Acknowledgement

We would like to express our sincere gratitude to all those who supported us throughout the course of this project titled **Deeflyzer: Hybrid Model to Detect Complex Deepfake in Digital Media.**

First and foremost, we would like to thank our project guide, **Dr. Jitendra Musale**, for his valuable guidance, support, and encouragement. His insightful feedback and technical expertise were instrumental in shaping the direction of this project.

We extend our heartfelt thanks to our **Project Coordinator, Dr. Pranjali More**, for her continuous support, timely suggestions, and for ensuring the smooth progress of the project. We are also grateful to the **Computer Engineering Department** of **Anantrao Pawar College of Engineering and Research** for providing the necessary resources and a conducive environment for carrying out this work.

Lastly, we would like to acknowledge the cooperation and support of our peers and team members, whose collaboration and motivation were crucial during the research and development phases. This project has significantly enhanced our understanding of deepfake detection and its critical role in maintaining the authenticity and integrity of digital media.

NAME OF THE STUDENTS

SOHAM VIJAY KOLAPKAR
RIYA GIRISH KSHIRSAGAR
CHARUDATTA SUNIL THAKARE
SHANTANU MANOJ SHINDE

Contents

List of Figures

List of Tables

List of Abbreviations

AI	: Artificial Intelligence
API	: Application Programming Interface
ASVspoof	: Automatic Speaker Verification Spoofing and Countermeasures
CNN	: Convolutional Neural Network
DFDC	: DeepFake Detection Challenge
DL	: Deep Learning
ER	: Entity-Relationship
GAN	: Generative Adversarial Network
GUI	: Graphical User Interface
LSTM	: Long Short-Term Memory
ML	: Machine Learning
MIL	: Multiple Instance Learning
MTCNN	: Multi-task Cascaded Convolutional Networks
NPR	: Neighboring Pixel Relationships
RNN	: Recurrent Neural Network
SDLC	: Software Development Life Cycle
SVM	: Support Vector Machine
UI	: User Interface
UML	: Unified Modeling Language

Abstract

The increasing sophistication of deepfake technology poses serious risks to media integrity, security, and public trust. This project presents Deeflyzer, a hybrid deepfake detection system that leverages both visual and auditory modalities to accurately identify manipulated content. The proposed framework integrates a convolutional neural network (InceptionV3) with a recurrent neural network (LSTM) for video analysis, enabling the extraction of both spatial and temporal features from video frames. Additionally, it incorporates Wav2Vec 2.0, a transformer-based speech model, to detect audio-based deepfakes by analyzing vocal anomalies and synthetic speech patterns. For video input, key frames are extracted, facial regions are localized using MTCNN, and deep visual features are learned for classification. For audio, the input is preprocessed, converted to waveform format if necessary, and then evaluated using the pretrained audio model. The system delivers high accuracy in classifying content as real or fake and provides localized evidence by displaying manipulated facial frames. The hybrid approach ensures robustness across various forms of deepfake attacks, offering a comprehensive solution for real-world applications. A web-based interface built with NextJS and FastAPI for backend integrated with a user-friendly front end allows seamless media upload, processing, and result visualization. Experimental results on benchmark datasets demonstrate the effectiveness of Deeflyzer, achieving over 96 accuracy for both video and audio classification tasks. The proposed system contributes to the ongoing efforts in digital forensics and media authentication by providing a scalable, interpretable, and accurate detection framework.

Chapter 1

Synopsis

1.1 Project Title

Deeflyzer: Hybrid Model to Detect Complex Deepfake in Digital Media

1.2 Project Option

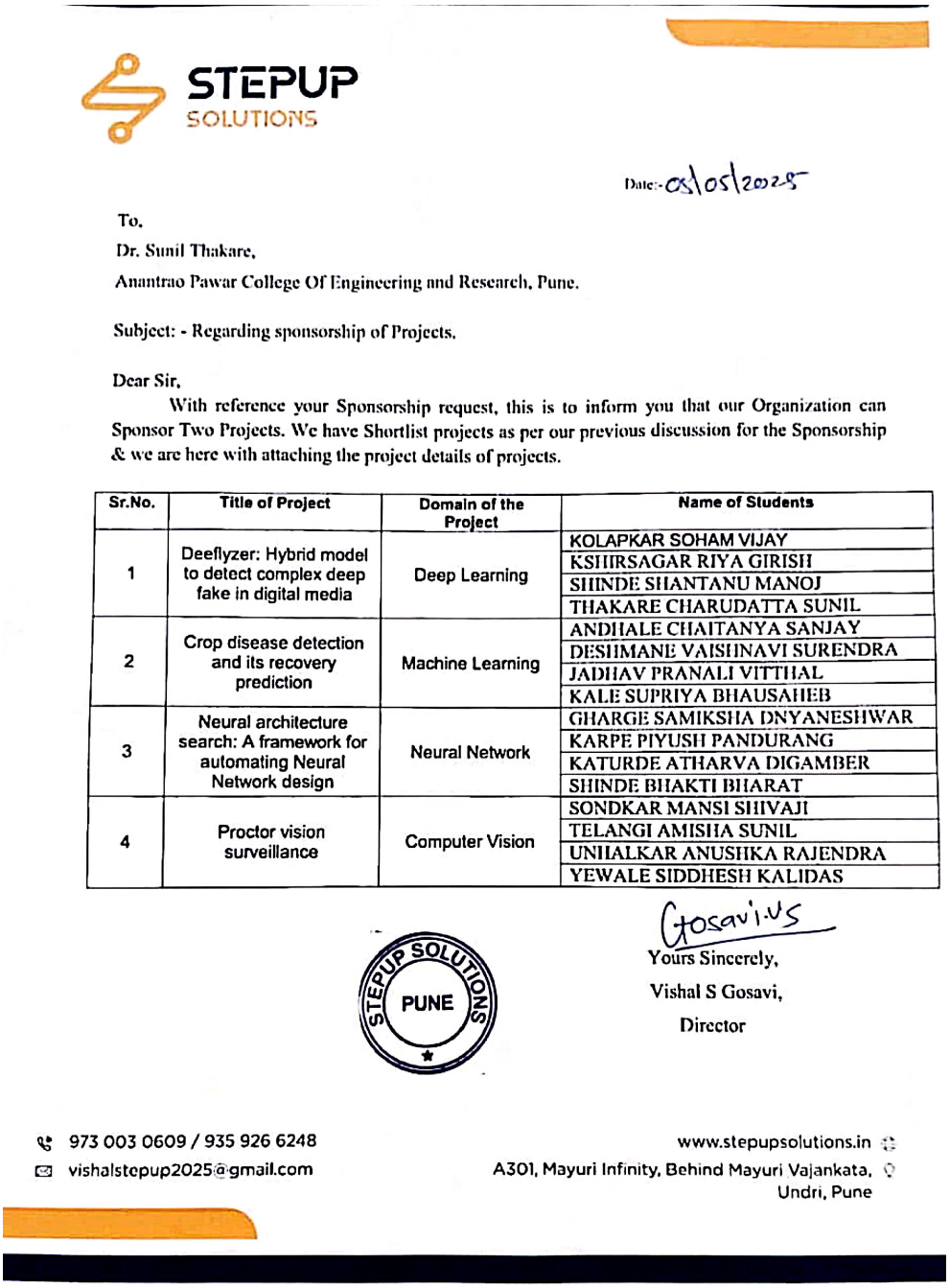
Industry Sponsored

1.3 Internal Guide

Dr. Jitendra C. Musale

1.4 Sponsorship and External Guide

Sponsored by STEPUP SOLUTIONS



STEPUP SOLUTIONS

Date:- 05/05/2025

To,
Dr. Sunil Thakare,
Anantrao Pawar College Of Engineering and Research, Pune.

Subject: - Regarding sponsorship of Projects.

Dear Sir,

With reference your Sponsorship request, this is to inform you that our Organization can Sponsor Two Projects. We have Shortlist projects as per our previous discussion for the Sponsorship & we are here with attaching the project details of projects.

Sr.No.	Title of Project	Domain of the Project	Name of Students
1	Deeflyzer: Hybrid model to detect complex deep fake in digital media	Deep Learning	KOLAPKAR SOHAM VIJAY
			KSHIRSAGAR RIYA GIRISH
			SHINDE SHANTANU MANOJ
			THAKARE CHARUDATTA SUNIL
2	Crop disease detection and its recovery prediction	Machine Learning	ANDHALE CHAITANYA SANJAY
			DESHMANE VAISHNAVI SURENDRA
			JADHAV PRANALI VITTHAL
			KALE SUPRIYA BHAUSAHEB
3	Neural architecture search: A framework for automating Neural Network design	Neural Network	GHARGE SAMIKSHA DNYANESHWAR
			KARPE PIYUSH PANDURANG
			KATURDE ATHARVA DIGAMBER
			SHINDE BHAKTI BHARAT
4	Proctor vision surveillance	Computer Vision	SONDKAR MANSI SHIVAJI
			TELANGI AMISHA SUNIL
			UNHALKAR ANUSHKA RAJENDRA
			YEWALE SIDDHESH KALIDAS

Yours Sincerely,
Vishal S Gosavi,
Director

STEPUP SOLUTIONS PUNE

973 003 0609 / 935 926 6248
vishalstepup2025@gmail.com
www.stepupsolutions.in
A301, Mayuri Infinity, Behind Mayuri Vajankata, Undri, Pune

1.5 Technical Keywords (As per ACM Keywords)

- Video Classification
- Audio Classification
- Image Segmentation

1.6 Problem Statement

With the rapid advancement of AI-generated media, deepfakes have become increasingly realistic and accessible, posing serious threats to personal privacy, media integrity, and public trust. Existing detection systems struggle with real-time performance, cross-modal accuracy, and robustness to adversarial techniques. This project aims to develop a hybrid deep learning system that effectively detects both video and audio deepfakes, improving accuracy, reliability, and interpretability for real-world use cases.

1.7 Abstract

Deepfake technology has advanced significantly, leading to the creation of highly realistic yet artificially manipulated media. The rapid advancement of generative adversarial networks (GANs) and specialized deep learning techniques, including InceptionV3, XceptionNet, and EfficientNet, has further complicated the detection of deepfakes, necessitating the development of robust detection mechanisms. Datasets such as ASVspoof 2019, FaceForensics++, Celeb-DF, and the DeepFake Detection Challenge (DFDC) have significantly contributed to research by providing diverse benchmarks for training and evaluation. Detection methods span feature extraction, temporal consistency analysis, multi-modal learning, and adversarial training. Feature extraction-based approaches leverage CNNs to identify spatial artifacts, while temporal analysis captures inconsistencies in motion or lip synchronization. Multi-modal learning integrates audio and visual features to improve detection accuracy. After extensive testing and evaluation, our findings indicate that among the numerous models and datasets analyzed, InceptionV3 consistently outperforms others in identifying spatial anomalies in deepfake content. Similarly, DFDC emerges as the most comprehensive dataset for video deepfake detection, while ASVspoof 2019 remains unmatched for audio deepfake detection. These insights highlight critical tools for advancing the fight against deepfake threats and guiding future innovations in detection systems.

1.8 Goals and Objectives

The primary goal of this project is to design and implement a robust and accurate deepfake detection system capable of analyzing both video and audio content using advanced deep learning techniques. The objectives of the project are outlined as follows:

- To develop a video-based deepfake detection module using a hybrid model combining InceptionV3 for spatial feature extraction and LSTM for temporal sequence analysis.
- To implement an audio deepfake detection module using the Wav2Vec 2.0 transformer-based architecture for extracting speech embeddings and identifying spoofed audio content.
- To build a preprocessing pipeline for cleaning and standardizing media inputs, including frame extraction, resizing, face localization for videos, and segmentation and normalization for audio.
- To train and evaluate the models using publicly available benchmark datasets such as DFDC, FaceForensics++, Celeb-DF (for video), and ASVSpoof 2019 (for audio).
- To incorporate a localization feature that highlights the manipulated segments in the media for improved interpretability and user trust.
- Develop a simple and interactive user interface for the upload of input media and visualization of results.
- To ensure modular design and scalable architecture, allowing for future integration with real-time systems or cross-modal fusion.

1.9 Relevant mathematics associated with the Project

1.9.1 Convolutional Neural Networks (CNN)

$$S(i,j) = (X * K)(i,j) = \sum_m \sum_n X(i+m, j+n) \cdot K(m,n) \tag{1.1}$$

where:

- X is the input image (frame),
- K is the kernel (filter),
- S(i,j) is the output feature map at position (i,j).

1.9.2 2. Long Short-Term Memory (LSTM)

LSTM networks model temporal relationships across video frame sequences. The LSTM cell is defined as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)\ i_t \quad = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)\ \tilde{C}t = \tanh(W_C \cdot [ht-1, x_t] + b_C)\ C_t \quad = f_t \odot C_{t-1} + \tag{1.2}$$

where:

- x_t: Input at time t,
- h_t: Hidden state,
- C_t: Cell state,
- f_t,i_t,o_t: Forget, input, and output gates,
- $\odot$: Element-wise multiplication.

1.10 Names of Conferences / Journals where papers can be published

- International Conference on Advances and Applications in Artificial Intelligence (ICAAAI 2025)
- 3rd IEEE International Conference on Knowledge Engineering and Communication Systems (ICKECS-2025)

1.11 Review of Conference/Journal Papers supporting Project idea

Tong Qiao et al. proposed an unsupervised deepfake video detection system utilizing enhanced contrastive learning. They reviewed existing methods and identified the need for unsupervised learning approaches. Their system leverages feature representation learning without labeled data, significantly improving detection performance. The proposed method adapts to diverse datasets, ensuring robust and scalable detection of deepfake videos in various real-world scenarios.

Yinlin Guo et al. proposed an audio deepfake detection framework combining self-supervised WavLM and a multi-fusion attentive classifier. They determined that integrating self-supervised learning with attention mechanisms improves accuracy. The system effectively captures subtle differences in audio signals, enhancing robustness against different manipulation methods.

This innovative method demonstrates improved generalization and effectiveness across multiple datasets.

