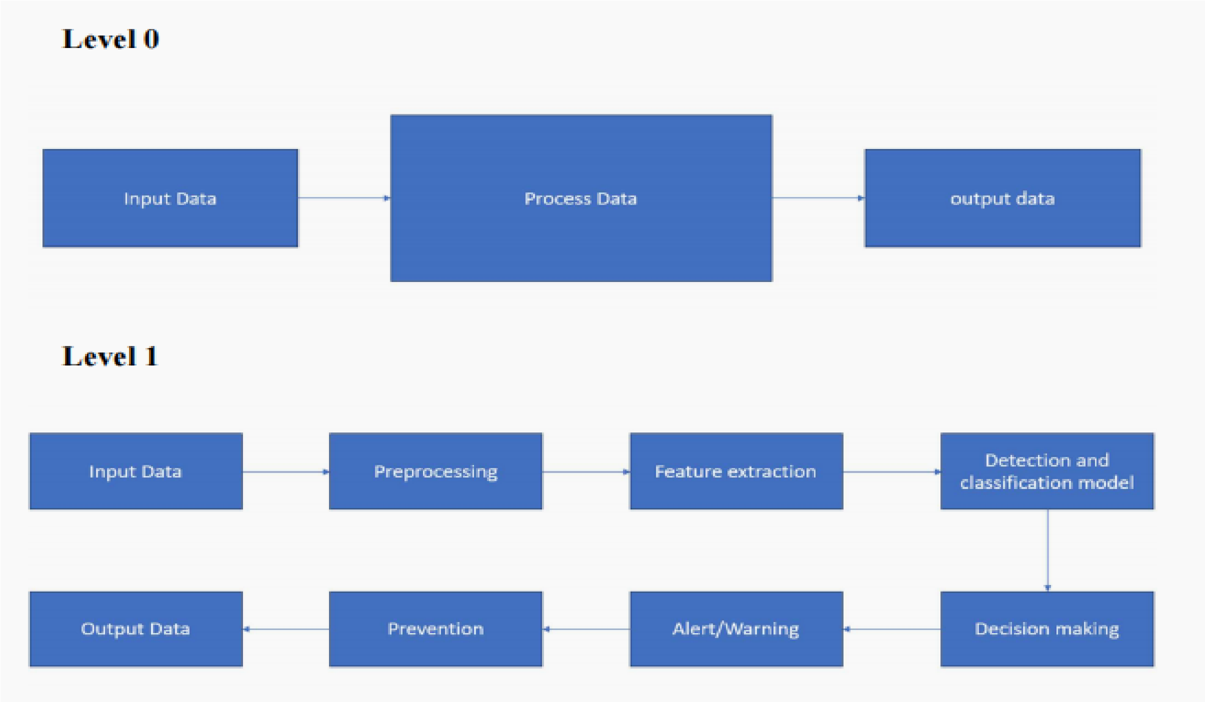
2. **Alternative Flow**
   - **System requests re-upload if the file is corrupted or unsupported.**
   - **User provides manual confirmation if results are inconclusive.**
   - **In live mode, system continues to monitor and detect media anomalies in real time.**

3. **Exceptions**
   - Media file format is not supported.
   - **Network/server issues during file upload or model inference.**
   - AI model fails to return results due to internal error.

4. **Includes**
   - AI-powered feature extraction (CNN, RNN)
   - Preprocessing and face detection
   - Deepfake classification engine
   - **Report generation and alert system**

5. **Notes/Issues**
   - **Future versions may include blockchain-based verification of media origin.**
   - Feature to allow public reporting of suspected deepfakes can be added.

3.3.2 Data Flow Diagram



4 Other Non-functional Requirements

4.1 Performance Requirements

To ensure optimal functionality of the Deepfake Detection System, the following performance benchmarks must be met:

- **P1.** The system must complete image analysis within 5 seconds
- **P2.** Video analysis should be completed within 15 seconds for 1-minute clips
- **P3.** The system must maintain 90% or higher detection accuracy on test datasets
- **P4.** Capable of processing at least 10 media files concurrently in real-time.
- **P5.** Preprocessing and feature extraction should not exceed 2 seconds per frame.

4.2 Safety and Security Requirements

To maintain system integrity and protect user data, the following safety and security requirements are implemented:

S1. All media data must be encrypted during transmission and storage.

S2. Role-based access must be implemented to restrict system control and data visibility.

S3. File uploads should be scanned and sanitized to avoid code injection attacks.

S4. System logs must avoid storing any personal user data.

S5. The system should comply with GDPR and CCPA standards for data privacy and protection.

S6. Access Control: Implement **role-based access control (RBAC)** for managing user permissions.

S7. User Content Privacy:

- Ensure AI-generated content complies with **copyright laws**.
- Provide options for **data anonymization** in text-based input.

S8. No Collection of Personal Data: The platform **does not store** personally identifiable information (PII). Any future user data collection must adhere to **GDPR/CCPA** compliance.

S9. Software Updates & Security Patching:

- Regularly update **AI models and system libraries** to prevent security vulnerabilities.
- Deliver software updates through **secure channels** to prevent tampering.

S10. Error Handling & Logging: Implement robust **error reporting** mechanisms to track failures in **AI model processing, video assembly, and API requests**.

4.3 Software Quality Attributes

The following software quality attributes ensure the reliability, maintainability, and usability of the system:

4.3.1 Usability

Requirement: The interface must be intuitive and easily navigable by non-technical users.