Manoj Kumar et al. presented a machine learning-based deepfake detection approach. They found that advanced machine learning algorithms, when applied to facial feature inconsistencies, yield better detection accuracy. The proposed system emphasizes lightweight algorithms for real-time applications and efficient detection, providing a practical solution to address deepfake prevalence on media platforms.

Huimin She et al. proposed a deepfake detection method leveraging graph neural networks (GNNs). They identified that GNNs effectively model spatial relationships in visual data, improving generalization. Their system enhances cross-dataset adaptability by addressing the limitations of overfitting in traditional models.

Lam Pham et al. introduced a deepfake audio detection system based on spectrogram features and ensemble deep learning models. They found that spectrograms capture subtle temporal and frequency patterns effectively. By integrating multiple architectures, the ensemble approach enhances robustness against various audio manipulations.

M. Sivabalamurugan and T. R. Swapna proposed a deepfake detection system focusing on local surface geometrical features. They identified that analyzing distortions in surface geometry significantly improves detection accuracy. The system captures minute inconsistencies in facial structures, ensuring reliable detection of both known and novel deepfake formats.

Amidela Anil Kumar et al. developed an explainable AI-enabled ensemble deep learning system for detecting deepfakes. They found that combining multiple models with interpretability improves both accuracy and trustworthiness. The system aggregates predictions from various architectures and provides robust detection with explainable outputs.

Daeun Song et al. proposed a GAN-based anomaly detection system for deepfake audio. Their system utilizes adversarial learning to identify discrepancies between real and fake audio distributions and adapts to evolving deepfake techniques for better generalization.

Xiaoke Yang et al. introduced AdaForensics, a dynamic deepfake detection framework. They found that a characteristic-aware, adaptive approach improves accuracy across diverse datasets. AdaForensics adjusts detection strategies based on specific deepfake attributes, improving robustness.

Atharva Kohapare et al. proposed a deep learning-based forgery detection system tailored for social media platforms. Their system is optimized for identifying manipulated content in low-quality uploads by integrating neural networks and preprocessing techniques.

Saima Waseem et al. presented a deepfake detection framework using attention-guided supervised contrastive learning. The system enhances feature extraction and class separation, resulting

in improved accuracy and scalability across different datasets.

Yuran Qiu et al. analyzed the vulnerabilities of deepfake detection models to backdoor attacks. They proposed defenses like adversarial training and data sanitization, emphasizing the importance of secure deepfake detection frameworks.

Cheng-Yao Hong et al. proposed a patch-based deepfake image detection technique using multiple instance learning (MIL). The image is treated as a bag of patches, and if any patch is manipulated, the entire image is classified as fake. Forged areas are localized using multi-label ranking.

Li Lin et al. addressed fairness and generalization issues in deepfake detection. Their work reveals that existing models perform inconsistently across different races and genders. Their approach enhances accuracy and fairness in detection across demographic groups.

Jongwook Choi et al.proposed a style-aware fake video detection technique using StyleGRU and contrastive learning. The system targets temporal inconsistencies in style latent vectors and uses style attention to identify visual artifacts. Their approach proved effective across multiple datasets.

1.12 Plan of Project Execution

The execution plan for the Deeflyzer Deepfake Detection System is structured into multiple phases as follows:

1.12.1 Phase 1: Requirement Analysis (Week 1)

- Define project scope: video and audio deepfake detection.
- Identify input/output formats (supported video/audio types).
- Research datasets: DFDC, FaceForensics++, Celeb-DF, ASVSpoof.
- Select deep learning models: InceptionV3, LSTM, Wav2Vec 2.0.
- Prepare software and hardware environment:
  - Python environment setup.
  - GPU availability (local, Colab, or cloud).

Deliverables: Requirement specification document, dataset sources list, model selection justification.

1.12.2 Phase 2: System Design (Week 2)

- Create architectural diagrams:
  - Data Flow Diagram (Level 0, 1, 2).
  - Class Diagram.
  - Component Diagram.
  - Deployment Diagram.
  - State Machine Diagram.
- Define module interfaces.
- Plan data pipelines:
  - Frame extraction.
  - Audio segmentation.
  - Feature extraction.
  - Classification.
  - Localization.

Deliverables: System design document with diagrams, interface specifications.

1.12.3 Phase 3: Data Collection & Preprocessing (Week 3–4)

- Download datasets: DFDC, FaceForensics++, Celeb-DF, ASVSpoof.
- Perform data cleaning:
  - Extract video frames.
  - Crop/resize faces (299x299 for InceptionV3).
  - Normalize audio samples.
- Organize datasets into training, validation, and test splits.
- Perform data augmentation (optional).

Deliverables: Preprocessed datasets, data statistics summary.

1.12.4 Phase 4: Model Development (Week 5–7)

- Implement InceptionV3 feature extractor (transfer learning).
- Build LSTM layer to process feature sequences.
- Implement Wav2Vec 2.0 audio feature extractor.
- Build classifier layers (sigmoid or softmax).
- Integrate modules into pipeline.
- Train video model on DFDC/FaceForensics++.
- Train audio model on ASVSpoof.
- Evaluate models using accuracy, precision, recall, F1-score.

Deliverables: Trained InceptionV3 + LSTM model, trained Wav2Vec 2.0 classifier, training logs, evaluation metrics.

1.12.5 Phase 5: Integration & Testing (Week 8–9)

- Integrate video and audio detection pipelines.
- Implement localization module (highlight tampered regions).
- Build frontend (NextJS):
  - Upload interface.
  - Display prediction result.
  - Highlight localization.
- Connect frontend to backend API.
- Test full workflow:
  - Upload media.
  - Detect → classify → localize → return result.
- Validate outputs on test set.

Deliverables: Integrated system, demo run outputs, user guide.

1.12.6 Phase 6: Deployment (Week 10)

- Package system (Docker if needed).
- Deploy backend API on server (local or cloud).
- Host frontend (or deploy via NextJS sharing).
- Connect data storage paths.
- Conduct user testing.

Deliverables: Deployed system ready for demonstration, deployment report.

1.12.7 Phase 7: Documentation & Final Report (Week 11)

- Write:
  - System design documentation.
  - Model training report.
  - Evaluation and benchmark comparison.
  - User manual.
  - Research-style paper (optional for publication).
- Prepare presentation slides.

Chapter 2

Technical Keywords

2.1 Area of Project

The area of this project lies in the intersection of **Artificial Intelligence (AI)**, **Machine Learning (ML)**, and **Multimedia Forensics**. Specifically, it focuses on the application of deep learning techniques for the detection of manipulated media, commonly referred to as *deepfakes*.
This project integrates both visual and audio modalities using state-of-the-art models such as **InceptionV3** for video frame analysis and **Wav2Vec 2.0** for audio deepfake detection. It also utilizes temporal modeling through **Long Short-Term Memory (LSTM)** networks for enhanced video sequence analysis. The system falls under the broader domain of:

- Computer Vision
- Speech Processing
- Deep Learning and Neural Networks
- Cybersecurity and Digital Media Authentication

The project contributes to the development of intelligent systems capable of safeguarding digital integrity and combating misinformation by verifying the authenticity of media content.

2.2 Technical Keywords

1. I. Computing Methodologies
    (a) I.2 ARTIFICIAL INTELLIGENCE
        i. I.2.6 Learning
            A. A. Connectionism and neural nets
            B. B. Concept learning
            C. C. Knowledge acquisition
            D. D. Machine learning algorithms
        ii. I.2.7 Natural Language Processing
            A. A. Speech recognition and synthesis
            B. B. Language models
    (b) I.4 IMAGE PROCESSING AND COMPUTER VISION
        i. I.4.8 Scene Analysis
            A. A. Tracking
            B. B. Face and gesture recognition

C. C. Time series analysis

D. D. Motion

2. H. Information Systems

   (a) H.2 DATABASE MANAGEMENT

      i. H.2.8 Database Applications

         A. A. Data mining

3. K. Computing Milieux

   (a) K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS

      i. K.6.5 Security and Protection

         A. A. Authentication

         B. B. Unauthorized access (e.g., hacking, phreaking)

Chapter 3

Introduction

3.1 Motivation

With the rapid advancement of artificial intelligence, deepfake technology has emerged as both a groundbreaking innovation and a significant cybersecurity threat. While deepfakes offer creative possibilities in entertainment and media, they also pose serious risks, including misinformation, identity theft, financial fraud, and reputational damage.

The increasing accessibility of deepfake generation tools has made it easier to manipulate digital content, leading to ethical and security concerns. High-profile cases of deepfake misuse in politics, finance, and personal identity highlight the urgent need for robust detection mechanisms. Existing solutions, such as Sensity, provide a foundation for deepfake detection but face limitations in accuracy, generalization, and real-time processing. By enhancing detection accuracy and reducing false positives, our system seeks to contribute to digital forensics, cybersecurity, and media integrity. The ultimate goal is to create a reliable and scalable solution that can help individuals, organizations, and governments combat the growing threat of deepfake-based deception.

3.2 Project Idea

The widespread use of deepfake technology has introduced significant challenges to societal trust, media authenticity, and ethical communication. Deepfake videos and audio, powered by advanced AI techniques such as Generative Adversarial Networks (GANs), are increasingly being used to manipulate reality, leading to misinformation, social unrest, and reputational damage.

One of the biggest concerns is the role of deepfakes in spreading false information across social media platforms. Fake political speeches, fabricated news reports, and altered testimonies can mislead the public, influencing elections, public opinion, and policy decisions. In a digital age where people rely on online media for information, the inability to verify authenticity can lead to mass confusion and distrust.

Moreover, deepfakes pose a serious threat to individual privacy and reputation. Fake videos impersonating celebrities, politicians, or even ordinary individuals have been used for defamation, harassment, and cyberbullying. This has led to psychological distress and legal challenges, especially for victims who struggle to prove their innocence in the face of highly convincing fake content.

Another critical issue is the erosion of trust in media and journalism. As deepfake technology becomes more sophisticated, people may start questioning even authentic news sources, leading to a post-truth society where facts become uncertain. This not only weakens democracy but also makes it easier for malicious actors to manipulate narratives for their benefit.

To address these concerns, this project aims to develop a robust deepfake detection system that can help verify digital content and ensure media integrity.

3.3 Literature Survey

Kuiyuan Zhang [1] and their team members studied about improving the generalization in deepfake detection. In their paper they addressed the issue of catastrophic forgetting which is a concept where a model when learns about new deepfake techniques, it tends to forget its existing or prior knowledge about classical deepfake techniques. This phenomenon significantly degrades the quality of the model.

B.V. Chowdary [2] along with their associates proposed an effective deepfake detec tion system. They surveyed all the existing model and found out that the combi-nation of CNN and RNN proves to be the most efficient in detecting deepfakes. The pro posed system makes use of the ResNet which is a CNN and LSTM which a RNN architecture.

B. Sarada [3] with their colleagues conducted a study on the topic of audio deepfake detection. In their study they proposed a solution in detecting AI cloned voices. They made use of the Generative Adversarial Network (GAN) along with Random Forest which is a machine learning algorithm usually used for classification.

Hao Teng [4] along with their team members proposed a solution on the problem of cross forgery which occurs when a model tries to detect a type of deepfake for which it was not originally trained. To overcome this problem, they suggested the use of extraction of static and dynamic features. Static features will be used to detect the old techniques of deepfake and dynamic features will be used to detect new techniques of deepfake. As per their experiments it is observed that this approach proves to be four times more accurate than single feature extraction models.

Haobo Liang, Yingxiong Leng [5] and their associates studied deeply in the field of face forgery. Their study indicates that the traditional methods for detecting for-gery cannot detect the latest forging techniques. They introduce a novel approach in detect ing forgery with knowledge distillation and DCT. It will achieve high amount of ac curacy and precision and will be able to detect subtle changes done on the face. 4

Amaan M. Kalemullah [6] with their co-authors conducted a study in the field of deepfake detec-

tion. They mainly focused on detection of deepfake in Human Faces. Their research proposes a comprehensive approach of using Convolutional Neural Network along with two Transfer Learning models ResNet-50 and EfficientNet B7 for detecting inconsistencies in human faces. These models when evaluated gave out the best accuracy in detecting manipulated facial content.

Cheng-Yao Hong [7] and their associates proposed a new way of detecting and identi fying deepfakes in images. Their approach to detecting deepfake is such that it divides the images into smaller parts called as patches and then it considers the whole image as a bag of patches. If one patch is manipulated then the whole im-age will be consid ered as fake and to do that, they used a unique way called the multiple instance learn ing (MIL). For identifying the specific part of the im-age which has been deepfake it used the multi-label ranking which label all parts of an images and returns the forged part.

Li Lin [8] along with their colleagues provided a solution to a problem is which oc curs while detecting deepfakes. In their paper they addressed the issue of fairness generalization. Their study states that existing deepfake models are trained to detect manipulations but they are not efficient in detecting for people from different race and gender. The experiments conducted by them state that using this method the accuracy and effectiveness of the state-of-the-art methods can be easily surpassed.

Jongwook Choi [9] and their team presents a new approach in detecting fake videos based on style latent vectors. Their approach consists of targeting the temporal incon sistencies in fake videos. They have made use of the StyleGRU module which has been trained using contrastive learning used to represent the feature of style latent vectors. They also added a style attention module to detect visual artifacts. After test ing this approach on multiple datasets, it is stated that it proves very effective.

Chuangchuang Tan [10] along with their colleagues studied the problems occurring in existing deepfake detection models. These differences are present in the images ma nipulated or generated using GAN. To tackle this, they introduced a new method Neighboring Pixel Relationships (NPR) which is used to identify these differences. This method showcases a 12.8methods.

Trevine Oorloff [11] with their associates introduce a new method called audio visual feature fusion which is learning method divided in two stages and they detect the differences between the audio and visual modalities. In the second stage the represen tations of the features are tuned

and actual deepfake classification is done. This ap proach deals 98.6the state-of-the-art methods. 5

Siyou Guo [12] and their team members studied the existing techniques used for deep-fake detection. In their study, they encountered a problem. The existing models used for deepfake detection ignore the subtle variation in the media. To over-come this issue, they proposed a progressive attention network which incorporates two at tention modules which are Efficient Multi-Scale Attention Module and Spatial and Channel Attention Module.

Tong Qiao [13] and associates proposed an unsupervised deepfake video detection system utilizing enhanced contrastive learning. They reviewed existing methods and identified the need for unsupervised learning approaches. Their system leverages feature representation learning without labeled data, significantly improving detection performance. The proposed method adapts to diverse datasets, ensuring robust and scalable detection of deepfake videos in various real-world scenarios.

Yinlin Guo [14] and their team proposed an audio deepfake detection framework com-bining self-supervised WavLM and a multi-fusion attentive classifier. They surveyed existing techniques and determined that integrating self-supervised learning with attention mechanisms improves accuracy. The system effectively captures subtle differences in audio signals, enhancing robustness against different manipulation methods. This innovative method demonstrates improved generalization and effectiveness across multiple datasets.

Manoj Kumar [15] and associates presented a machine learning-based deepfake detec-tion approach. They reviewed existing models and found that advanced machine learning algorithms, when applied to facial feature inconsistencies, yield better detection accuracy. The proposed system emphasizes lightweight algorithms for real-time applications and efficient detection, providing a practical solution to address deepfake prevalence on media platforms.

Huimin She [16] and their colleagues proposed a deepfake detection method leveraging graph neural networks (GNNs). Through their research, they identified that GNNs effectively model spatial relationships in visual data, improving generalization. Their system enhances cross-dataset adaptability by addressing the limitations of overfitting in traditional models. This approach ensures robustness against diverse deepfake types and formats, advancing detection capabilities.

Lam Pham [17] and associates introduced a deepfake audio detection system based on spec-

trogram features and ensemble deep learning models. Their analysis revealed that spectrograms capture subtle temporal and frequency patterns effectively. By integrating multiple architectures, the ensemble approach enhances robustness against various audio manipulations. This method outperforms traditional systems, making it a reliable tool for detecting audio deepfakes in real-world scenarios.

M Sivabalamurugan [18] and T R Swapna proposed a deepfake detection system fo-cusing on local surface geometrical features. They examined existing methods and identified that analyzing distortions in surface geometry significantly improves detection accuracy. The proposed system captures minute inconsistencies in facial structures, ensuring reliable detection of both known and novel deepfake formats. This approach offers a practical solution for combating digital face manipulation.

Amidela Anil Kumar [19] and colleagues developed an explainable AI-enabled ensemble deep learning system for detecting deepfakes. They reviewed existing methods and found that combining multiple models with interpretability improves both accuracy and trustworthiness. The system aggregates predictions from various architectures, providing robust detection against diverse fake formats. The explainability aspect enhances user confidence, making it suitable for critical applications like forensics and media monitoring.

Daeun Song [20] and their team proposed a GAN-based anomaly detection system for deepfake audio. Through their research, they identified that adversarial learning effectively identifies discrepancies between real and fake audio distributions. Their system adapts to evolving deepfake techniques and outperforms traditional methods in accuracy and generalization, offering a robust framework for audio deepfake detection in various applications.

Xiaoke Yang [21] and colleagues introduced AdaForensics, a dynamic deepfake detection framework. They identified that a characteristic-aware, adaptive approach improves accuracy across diverse datasets. AdaForensics adjusts detection strategies based on specific deepfake attributes, ensuring consistent performance against emerging manipulation techniques.

Chapter 4

Problem Definition and Scope

4.1 Problem Statement

4.1.1 Goals and objectives

- **Goal 1:** Develop a hybrid deepfake detection system capable of analyzing both video and audio modalities to improve detection accuracy.
- **Goal 2:** Integrate InceptionV3 and LSTM for spatial-temporal analysis of facial video frames.
- **Goal 3:** Employ Wav2Vec 2.0 to extract and classify latent features from audio signals.
- **Goal 4:** Create a user-friendly interface for uploading media and visualizing results, including localized manipulated regions.
- **Goal 5:** Provide reliable classification of media as Real or Fake and generate analytical reports for further review.

Objectives:

- Extract frames and detect faces from uploaded videos using MTCNN.
- Normalize and segment audio input for deep analysis.
- Train and test models using benchmark datasets such as DFDC and ASVspoof2019.
- Achieve a minimum accuracy threshold for both audio and video classification.
- Deploy the system with a functional frontend (NextJS) for real-time use.

4.1.2 Statement of scope

This project focuses on the design and implementation of a deepfake detection system that processes multimedia content (audio and video) to identify synthetic manipulations. The system utilizes pre-trained deep learning models including InceptionV3, LSTM, and Wav2Vec 2.0 to analyze spatial, temporal, and speech patterns. It offers a dual-path architecture that individually analyzes audio and video, and combines their results for final classification.
The scope includes preprocessing input data (frame extraction, face detection, audio normalization), feature extraction, classification, and display of results through a web-based interface. The system is aimed at academic research, digital forensics, and content authentication. It excludes real-time surveillance and mobile deployment, which may be explored in future extensions.

4.2 Major Constraints

The development of the Hybrid Deepfake Detection System is subject to the following major constraints, which influence the software's specification, design, implementation, and testing:

- **Hardware Limitations:** High computational power and GPU support are essential for real-time inference using deep learning models such as InceptionV3 and Wav2Vec 2.0. In the absence of dedicated hardware, performance may degrade significantly.
- **Model Size and Loading Time:** The pre-trained models used (e.g., Wav2Vec 2.0, InceptionV3 with LSTM) are memory-intensive and may cause long loading or inference times, particularly on systems with limited RAM or CPU/GPU capabilities.
- **Dataset Constraints:** The accuracy and generalizability of the system are highly dependent on the quality and diversity of the training datasets. Deepfake datasets such as DFDC and ASVspoof may not cover all real-world variations.
- **Real-time Detection:** The current system is not optimized for real-time video streaming or live media input. It only supports static file uploads for offline detection.
- **Audio Format Compatibility:** Input audio must be in or converted to `.wav` format. Additional preprocessing is required for other formats, which may introduce latency.
- **Browser/Platform Dependencies:** The front-end interface developed using NextJS may behave differently across browsers and platforms, limiting portability without proper cross-platform testing.
- **Security Concerns:** As media files are uploaded and processed, ensuring secure handling of potentially sensitive content is critical but not fully implemented in this version.
- **Scalability Constraints:** The current design supports single-user operations. Scaling the system to support concurrent users or integration into larger platforms would require architectural redesign.

4.3 Methodologies of Problem solving and efficiency issues

The problem of deepfake detection can be addressed using multiple methodologies, each with varying degrees of performance, complexity, and accuracy. This project adopts a hybrid deep learning-based approach to maximize generalizability and detection precision, while being mindful of computational efficiency.

- **Traditional Feature-Based Methods:** Earlier approaches relied on handcrafted features such as head pose estimation, blinking patterns, or frequency inconsistencies. While computationally less expensive, these methods lack robustness against advanced generative models and often fail under real-world scenarios with varying quality and resolution.
- **CNN-Based Methods:** Convolutional Neural Networks (e.g., ResNet, Xception, InceptionV3) provide significant improvements in performance by learning visual features automatically. In this project, **InceptionV3** is employed due to its efficient architecture that processes multiple filter sizes in parallel, improving feature diversity and reducing redundant computation.
- **Temporal Modeling with LSTM:** Spatial-only analysis may miss temporal inconsistencies in videos. To address this, **Long Short-Term Memory (LSTM)** networks are integrated after feature extraction to capture sequential patterns across frames, thereby increasing robustness against subtle frame-level manipulations.
- **Audio Deepfake Detection using Wav2Vec 2.0:** For audio, transformer-based models like **Wav2Vec 2.0** outperform traditional spectral feature-based classifiers. Although computationally intensive, they offer superior accuracy and noise robustness in detecting synthetic speech.
- **Efficiency Trade-offs:** While hybrid models offer high detection accuracy, they require significant computational resources. To balance performance and efficiency:
  - Pre-trained models are used to reduce training time.
  - Frame and face truncation is applied to limit input size.
  - Lightweight preprocessing is preferred to minimize latency.

Overall, the chosen methodology prioritizes detection accuracy and generalization across diverse datasets, while selectively optimizing modules to manage efficiency constraints.

4.4 Outcome

The outcome of this project is a fully functional hybrid deepfake detection system capable of identifying manipulated media content in both audio and video formats. The system integrates state-of-the-art deep learning models including InceptionV3 with LSTM for video frame analysis and Wav2Vec 2.0 for audio classification. The key outcomes are:

- A robust multimedia authentication tool that detects deepfakes with high accuracy using both spatial and temporal features.

- Implementation of an end-to-end pipeline involving frame extraction, face detection, feature extraction, classification, and result reporting.
- A user-friendly web interface developed using NextJS that allows users to upload media files and view detection results interactively.
- The ability to visualize and highlight detected fake faces in video frames, thereby enhancing interpretability for the end-user.
- Audio preprocessing support including format conversion and resampling to ensure compatibility with the classification model.
- Integration of pre-trained models to reduce training time and leverage generalized knowledge from large-scale datasets.
- Real-time classification feedback provided to users for both video and audio deepfake detection tasks.

This project contributes to the growing need for digital content verification tools and can be used in media forensics, social media monitoring, and cybersecurity applications.

4.5 Applications

The hybrid deepfake detection system has wide-ranging applications across multiple domains that require authentication and verification of multimedia content. Some of the key applications include:

- **Digital Media Forensics:** Used by forensic analysts to verify the authenticity of video and audio evidence in legal and investigative contexts.
- **Social Media Platforms:** Can be integrated into platforms like Facebook, Instagram, or YouTube to automatically detect and flag manipulated media content before it spreads.
- **News and Journalism:** Assists media organizations in fact-checking and validating the integrity of media content before publication.
- **Law Enforcement Agencies:** Enables the detection of fake videos or audio used for misinformation, extortion, or character defamation in cybercrime cases.
- **Video Conferencing and Communication Tools:** Helps detect spoofed identities and manipulated content in video calls or voice communications, enhancing user trust and platform security.

- **Content Moderation:** Supports automated moderation systems in identifying synthetic or harmful media and enforcing community guidelines.
- **Media Archives and Libraries:** Ensures the archival of only verified and authentic media, preserving digital history with integrity.
- **Education and Research:** Serves as a case study or tool in academic settings for exploring AI ethics, multimedia security, and neural network applications.
- **Corporate and Political Security:** Helps prevent the misuse of synthetic media to impersonate public figures or executives in scams or misinformation campaigns.

4.6 Hardware Resources Required

The following hardware resources are necessary for the effective development, training, and deployment of the hybrid deepfake detection system:

- **Processor (CPU):** Minimum Intel Core i5 / AMD Ryzen 5 or higher; recommended Intel Core i7 / AMD Ryzen 7 for faster data preprocessing and model loading.
- **Graphics Processing Unit (GPU):** A CUDA-enabled NVIDIA GPU with a minimum of 4 GB VRAM (e.g., GTX 1650) is recommended for deep learning inference. For training or large-scale evaluation, 8 GB or higher (e.g., RTX 3060 or better) is preferable.
- **RAM:** Minimum 8 GB of system memory; 16 GB or more is recommended to handle concurrent processes such as video frame extraction and model inference.
- **Storage:** At least 50 GB of free disk space to store video/audio datasets, extracted frames, pre-trained model weights, and intermediate outputs.
- **Audio-Visual Input Devices (Optional):** Webcam and microphone for real-time media capture and testing, particularly during demonstrations or GUI-based deployments.
- **Operating System:** Windows 10/11, Linux (Ubuntu 20.04 or higher), or macOS with support for TensorFlow, PyTorch, CUDA, and compatible drivers.

4.7 Software Resources Required

The development and deployment of the hybrid deepfake detection system require the following software resources:

- **Operating System:**
  - Windows 10/11, Ubuntu 20.04+, or macOS
- **Programming Language:**
  - Python 3.8 or above
- **Deep Learning Libraries:**
  - TensorFlow (v2.x)- for InceptionV3 and LSTM-based video classification
  - PyTorch (v2.x)- for Wav2Vec 2.0 audio deepfake detection
- **Audio Processing:**
  - Torchaudio - for loading and resampling audio
  - Pydub - for audio format conversion
- **Computer Vision:**
  - OpenCV - for video frame extraction and image handling
  - MTCNN - for facial detection in frames
- **Web Interface Frameworks:**
  - NextJS - for interactive web-based user interface (optional)
  - FastAPI - for backend API and integration with frontend (optional)
- **Model Deployment:**
  - HuggingFace Transformers - for downloading and using Wav2Vec 2.0
  - Keras - high-level API for TensorFlow models
- **Development Tools:**
  - Jupyter Notebook / VS Code / PyCharm - for model prototyping and testing
  - Git - for version control and collaboration

[utf8]inputenc

Chapter 5

Project Plan

5.1 Project Estimates

5.1.1 Effort Estimation

The following is a breakdown of estimated effort required for each phase of the hybrid deepfake detection system project:

- **Requirement Gathering and Analysis:** 8 person-hours
- **Dataset Collection and Preprocessing (DFDC, ASVspoof2019):** 20 person-hours
- **Model Development:**
  - InceptionV3 for frame-based feature extraction - 15 person-hours
  - LSTM for video sequence learning - 10 person-hours
  - Wav2Vec 2.0 for audio classification - 12 person-hours
- **Hybrid Integration and Feature Fusion:** 10 person-hours
- **Model Training and Validation:** 20 person-hours
- **Result Analysis and Localization Visualization:** 8 person-hours
- **Frontend Development (NextJS):** 12 person-hours
- **Testing and Debugging:** 10 person-hours
- **Documentation and Report Writing:** 10 person-hours

Total Estimated Effort: 125 person-hours

5.1.2 Cost Estimation (Assuming ₹500 per hour)

- **Development Cost (Labor):** ₹62,500
- **Cloud/Hardware Resources (GPU, Storage):** ₹15,000
- **Miscellaneous (Internet, Tools, Subscriptions):** ₹5,000

Total Estimated Cost: ₹82,500

5.1.3 Time Estimation

- **Total Duration:** 8-10 weeks
- **Team Size:** 4 members
- **Working Hours per Week:** 30-32 hours (approx. 8 hours/member/week)
- **Delivery Milestones:**
  - Phase 1: Dataset and Preprocessing - Week 2
  - Phase 2: Model Development - Week 4
  - Phase 3: System Integration and Testing - Week 7
  - Phase 4: Final Report and Deployment - Week 10

5.2 Risk Mitigation, Monitoring, and Management Plan

5.2.1 Risk Identification

Risk ID	Risk Description	Category
R1	Dataset quality or availability issues	Technical
R2	Model overfitting during training	Technical
R3	Limited computing resources for training models	Resource
R4	Lack of real-time detection support	Functional
R5	Incorrect labeling in datasets	Data Quality
R6	Integration complexity (e.g., combining models)	Technical
R7	Deadline slippage due to scope creep	Project Schedule
R8	Difficulty in audio deepfake detection	Research
R9	Software/library compatibility issues	Technical
R10	Lack of user-friendly output interface	UI/UX

Table 5.1: Risk Identification Table

Purpose:

This table helps systematically identify potential risks that may affect the project across different categories such as technical, data quality, functional, and project management.