Implementation: Include drag-and-drop file upload, progress bars, tooltips, and simplified menus.

Verification: Usability testing through user feedback sessions and A/B testing.

4.3.2 Maintainability

Requirement: The system should be easy to **update, debug, and extend**.

Implementation: Use modular architecture, version control (Git), and inline documentation.

Verification: Regular code reviews, automated tests, and update logs.

4.3.3 Scalability

- **Requirement**: The system must handle increased workloads and user growth efficiently.
- **Implementation**: Use scalable cloud platforms **(e.g., AWS, GCP) and containerized services (Docker, Kubernetes)**
- **Verification**: Successfully adding **new AI models and features** within a reasonable timeframe will verify adaptability.

4.3.4 Reliability

- **Requirement**: The system should operate without crashes or data loss and have 99%+ uptime.
- **Implementation**: Deploy on redundant infrastructure with exception handling and backup logging.
- **Verification**: Uptime monitoring, stress testing, and error tracking systems

5 Other Non-functional Requirements

This section outlines additional technical, legal, and functional requirements not covered in the previous sections, which are essential for the complete development and deployment of the **Deep Fake Detection and Prevention System – A Comprehensive Approach using AI**. These requirements support the project's goal to empower stakeholders in combating digital misinformation and safeguarding the integrity of visual media.

5.1 Database Requirements (If Applicable)

- **R1. Detection Results Storage:** The system should securely store logs of media scans, including confidence scores, timestamps, and classification results (real/fake) using a scalable database like PostgreSQL or MongoDB.
- **R2. Model Training Logs:** Maintain records of model training, including datasets used, parameters, and validation accuracy, for future audits and improvement.
- **R3. Data Security & Integrity:** All stored content (e.g., logs, detection results) must be encrypted and protected against unauthorized access or modification, adhering to security standards like AES and SHA..

5.2 Internationalization Requirements (If Applicable)

- **R4. Multilingual User Interface:**The platform should provide a multilingual UI to support international users, especially in digital forensics and policy enforcement sectors.
- **R5. Unicode Support:** The system must support UTF-8 encoded text to enable proper handling of multilingual datasets and file names.
- **R6. Cultural Sensitivity:** Detection and prevention algorithms should be evaluated for cultural biases to ensure fairness in analysis across different demographic groups.

5.3 Legal Requirements

- **R7. Privacy and Consent:** Users must consent before their media files are analyzed. The system must clearly state that it does not retain or share content without user permission.
- **R8. Regulatory Compliance:** The platform must comply with data protection laws such as **GDPR**, **CCPA**, and local cyber laws related to media forensics and misinformation.

- **R9. Ethical AI Standards:** Deepfake detection must follow responsible AI practices, including transparency of decisions and the right to contest a false classification.

5.4 Reuse Objectives

- **R10. AI Module Reusability:** Detection modules (e.g., CNN/RNN classifiers, adversarial detectors) should be developed as reusable components for integration into other forensic tools.
- **R11. Preprocessing Pipelines:** Face detection, noise filtering, and frame extraction modules should be designed to be modular and portable for future projects.
- **R12. Community Integration:** Where applicable, tools and findings should be released to the open-source community to encourage further research and refinement..

5.5 Development Environment Requirements

- **R13. Technology Stack: Python 3.8+, TensorFlow, PyTorch, OpenCV, Streamlit (or Flask), and Dlib.**
- **R14. Version Control and Collaboration: Git (GitHub or GitLab) must be used for source control, project tracking, and team collaboration.**
- **R15. Model Lifecycle Management: Include mechanisms for continuous training, evaluation, and replacement of detection models as deepfake techniques evolve.**
- **R16. Cloud Deployment & Scaling:** The system should be deployable on cloud platforms like AWS, GCP, or Azure with optional edge deployment for mobile or low-power devices.

5.6 Documentation Requirements

- **R17. Technical Documentation:** All components (data pipelines, models, APIs) must be well-documented for future developers and contributors.
- **R18. User Guide:** Provide end-users with a comprehensive guide on how to upload files, interpret results, and report false positives or negatives.
- **R19. Research and Audit Reports:** Include clear documentation of datasets, evaluation metrics, and model performance for research transparency and validation.

  R20. Versioning and Release Notes: Each new update should include version history, new features, resolved issues, and changes in model behavior.

6 References

- **Deep Learning for Deepfake Detection** – A survey on deep learning techniques used in identifying manipulated media.

- **Generative Adversarial Networks (GANs): A Comprehensive Survey** – IEEE Transactions on Neural Networks

- **FaceForensics++ Dataset Research** – A benchmark dataset for deepfake detection, highlighting dataset usage and performance metrics

- **Media Integrity and Trust in Digital Platforms** – Research by Oxford Internet Institute on misinformation and detection technologies.

- **Ethical AI for Deepfake Mitigation** – Guidelines by the Partnership on AI and IEEE on responsible detection and content analysis.

SRS DOCUMENT REVIEW

CERTIFICATION

This Software Requirement Specification (SRS) Document is reviewed and certified to proceed for the project development by the Departmental Review Committee (DRC).

Date of SRS Submitted:	
Date of Review:	
Supervisor Comments:	
Supervisor Sign. & Date.	
Coordinator Sign. & Date	
HOD Sign. & Date	
Dept. Stamp	