5.2.2 Risk Assessment Matrix

Risk ID	Probability (1-5)	Impact (1-5)	Risk Exposure	Priority
R1	4	4	16	High
R2	3	5	15	High
R3	4	4	16	High
R4	2	3	6	Medium
R5	3	4	12	High
R6	2	3	6	Medium
R7	2	4	8	Medium
R8	3	5	15	High
R9	3	3	9	Medium
R10	2	2	4	Low

Table 5.2: Risk Assessment Table

Purpose:

This table quantifies each identified risk using a Probability (P) and Impact (I) score, and then calculates Risk Exposure to determine the severity and assigns a priority level (High, Medium, Low).

5.2.3 Risk Mitigation Plan Table

Risk ID	Probability (1-5)	Impact (1-5)	Risk Exposure	Priority
R1	4	4	16	High
R2	3	5	15	High
R3	4	4	16	High
R4	2	3	6	Medium
R5	3	4	12	High
R6	2	3	6	Medium
R7	2	4	8	Medium
R8	3	5	15	High
R9	3	3	9	Medium
R10	2	2	4	Low

Table 5.3: Risk Mitigation Table

Purpose: This table provides specific, actionable strategies to reduce the likelihood or impact of each risk identified.

5.2.4 Risk Monitoring Plan Table

Risk ID	Monitoring Plan
R1	Weekly dataset review checkpoints. Report anomalies in data logs.
R2	Monitor training/validation accuracy and loss every epoch. Visualize using TensorBoard.
R3	Log GPU/CPU/memory usage. Switch to cloud if local fails.
R4	Track feature development scope with supervisor. Avoid adding new features mid-way.
R5	Random sample verification before each major model run.
R6	Unit testing after each module completion.
R7	Weekly progress meetings with milestone reviews.
R8	Literature review of new audio deepfake papers. Log model accuracy trends.
R9	Check compatibility before library upgrades. Use version control.
R10	User feedback from mock demos. Iterate UI once.

Table 5.4: Risk Monitoring Plan

Purpose: This outlines how each risk will be tracked during the project, including metrics, tools, and frequency of checks.

5.3 Project Schedule

The development of the hybrid deepfake detection system is divided into the following phases with associated tasks and deliverables:

Phase	Duration	Activities and Deliverables
Phase 1: Requirement Analysis and Planning	Week 1	• Identify project scope, objectives, and user requirements • Select suitable datasets (DFDC, ASVspoof 2019) • Deliverables: Project Plan Document, Use Case Diagram
Phase 2: Data Preparation and Preprocessing	Week 2-3	• Frame extraction, face detection, audio extraction • Data cleaning and normalization • Deliverables: DFD Level 0, 1, and 2, ER Diagram
Phase 3: Model Design and Training	Week 4-6	• InceptionV3 for video, LSTM for sequence analysis • Wav2Vec 2.0 for audio classification • Deliverables: Class Diagram, Activity Diagram, Sequence Diagram
Phase 4: System Integration and UI Development	Week 7-8	• Integrate backend models with NextJS GUI • Build classification dashboard and output visualization • Deliverables: GUI Screenshots, Architecture Diagram

Phase	Duration	Activities and Deliverables
Phase 5: Testing and Documentation	Week 9-10	• Functional and non-functional testing • Documentation, final report, and user manual • Deliverables: Test Plan, Final Report, Deployment Guide

5.3.1 Project Task set

The development of the Hybrid Deepfake Detection System was broken down into the following major tasks to ensure systematic progress and effective team collaboration:

- **Requirement Analysis:**
  - Identify functional and non-functional requirements.
  - Define system scope and technical constraints.
  - Survey existing deepfake detection techniques.
- **Dataset Collection and Preprocessing:**
  - Collect video and audio deepfake datasets (e.g., DFDC, Celeb-DF).
  - Perform frame extraction and face detection.
  - Normalize audio inputs and convert to compatible formats.
- **Model Development:**
  - Train InceptionV3 + LSTM model for video deepfake detection.
  - Fine-tune Wav2Vec 2.0 model for audio deepfake classification.
  - Validate and evaluate model performance.
- **System Integration:**
  - Integrate models into a unified detection pipeline.
  - Design the backend logic to manage inputs, predictions, and outputs.
- **Frontend Development:**

  - Build the user interface using NextJS.
  - Enable file uploads and display results with detected fake content.

- **Testing and Validation:**
  - Conduct unit, integration, and system testing.
  - Measure performance and accuracy metrics.

- **Documentation and Reporting:**
  - Prepare user and developer documentation.
  - Compile the final project report and presentation.

5.3.2 Task Network

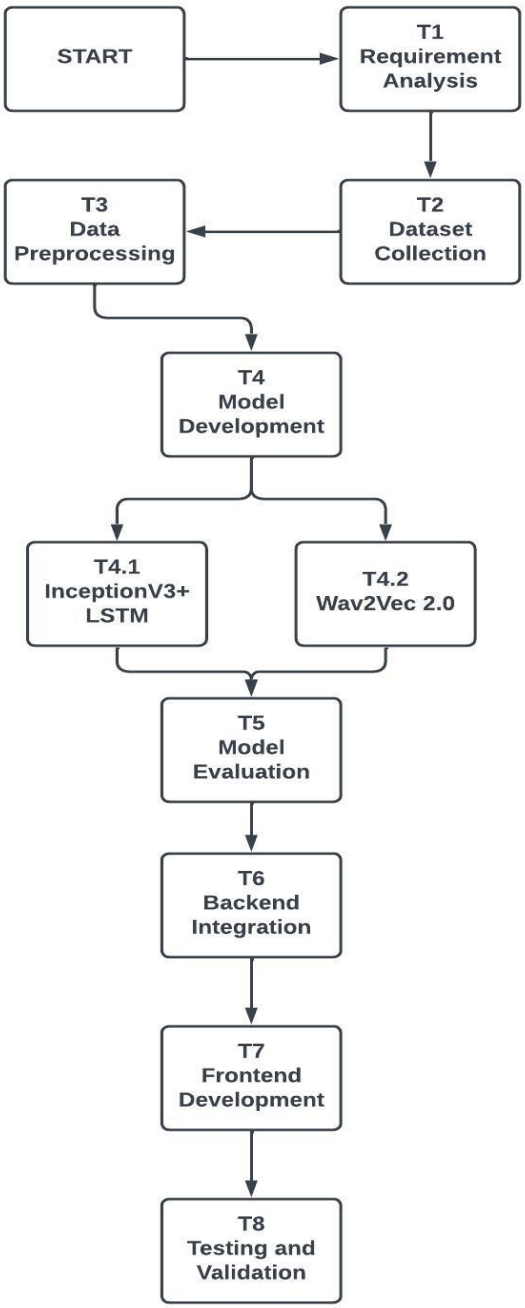


Figure 5.1: Task Network

5.3.3 Timeline Chart

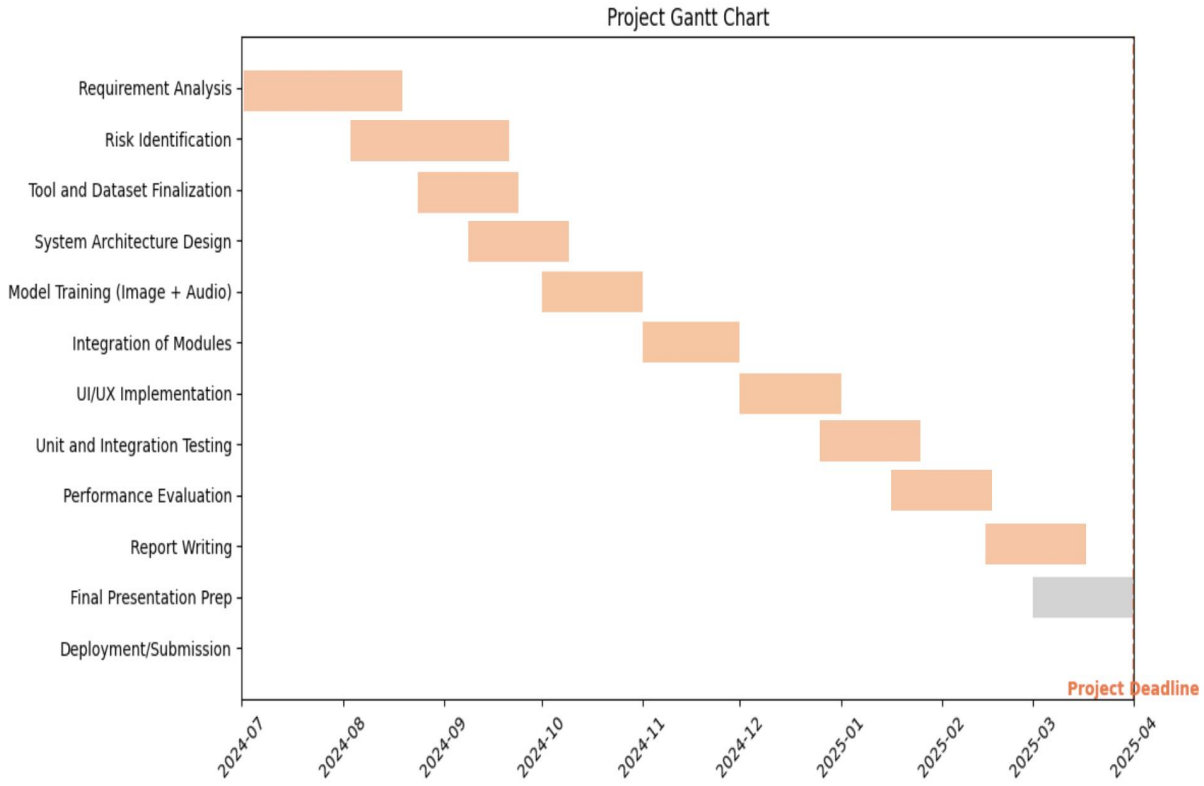


Figure 5.2: Gantt Chart

5.4 Team Organization

5.4.1 Team Structure

The development of the Hybrid Deepfake Detection System was carried out by a team of four members. The team was structured to ensure balanced distribution of responsibilities and smooth collaboration throughout the project lifecycle. The roles and responsibilities of each member are outlined below:

- **Team Leader/Full Stack Developer - Soham Kolapkar**
  Responsible for overall project coordination, milestone tracking, report finalization, and integration of all components. Acted as the point of contact between the team and external supervisors or evaluators. Handled the implementation of core deepfake detection models using TensorFlow and PyTorch. Focused on video frame processing, audio preprocessing,

and model integration logic.

- **Frontend Developer - Riya Kshirsagar**
  Developed the graphical user interface using NextJS. Ensured smooth user interaction, file handling, and real-time display of results including detected fake faces.

- **Backend Developer - Charudatta Thakare**
  Handled the implementation of core deepfake detection models using TensorFlow and PyTorch. Focused on video frame processing, audio preprocessing, and model integration logic.

- **Data Analyst and Tester - Shantanu Shinde**
  Responsible for dataset preprocessing, evaluation metrics analysis, model performance testing, and writing automated test cases for validation of system outputs.

5.4.2 Management reporting and communication

- **Progress Reporting:**
  - Weekly status reports were prepared by the team leader and shared with the mentor during scheduled lab sessions.
  - Each report included updates on completed tasks, issues faced, solutions implemented, and upcoming goals.
  - Demonstrations of modules were conducted during lab hours for validation and feedback.

- **Intra-Team Communication:**
  - A shared Google Drive was used for real-time collaboration on code, reports, and documentation.
  - WhatsApp and Telegram groups were used for daily coordination and quick decision-making.
  - Google Meet and in-person lab discussions were conducted weekly to plan development milestones and resolve technical issues.

- **Mentor and Faculty Communication:**
  - Regular updates were given to the project mentor during allotted lab hours, as per the institutional timetable.
  - Mentor feedback was documented and action points were discussed and tracked in the following sessions.

Chapter 6

Software Requirements Specification

6.1 Introduction

6.1.1 Project Scope

The scope of this project encompasses the development of a hybrid deepfake detection system capable of analyzing both video and audio media to identify manipulated content with high precision. It integrates advanced deep learning models like InceptionV3 combined with LSTM for spatial-temporal video analysis and Wav2Vec 2.0 for audio-based deepfake detection-enabling the system to capture both visual and vocal irregularities. The project focuses on processing media inputs by extracting frames, detecting faces, and evaluating facial and vocal authenticity through neural network-based classification. It also includes the implementation of a user friendly web interface using NextJS, allowing real-time media upload, analysis, and result visualization. The system supports face localization, helping users visualize which parts of the video are potentially fake. Its design allows deployment in practical applications such as journalism, law enforcement, content moderation, and digital forensics. The project is scalable and can be extended to support live media streams, larger datasets, or additional deepfake detection models in future developments.

6.2 Overview of responsibilities of Developer

- **Model Design and Implementation:**
  - Developed and fine-tuned deep learning models using TensorFlow and PyTorch, including InceptionV3 with LSTM for video-based detection and Wav2Vec 2.0 for audio-based classification.
  - Implemented face detection using the MTCNN algorithm and integrated it with the frame extraction module.
- **Data Preprocessing:**
  - Extracted video frames and normalized audio samples.
  - Handled image resizing, data augmentation, and format conversion for consistent model input.
- **Backend Development:**
  - Created the backend logic to support both audio and video analysis workflows.
  - Managed integration of the machine learning models with the user interface for real-time prediction.

- **Testing and Debugging:**
  - Performed unit testing for model predictions and I/O functions.
  - Validated system performance against standard datasets and real-world inputs.
- **Optimization and Deployment:**
  - Optimized model loading and inference time to improve usability.
  - Supported deployment using NextJS to provide a smooth user experience.
- **Documentation and Reporting:**
  - Contributed to the preparation of user manuals and project documentation.
  - Recorded issues, changes, and resolutions during each phase of development.

6.3 Usage Scenario

6.3.1 User Profiles

The system interacts with various categories of users. Each user type has a specific role and responsibility within the Deepfake Detection System. The descriptions of all user categories are provided below:

- **Admin:** Responsible for managing the overall system operations, including dataset management, user access control, and system performance monitoring. Admins can retrain models, update configurations, and oversee logs.
- **Researcher / Developer:** Individuals involved in developing, testing, and optimizing the hybrid detection model. They utilize the system for evaluating model accuracy, performing experiments with datasets, and integrating improvements such as localization of manipulated media.
- **Media Analyst:** Professionals who analyze video and audio content to verify its authenticity. They use the system to upload suspect media and interpret the predictions, including visualization of tampered segments.
- **General User:** Any individual such as a journalist, student, or digital consumer who wants to check whether a media file is real or fake. General users can upload content and view detection results but do not have access to system settings.

- **NGO / Legal Authority:** Governmental or non-governmental bodies, legal institutions, or law enforcement agencies that use the system for digital evidence validation. They use the deepfake detection results to support investigations or reports involving manipulated media.

6.3.2 Use Cases

Sr No.	Use Case	Description	Actors	Assumptions
1	Flag Deepfake Media on Social Platforms	Automatically detect and flag deepfake videos and audios spreading misinformation or political propaganda.	Social Media Platforms	Uploaded content is scanned in real time.
2	News Verification	Check authenticity of submitted video/audio before publishing news to prevent misinformation.	Journalists, News Agencies	Files are submitted in supported formats with metadata.
3	Biometric Security Validation	Detect facial or voice deepfakes attempting to breach biometric authentication systems.	Authentication Systems, Security Admins	System is integrated with deepfake detection engine.
4	Voice-based Fraud Detection	Detect voice deepfakes in financial voice banking systems to prevent fraud.	Financial Institutions	Access to voice communication logs is available.
5	Forensic Evidence Validation	Analyze video/audio evidence to detect forgery in legal/criminal investigations.	Law Enforcement, Forensics Teams	Media used in cases is accessible for scanning.
6	Cybercrime Deepfake Tracking	Identify deepfake content used in online scams or harassment.	Cybercrime Units	Offending content is collected from online sources.

Sr No.	Use Case	Description	Actors	Assumptions
7	Celebrity Deepfake Protection	Detect unauthorized use of celebrity faces or voices in malicious deepfake content.	Entertainment Industry, IP Lawyers	Public figure data is registered for comparison.
8	Copyright Violation Detection	Identify unauthorized use of copyrighted media in manipulated content.	Copyright Authorities	Original copyrighted datasets are available.
9	Fake Advertisement Detection	Detect deepfake media in misleading product reviews or promotional content.	E-commerce Platforms, Brand Managers	Media metadata is analyzed for source verification.
10	Endorsement Authenticity Verification	Ensure product endorsements are genuine and not impersonated by deepfakes.	Companies, Marketing Teams	Public representative profiles are pre-verified.
11	Online Exam Impersonation Check	Validate identity of students during remote exams to prevent deepfake impersonation.	Educational Institutions	Real-time camera/mic access is allowed.
12	Virtual Meeting Authentication	Verify identity in sensitive online meetings to block deepfake impersonators.	Remote Work Organizations	Real-time identity checks are supported.

6.3.3 Use Case View

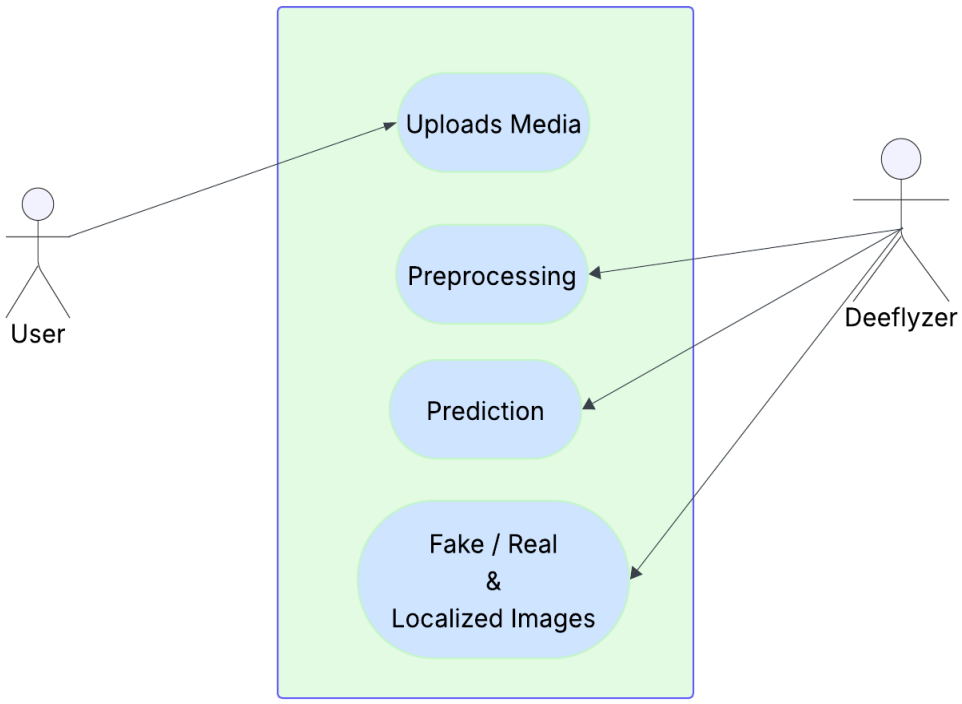


Figure 6.1: Use Case Diagram

The use case diagram illustrates the primary interactions between users and the hybrid deepfake detection system. It includes the following use cases:

- **Uploads Media:** The user initiates the process by uploading either an audio or video file. This is the entry point into the system, enabling further processing and analysis.
- **Preprocessing:** Once media is uploaded, the system performs preprocessing. For videos, this includes frame extraction and face detection, and for audio, it includes normalization and segmentation.
- **Prediction:** After preprocessing, the system proceeds with deepfake detection using trained models InceptionV3 for visual content and Wav2Vec 2.0 for audio content. This stage determines the authenticity of the media.
- **Fake / Real & Localized Images:** The results are then displayed to the user. This includes the classification result (Real or Fake) and visualizations such as localized manipulated regions (in the case of fake media).

There are two actors in the system:

- **User:** Initiates the system by uploading media and receives the detection result.
- **System:** Responsible for executing all internal processes including preprocessing, feature extraction, classification, and result visualization.

6.4 DATA MODEL AND DESCRIPTION

6.4.1 Data Description

- **Video File**
  - **Description:** Input video file to be analyzed for deepfake detection.
  - **Attributes:** File Name, Format (e.g., .mp4, .avi), Size, Duration, Frame Rate.
  - **Operations:** Frame extraction, face detection, resizing, normalization.
- **Audio File**
  - **Description:** Audio extracted from input video or provided independently for spoofing detection.
  - **Attributes:** File Name, Format (e.g., .wav, .mp3), Sample Rate, Channels.
  - **Operations:** Segmentation, normalization, feature extraction (Wav2Vec 2.0).
- **Extracted Frames**
  - **Description:** Image frames extracted from videos for spatial feature analysis.
  - **Attributes:** Frame Number, Resolution, Timestamp.
  - **Operations:** Face cropping, resizing, InceptionV3 feature extraction.
- **Feature Vectors (Video)**
  - **Description:** High-dimensional vectors extracted from InceptionV3 representing spatial features.
  - **Attributes:** Vector ID, Frame Reference, Feature Dimensions.
  - **Operations:** Passed to LSTM for temporal sequence modeling.
- **Feature Vectors (Audio)**
  - **Description:** Latent feature representations of audio generated by Wav2Vec 2.0.
  - **Attributes:** Segment ID, Temporal Encoding, Feature Dimensions.
  - **Operations:** Used by classifier to determine authenticity.

- **Prediction Results**
  - **Description:** Final output indicating whether the input media is real or deepfake.
  - **Attributes:** Media ID, Probability Score, Classification (Real/Fake), Tampered Regions.
  - **Operations:** Displayed to user, stored in logs or database.
- **Datasets**
  - **Description:** Publicly available datasets used for training/testing models.
  - **Examples:** DFDC, Celeb-DF, FaceForensics++, ASVSpoof 2019.
  - **Attributes:** Dataset Name, Number of Samples, Label Distribution (Real/Fake), Source.
  - **Operations:** Data loading, preprocessing, model training.

6.4.2 Data objects and Relationships

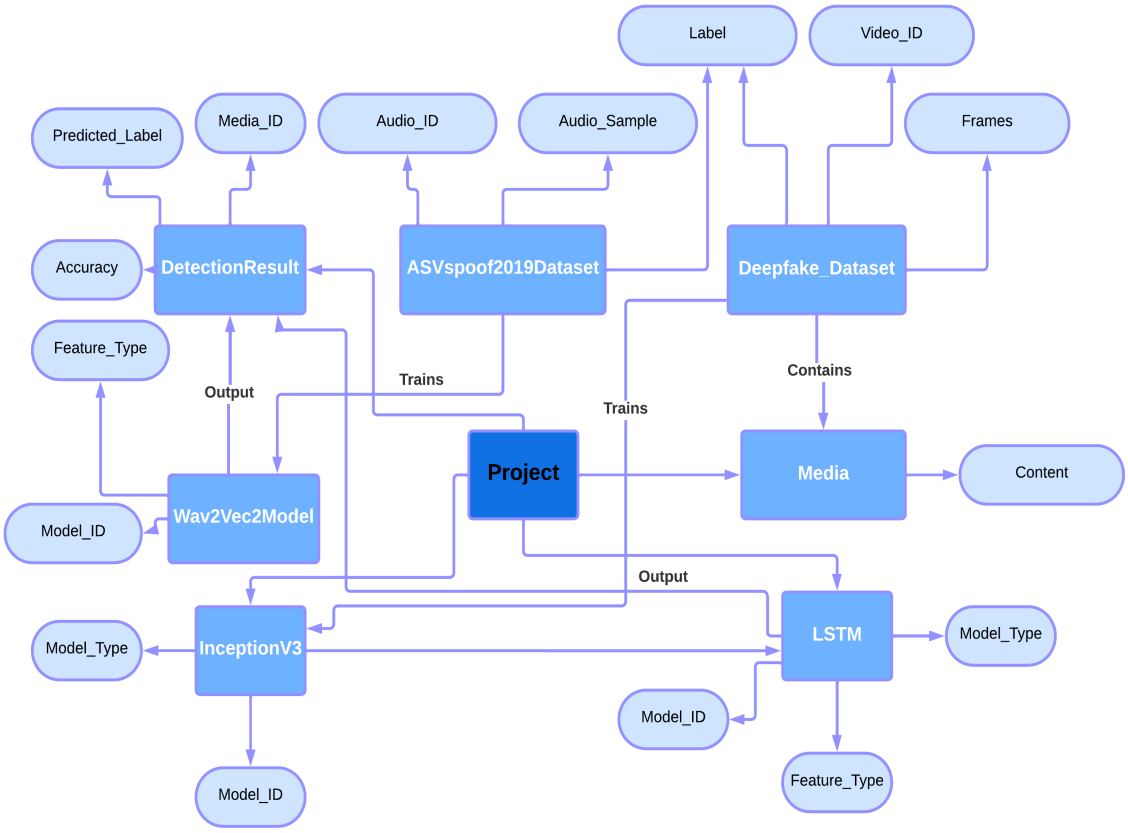


Figure 6.2: Entity Relationship Diagram

The Entity-Relationship (ER) diagram illustrates the interaction between the core components of the hybrid deepfake detection system. Below is the breakdown of each entity and its role:

- **Project:**
  - Central coordinating unit.
  - Manages training for models like `Wav2Vec2Model`, `InceptionV3`, and `LSTM`.
  - Connects datasets to model components and outputs to the detection system.
- **ASVspoof2019Dataset:**
  - Stores audio samples and identifiers.
  - Attributes: `Audio_ID`, `Audio_Sample`.
  - Used to train the audio model (`Wav2Vec2Model`).
- **Deepfake_Dataset:**
  - Contains labeled video samples and associated frames.
  - Attributes: `Video_ID`, `Frames`, `Label`.
  - Supports training of the `InceptionV3` and `LSTM` models.
- **Media:**
  - Stores and provides media content to models.
  - Linked to the `Deepfake_Dataset` via a "contains" relationship.
  - Attribute: `Content`.
- **InceptionV3:**
  - A CNN-based model for image feature extraction.
  - Outputs: `Model_ID`, `Model_Type`.
- **Wav2Vec2Model:**
  - Processes raw audio input using a transformer-based architecture.
  - Outputs: `Model_ID`, `Feature_Type`.
- **LSTM:**
  - Captures temporal dependencies in video data sequences.
  - Outputs: `Model_ID`, `Model_Type`, `Feature_Type`.

- **DetectionResult:**
  - Stores final classification results and performance metrics.
  - Attributes: `Media_ID`, `Predicted_Label`, `Accuracy`, `Feature_Type`.

6.5 FUNCTIONAL MODEL AND DESCRIPTION

6.5.1 Data Flow Diagrams

Level 0 Data Flow Diagram

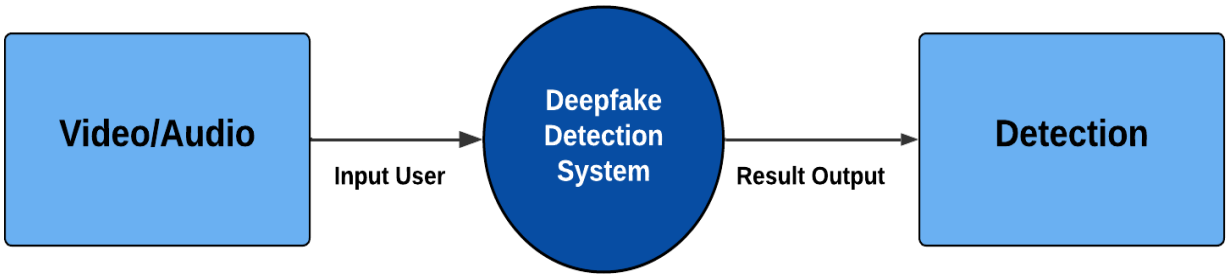


Figure 6.3: DFD Level-0

This Level-0 Data Flow Diagram (DFD) represents the most basic overview of the deepfake detection system. It includes the main components and their interaction with data. The DFD describes how the input data (video or audio) is processed by the system to generate a detection output.

- **Input:** Video/Audio media content is provided as input.
- **Process:** The Deepfake Detection System processes the input using AI/ML models to analyze and extract relevant features.
- **Output:** The system produces a final detection result indicating whether the input is real or fake.

Level 1 Data Flow Diagram

This Level-1 Data Flow Diagram provides a more detailed breakdown of the internal components of the deepfake detection system. The data flow proceeds as follows:

1. **Video/Audio Input:** The system receives raw video or audio input from the user.

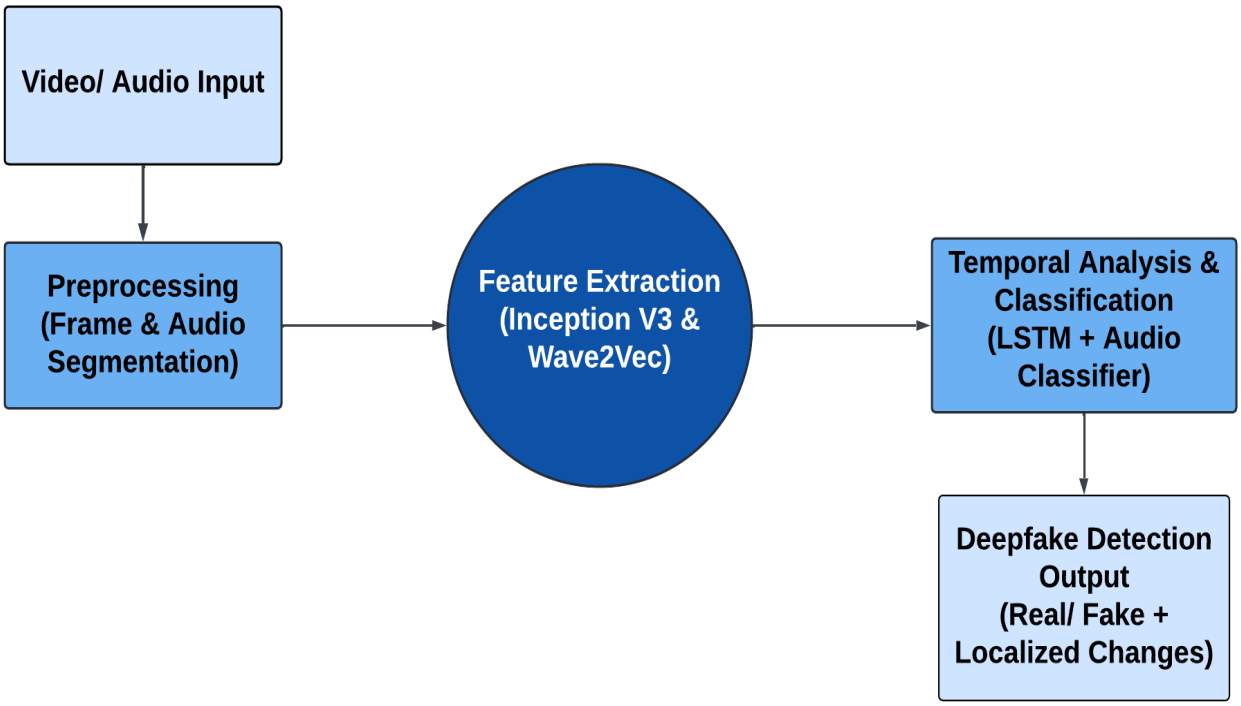


Figure 6.4: DFD Level-1

2. **Preprocessing:** This stage involves frame-wise and audio segmentation to prepare the data for feature extraction.

3. **Feature Extraction:** In this stage, visual features are extracted using Inception V3 and audio features using Wave2Vec.

4. **Temporal Analysis & Classification:** An LSTM network and an audio classifier analyze sequential dependencies and classify the content as real or fake.

5. **Output:** The system returns whether the input is real or fake, along with any localized changes or anomalies.

Level 2 Data Flow Diagram

This Level-2 Data Flow Diagram illustrates the detailed components and processing flow of the hybrid deepfake detection system. The system integrates both visual and audio-based analysis for improved accuracy. The processing steps are as follows:

1. **Frame Extraction:** Video input is processed to extract individual frames.

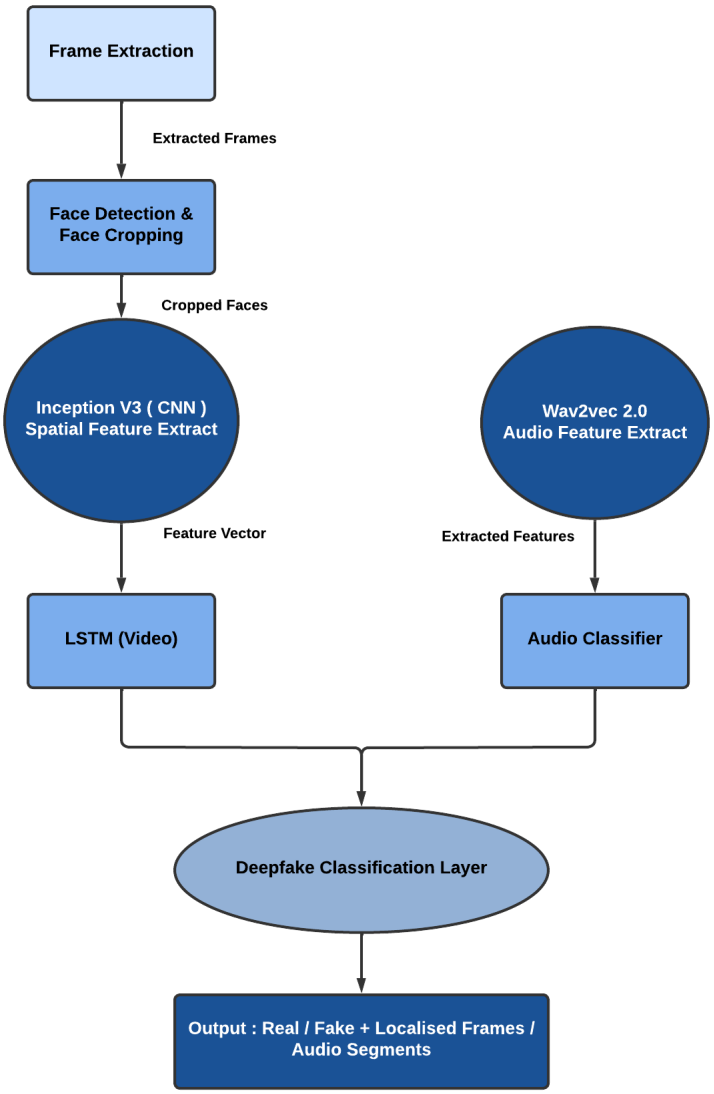


Figure 6.5: DFD Level-2

2. **Face Detection and Cropping:** Each frame undergoes face detection using MTCNN and the faces are cropped for focused analysis.

3. **Inception V3 - Spatial Feature Extraction:** Cropped face images are passed through a Convolutional Neural Network (Inception V3) to extract spatial features.

4. **LSTM (Video):** The sequence of features is processed by a Long Short-Term Memory (LSTM) network to learn temporal dependencies in video frames.

5. **Wav2Vec 2.0 - Audio Feature Extraction:** The audio stream is segmented and processed using Wav2Vec 2.0 to obtain deep speech embeddings.

6. **Audio Classifier:** Extracted features are classified as real or fake speech using a fine-tuned

audio classifier.

7. **Deepfake Classification Layer:** Outputs from both LSTM (video) and audio classifier are combined to make a final classification.

8. **Output:** The system provides a binary classification (Real/Fake) and localized visual or audio segments suspected to be manipulated.

6.5.2 Activity Diagram

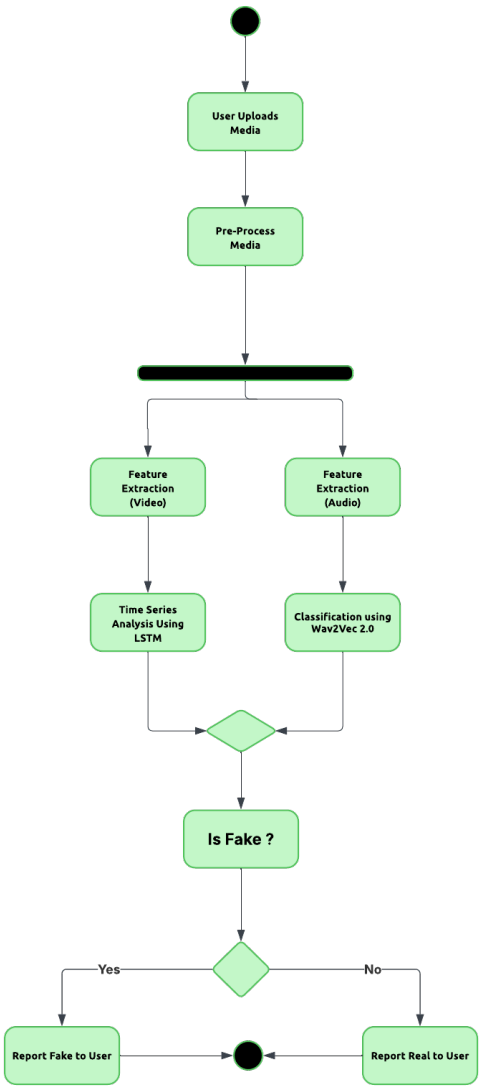


Figure 6.6: Activity Diagram

The UML activity diagram illustrates the workflow of a deepfake detection system from user input to final classification results. Below is the description of each component:

- **Start:** This is the initial state, marking the beginning of the process.
- **User Uploads Media:** The user uploads an audio or video file to be analyzed for authenticity.
- **Pre-Process Media:** The uploaded media is pre-processed to prepare it for analysis. This may include steps like frame extraction for video or audio sample normalization.
- **Parallel Feature Extraction:**
  - **Feature Extraction (Video):** For video files, the system uses deep learning models to extract relevant features from the frames.
  - **Feature Extraction (Audio):** For audio files, the system extracts audio features using a model like Wav2Vec 2.0.
- **Time Series Analysis Using LSTM (for Video):** For video input, the LSTM model performs time-series analysis on the extracted video features to understand temporal relationships between frames.
- **Classification Using Wav2Vec 2.0 (for Audio):** For audio input, the Wav2Vec 2.0 model performs classification based on extracted features to detect if the audio is real or fake.
- **Is Fake? (Decision):** This decision node checks the classification results to determine if the media is detected as fake or real.
- **Report Fake to User:** If the media is classified as fake, a report is generated and sent back to the user, indicating that the media is fake.
- **Report Real to User:** If the media is classified as real, the system reports back to the user that the media is authentic.
- **End:** This final state marks the end of the process, concluding the detection workflow.

6.5.3 State Diagram

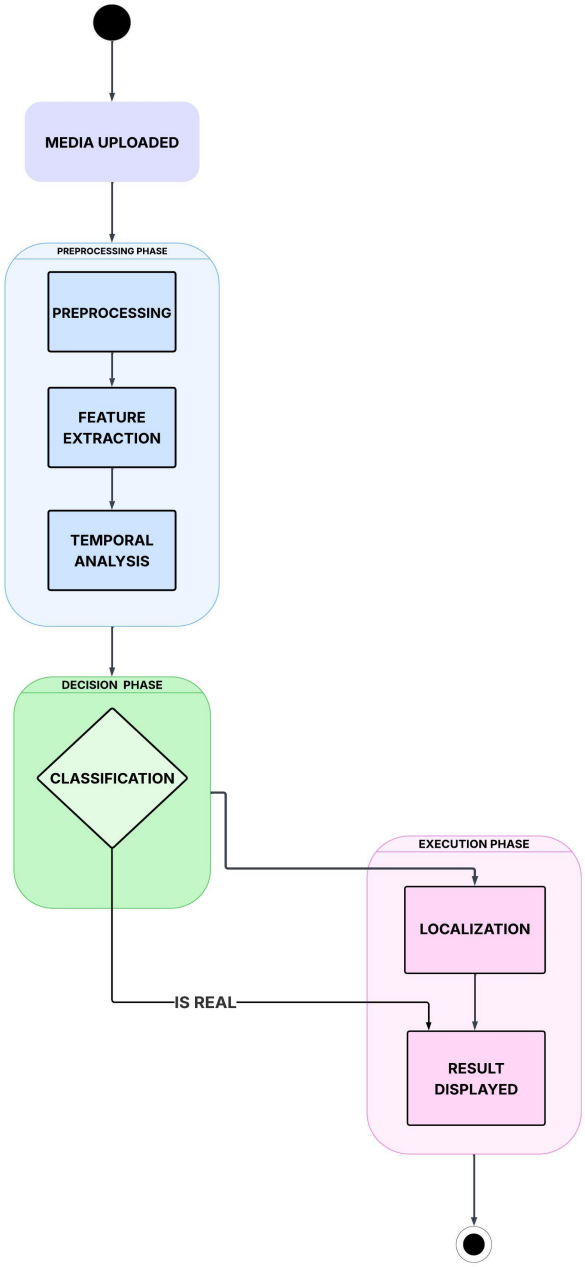


Figure 6.7: State Machine Diagram

The state machine diagram describes the operational flow of the hybrid deepfake detection system. It defines the transition of system states from media input to the final result display. The following steps illustrate the process:

- **Media Uploaded:** The system begins when the user uploads media (either audio or video). This triggers the start of the detection process.
- **Preprocessing Phase:** This phase is responsible for preparing the media data for feature

extraction. It includes:

  - *Preprocessing:* Audio is normalized and segmented, and video frames are extracted and cropped for further processing.
  - *Feature Extraction:* Key features are extracted using InceptionV3 for video and Wav2Vec 2.0 for audio.
  - *Temporal Analysis:* Sequential dependencies within the data are analyzed using LSTM for video sequences.

- **Decision Phase:**
  - *Classification:* A decision node classifies the input media as either real or fake based on the extracted and analyzed features.
- **Execution Phase:** Depending on the classification output, the system proceeds to:
  - *Localization:* If the input is fake, the regions of manipulation are localized visually or temporally.
  - *Result Displayed:* The final output is rendered to the user, indicating whether the media is real or fake, and in the case of fake media, highlighting the tampered regions.
- **End:** The system reaches the final state after displaying the result.

6.5.4 Sequence Diagram

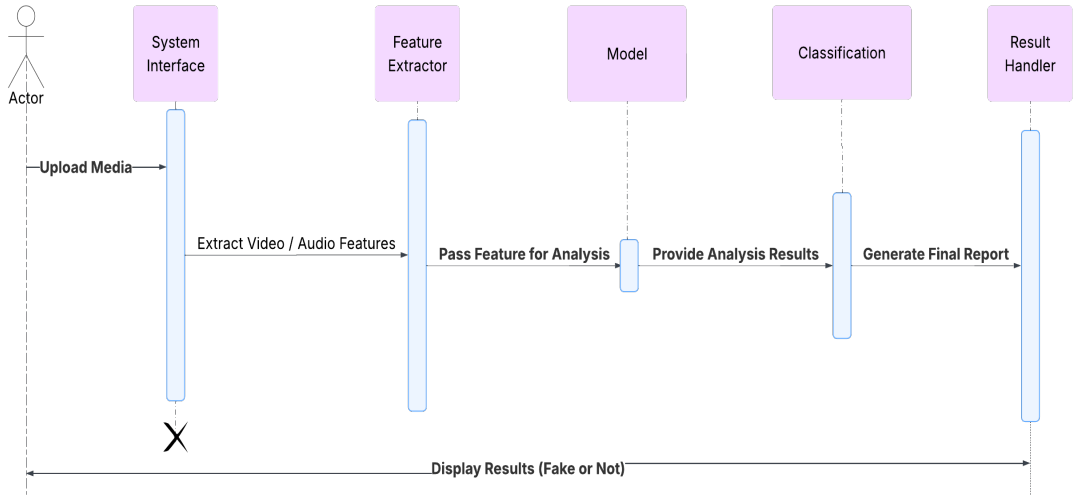


Figure 6.8: Sequence Diagram

- **User Interaction:** The user initiates the process by uploading media (audio or video) to the System Interface.
- **System Interface Processing:** The System Interface receives the media and passes it to the Feature Extractor for processing.
- **Feature Extraction:** The Feature Extractor processes the media and extracts essential features such as video frames for visual content or audio samples for audio content.
- **Model Processing:** The extracted features are then processed using deep learning models:
  - InceptionV3 is used for spatial analysis of video content.
  - Wav2Vec 2.0 is used for audio signal representation and analysis.

  The model generates predictions which are passed to the Classification component.
- **Classification:** The Classification component analyzes the outputs from the model and classifies the input media as either real or fake.
- **Result Handling and Output:** The Result Handler receives the classification result and conveys it back to the user through the System Interface.

6.5.5 Design Constraints

- **Hardware Limitations:**

  - Deep learning models like InceptionV3, LSTM, and Wav2Vec 2.0 require high computational resources for training and inference.
  - GPU acceleration is essential for efficient model training and real-time deepfake detection.
  - Limited memory availability may restrict batch size or input resolution.

- **Software Dependencies:**
  - The system depends on Python libraries such as TensorFlow, PyTorch, OpenCV, NumPy, and Librosa.
  - Framework versions must be compatible and properly configured in the deployment environment.
  - Operating system compatibility must be ensured (preferably Linux-based for deployment).

- **Dataset Constraints:**
  - Public datasets (e.g., DFDC, ASVSpoof 2019) may contain noise, bias, or imbalance which could affect model performance.
  - Large datasets require significant storage and preprocessing time.
  - Label accuracy and authenticity must be validated for effective training.

- **Real-Time Processing Requirements:**
  - The system must process media files and return predictions within an acceptable timeframe for real-time or near-real-time applications.
  - Latency must be minimized to support use cases such as live verification or forensic scanning.

6.5.6 Software Interface Description

User Interface

- The system provides a web-based graphical user interface (GUI) for interaction.
- Users can upload video or audio files, initiate detection, and view results including probability scores and localization heatmaps.
- The interface includes buttons for file upload, media playback, and viewing tampered regions.

- Designed using HTML, CSS, JavaScript (optionally React.js or Flask with Jinja2 for backend rendering).

Model Interface

- The detection models (InceptionV3 + LSTM and Wav2Vec 2.0) are loaded and exposed via a Python-based backend.
- Communication occurs through internal APIs or function calls from the Flask/Django server.
- Input: Preprocessed frame or audio data; Output: Binary classification (Real/Fake), probability score, and localization map.

Dataset Interface

- Public datasets such as DFDC, FaceForensics++, Celeb-DF, and ASVSpoof 2019 are integrated for training and testing.
- Data is accessed from local storage or mounted volumes and read using Python libraries (e.g., OpenCV, Pandas, Librosa).
- Interface supports batch loading, preprocessing (resizing, normalization), and augmentation during training.

System/Hardware Interface

- The system requires GPU support for efficient model training and inference (e.g., NVIDIA CUDA-compatible devices).
- Interfaces with camera and microphone devices may be included for live deepfake detection scenarios.
- Hardware drivers and OS-level configurations must support multimedia processing (FFmpeg, ALSA for audio).

Network Interface

- The system operates on a local network or over the internet when deployed as a web application.
- HTTPS is recommended for secure transmission of media and results.
- May require firewall and proxy configurations for integration with enterprise environments or cloud deployment.

6.6 Nonfunctional Requirements

6.6.1 Performance Requirements

- The system shall process and classify uploaded video files within 20-30 seconds, depending on file size and system specifications.
- The audio deepfake detection shall provide results within 10-15 seconds for standard-length audio clips.
- The application shall maintain responsiveness during classification by using asynchronous processing or GPU acceleration if available.
- The system shall handle concurrent requests from multiple users when deployed on a server.

6.6.2 Safety Requirements

- The system shall ensure no personal data or uploaded media files are stored permanently.
- Temporary files such as frames and face images shall be automatically deleted after classification is completed.
- The application shall notify users of unsupported or potentially corrupted files to prevent misinterpretation.

6.6.3 Security Requirements

- The system shall restrict access to uploaded files and results to the uploading user only in multi-user deployments.
- All file uploads shall be validated to prevent malicious content from being executed or saved on the server.
- For web deployments, HTTPS shall be used to secure data transfer between client and server.
- API tokens used for accessing pretrained models (e.g., Hugging Face) shall be stored securely and not exposed in the frontend.

6.6.4 Software Quality Attributes

- **Usability:** The interface shall be intuitive, with clear options and outputs for all types of users.

- **Reliability:** The system shall provide consistent results across different sessions and media inputs.
- **Maintainability:** The modular architecture shall allow easy updates or replacement of detection models.
- **Portability:** The application shall run on major operating systems including Windows, Linux, and macOS.
- **Scalability:** The system shall be capable of being scaled to handle high volumes of media files when deployed on a server.

Chapter 7

Detailed Design Document using Appendix A and B

7.1 Introduction

This document specifies the design architecture and implementation strategy used to develop the **Deeflyzer** system - a hybrid deepfake detection framework for analyzing and identifying manipulated media. The primary goal of this project is to provide a robust solution that addresses the growing threat of deepfakes in digital content, especially those involving human faces and cloned voices.

Deepfakes are synthetic media generated using deep learning techniques, capable of convincingly replacing faces, altering speech, or fabricating events that never occurred. These manipulations pose serious risks to personal privacy, political stability, and information credibility. As such, detecting deepfakes with high accuracy and reliability is a pressing concern.

The proposed system combines multiple deep learning models in a hybrid architecture:

- **InceptionV3 (CNN)** for spatial feature extraction from video frames.
- **LSTM (RNN)** for modeling temporal inconsistencies in frame sequences.
- **Wav2Vec 2.0** for extracting latent audio features to detect AI-generated speech.

This document outlines the design methodology, system architecture, data flow, component interaction, and the deployment strategy used in the development of Deeflyzer. It includes detailed specifications of the preprocessing pipeline, model training phases, evaluation metrics, and user interface design. The goal is to ensure a scalable and efficient system capable of detecting deepfakes in both audio and video formats, with localization of tampered regions.

7.2 Architectural Design

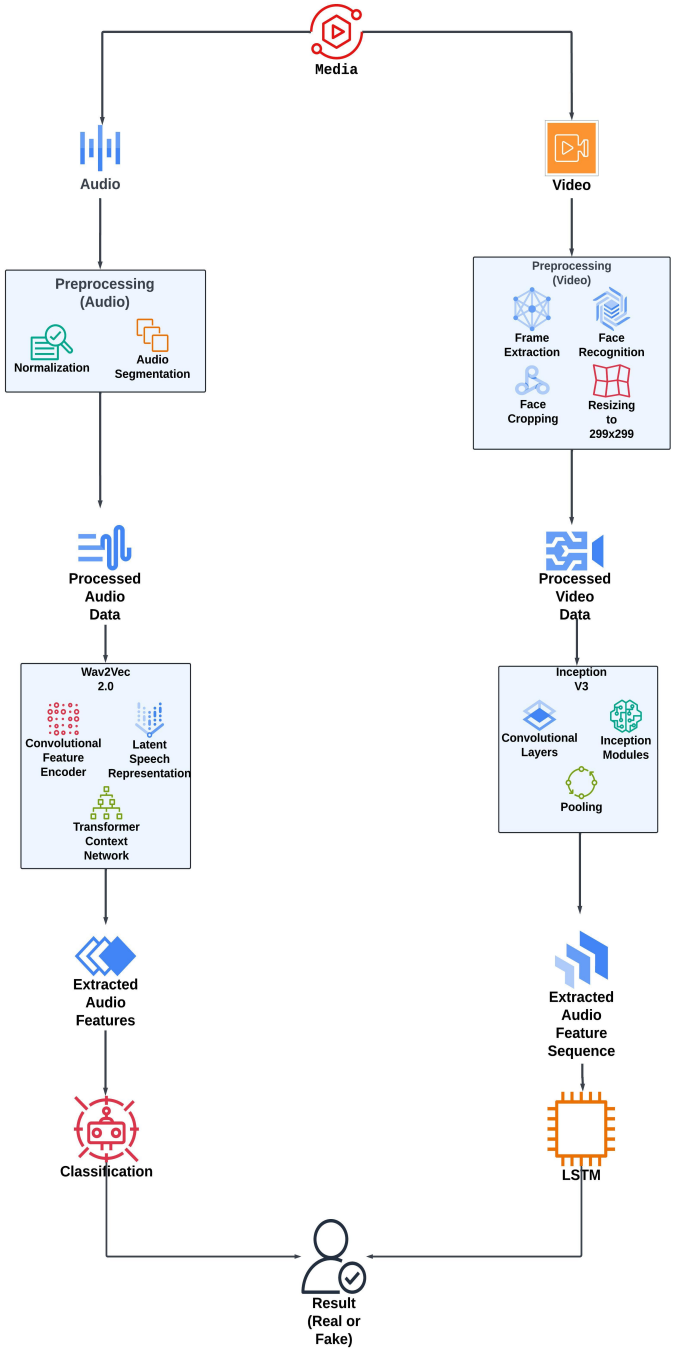


Figure 7.1: Architectural Design

1. **Media Input**
   - The system accepts a media file containing both **audio** and **video** components.
2. **Audio and Video Separation**

- The media file is split into:
  - **Audio**
  - **Video**

3. **Audio Pipeline**

   (a) **Audio Preprocessing**

   - **Normalization**: Standardizes the audio volume.
   - **Audio Segmentation**: Splits audio into manageable chunks.

   (b) **Audio Feature Extraction using Wav2Vec 2.0**

   - **Convolutional Feature Encoder**: Captures low-level acoustic features.
   - **Latent Speech Representation**: Encodes speech into vector representations.
   - **Transformer Context Network**: Understands temporal relationships in audio.

   (c) **Audio Classification**

   - The extracted features are classified as **Real** or **Fake**.

4. **Video Pipeline**

   (a) **Video Preprocessing**

   - **Frame Extraction**: Extracts individual frames from video.
   - **Face Recognition**: Detects faces in frames.
   - **Face Cropping**: Crops detected face regions.
   - **Resizing**: Resizes faces to 299×299 pixels.

   (b) **Video Feature Extraction using InceptionV3**

   - **Convolutional Layers**: Learn spatial features from faces.
   - **Inception Modules**: Capture multi-scale details.
   - **Pooling Layers**: Reduce dimensionality and suppress noise.

   (c) **Temporal Feature Learning (LSTM)**

   - An LSTM processes the sequential frame features to detect temporal inconsistencies.

   (d) **Video Classification**

   - The LSTM output is classified as **Real** or **Fake**.

5. **Final Result Integration**

   - Outputs from both audio and video classifiers are fused.
   - The final result is presented to the user: **Real** or **Fake**.

7.3 Data design (using Appendices A and B)

This section outlines the data structures, database schema, and file formats used in the Deeflyzer system for deepfake detection.

7.3.1 Internal Software Data Structures

This subsection outlines the internal data structures used by the Deeflyzer system and passed between its major software components. These structures support processing workflows for both video and audio deepfake detection.

- `FrameMatrix[299][299][3]`

  A 3D array representing a single RGB video frame of size 299x299 pixels, passed from the preprocessing module to the InceptionV3 feature extractor.

- `AudioSegment[n]`

  A 1D array containing a normalized raw waveform of segmented audio, where n is the number of samples. This is passed from the audio preprocessing module to the Wav2Vec 2.0 extractor.

- `FeatureVector[n]`

  A 1D array of extracted features (e.g., embeddings) from either InceptionV3 (video) or Wav2Vec 2.0 (audio). These are passed to the LSTM temporal analyzer or the final classifier layer.

- `TemporalSequence[m][n]`

  A 2D matrix where each row is a feature vector from one video frame; m is the number of frames, and n is the length of the feature vector. Passed from feature extraction to the LSTM module for sequential pattern analysis.

- `PredictionResult`

  A dictionary-like structure passed from the classification module to the frontend or output renderer. It contains:

  - `label`: Real or Fake
  - `confidence`: float (0.0 to 1.0)
  - `media_type`: "audio" or "video"

- `LocalizationMap`

  A 2D binary mask (for video frames) or a list of timestamp intervals (for audio) indicating

regions detected as tampered. Passed from the localization module to the frontend for visualization.

- `ProcessedMedia`

  A wrapper object that aggregates cleaned and preprocessed data ready for analysis. Contains:

  - `frames: list[FrameMatrix]`
  - `audio: list[AudioSegment]`
  - `meta_info: dict` (filename, duration, etc.)

These data structures ensure modularity, consistency, and reusability across the video and audio pipelines of the system.

7.3.2 Global data structure

This subsection outlines the global data structures that are accessible to multiple components of the Deeflyzer system. These structures facilitate shared configurations, model references, and standardized labels in both audio and video deepfake detection pipelines.

- `ModelRegistry`

  A centralized dictionary that maps the names of the models to their corresponding file paths or loaded instances. It is accessible to all inference components.

  - `InceptionV3`: path or model instance
  - `LSTM`: path or model instance
  - `Wav2Vec2`: path or model instance

- `LabelMap`

  A globally accessible dictionary used for the interpretation of classification output.

  - `0: ''Real''`
  - `1: ''Fake''`

  Used in evaluation, recording, and presentation of results.

- `ConfigParams`

  A dictionary of key system parameters passed across modules for consistent pre-processing and model behavior.

  - `frame_size = (299, 299)`

  - `sampling_rate = 16000`
  - `confidence_threshold = 0.5`

- `Logger`

  A global logging object for writing run-time events, errors, and evaluation metrics. Accessible by all major modules (preprocessing, inference, back-end).

- `DeviceContext`

  A global hardware configuration object used to set the processing context (CPU/GPU).

  - `device = ‘‘cuda’’` or `‘‘cpu’’`
  - Used during model loading and inference.

- `ResultCache`

  A global in-memory structure used for temporarily storing intermediate predictions and localization results, especially in batch or multimodal inference pipelines.

These global structures enable loose coupling between modules and centralized configuration control, ensuring the scalability and modularity of the system architecture.

7.3.3 Temporary data structure

This subsection outlines the temporary data structures used during the execution of the Deeflyzer system. These structures are created dynamically, used during intermediate computation stages, and typically discarded after inference or training. They facilitate modular processing and efficient pipeline execution.

- `TempFrameStorage`

  A temporary list or directory structure holding extracted video frames during preprocessing.

  - Format: `List[FrameMatrix]` or `/temp/frames/frame_001.jpg`
  - Lifecycle: Created during frame extraction; deleted after feature extraction.

- `TempAudioChunks`

  Segmented audio data temporarily stored for batch processing with the Wav2Vec 2.0 model.

  - Format: `List[AudioSegment[n]]`
  - Lifecycle: Exists during audio pre-processing and is discarded after feature extraction.

- `IntermediateFeatures`

  A buffer storing feature vectors produced by CNN or audio encoder, passed to the LSTM or classification layer.

  - Format: `List[FeatureVector]`
  - Used for temporal analysis and ensemble classification.

- `InferenceBatchBuffer`
  Stores a batch of video/audio inputs queued for inference in streaming or batch mode.
  - Format: `Dict{input_id: input_tensor}`
  - Used by inference engine for temporary batch classification.

- `TempResultHolder`
  A temporary dictionary storing classification results before being committed to the final output format or logs.
  - Keys: `label, confidence, localized_frames`
  - Cleared after rendering or output delivery.

- `VisualizationOverlay`
  A temporary structure used to hold graphical masks for tampered region highlighting before final rendering.
  - Format: `2D array or RGBA overlay`
  - Used only during result visualization.

These temporary structures play a critical role in managing pipeline stages, optimizing memory usage, and ensuring that intermediate computation does not interfere with persistent data storage.

7.3.4 File Formats and Data Sources

- **Input File Formats**
  - Video: `.mp4`, `.avi`, `.mov`
  - Audio: `.wav`, `.mp3`

- **Model Files**
  - InceptionV3: `inception_v3.h5` or `.pth`
  - LSTM: `lstm_temporal.pt`
  - Wav2Vec 2.0: `wav2vec_model.bin`

- **Dataset Sources**
  - `DFDC`, `FaceForensics++`, `Celeb-DF` (video)
  - `ASVSpoof 2019`, `FakeAVCeleb` (audio)

7.3.5 Output Formats

- Prediction result: JSON
  - {"label": "Fake", "confidence": 0.963, "localization": [frame_coords/audio_ti
- Visualization: Rendered image/video/audio highlighting tampered segments

7.4 Compnent Design

7.4.1 Class Diagram

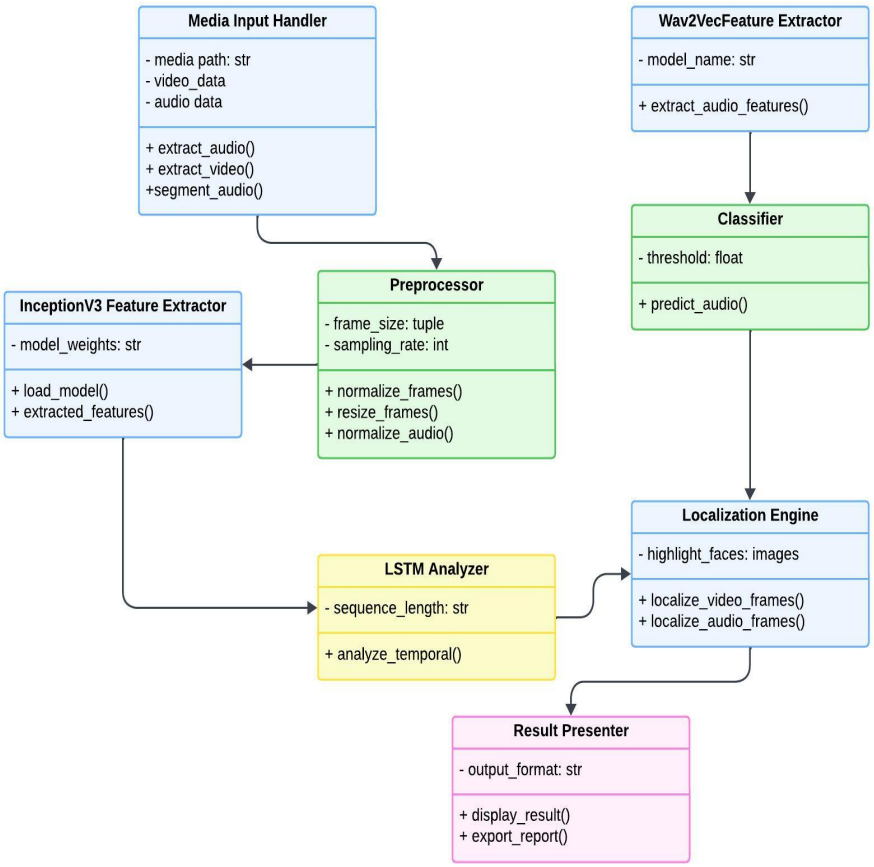


Figure 7.2: Class Diagram

The class diagram illustrates the architecture of the deepfake detection system using object-oriented design. It includes the following classes and their key responsibilities:

- **Media Input Handler**

  Responsible for handling media input such as audio and video files. It provides methods to extract video frames, extract audio, and segment audio samples.

  Attributes:

  - media_path: str
  - video_data
  - audio_data

  Methods:

  - extract_audio()
  - extract_video()
  - segment_audio()

- **Preprocessor**

  Manages preprocessing of audio and video data, including normalization and resizing.

  Attributes:

  - frame_size: tuple
  - sampling_rate: int

  Methods:

  - normalize_frames()
  - resize_frames()
  - normalize_audio()

- **InceptionV3 Feature Extractor**

  Loads the InceptionV3 model and extracts features from video frames.

  Attributes:

  - model_weights: str

  Methods:

  - load_model()
  - extracted_features()

- **Wav2VecFeature Extractor**

  Extracts audio features using a pre-trained Wav2Vec2.0 model.

  Attributes:

  - model_name: str

  Methods:

  - extract_audio_features()

- **Classifier**

  Predicts whether the input is real or fake based on the extracted features.

  Attributes:

  - threshold: float

  Methods:

  - predict_audio()

- **Localization Engine**

  Localizes the regions in video frames or audio segments that indicate manipulation.

  Attributes:

  - highlight_faces: images

  Methods:

  - localize_video_frames()
  - localize_audio_frames()

- **LSTM Analyzer**

  Performs temporal analysis on video features using LSTM networks.

  Attributes:

  - sequence_length: str

  Methods:

  - analyze_temporal()

- **Result Presenter**

  Displays or exports the final classification result to the user.

  Attributes:

  - output_format: str

  Methods:

  - display_result()
  - export_report()

Chapter 8

Project Implementation

8.1 Introduction

The implementation phase focuses on translating the design of the hybrid deepfake detection system into a functional and executable solution. This phase involves integrating various components such as data preprocessing, model inference, audio-video classification, and user interface development. The system is developed using Python, leveraging powerful libraries like TensorFlow and PyTorch to build and execute deep learning models. The implementation is divided into modular components to enhance maintainability and scalability. These include video frame extraction, face detection using MTCNN, feature extraction with InceptionV3, temporal analysis using LSTM, and audio classification using Wav2Vec 2.0. Each component has been tested individually and then integrated to ensure seamless workflow from media input to classification output. The system is further supported by a graphical user interface developed using NextJS and FastAPI, allowing users to upload media files and receive real-time feedback on the authenticity of the content. Special care has been taken to optimize inference speed and minimize memory usage to ensure efficient performance even on moderately configured systems.

8.2 Tools and Technologies Used

The implementation of the hybrid deepfake detection system leverages a wide range of tools, frameworks, and technologies to ensure robustness, accuracy, and user accessibility. The key tools and technologies used are:

- **Python 3.8+**: The primary programming language used for developing the entire system, chosen for its extensive libraries and community support in machine learning and deep learning.
- **TensorFlow 2.x and Keras**: Utilized for constructing and deploying the video-based deepfake detection model using InceptionV3 and LSTM layers for spatial and temporal analysis.
- **PyTorch and HuggingFace Transformers**: Employed for implementing the Wav2Vec 2.0 model, which is used for audio-based deepfake detection through fine-tuned transformer-based classification.
- **OpenCV**: Used for extracting frames from input videos and handling image operations such as resizing and format conversion.
- **MTCNN (Multi-task Cascaded Convolutional Neural Network)**: Applied for face detection in video frames to isolate and analyze facial regions, which are critical for deepfake

classification.

- **Torchaudio and Pydub**: Utilized for loading, converting, and preprocessing audio files to ensure compatibility with the Wav2Vec 2.0 model.
- **NextJS**: These frameworks are used for developing the web-based user interface, allowing users to upload media, run predictions, and view results in an interactive environment.
- **NumPy and Pandas**: For handling numerical operations, data manipulation, and intermediate result storage during preprocessing and inference.
- **Google Colab / Jupyter Notebook**: Used during development and testing phases for model prototyping, experimentation, and performance evaluation.
- **CUDA and cuDNN (optional)**: Employed for GPU acceleration to reduce training and inference time when running models on compatible hardware.

8.3 Methodologies/Algorithm Details

The implementation of the Hybrid Deepfake Detection System is structured into multiple modules, each using specialized technologies and deep learning models for detecting fake content in both audio and video streams. The following methodology outlines the core components and their respective workflows:

- **Frame Extraction and Face Detection:** The system extracts frames from uploaded video files using OpenCV. These frames are passed to the MTCNN (Multi-task Cascaded Convolutional Networks) detector to locate and crop human faces with high precision. This helps in isolating regions of interest from each frame for analysis.
- **Image-Based Deepfake Detection using InceptionV3 + LSTM:** Cropped face images are preprocessed (resized to 224×224 and normalized) and passed through a hybrid model combining InceptionV3 and LSTM. InceptionV3 is a pre-trained convolutional neural network that extracts deep spatial features from images. These features are then passed to a Long Short-Term Memory (LSTM) network to analyze temporal consistency across frames. The final classification is performed using a sigmoid-activated dense layer to determine if the video is real or fake.
- **Audio Deepfake Detection using Wav2Vec 2.0:** For audio-based inputs, the system uses Wav2Vec 2.0, a self-supervised model by Facebook AI, fine-tuned for binary classification. Audio inputs are converted to 16kHz mono format and processed into feature

embeddings. The model then predicts whether the audio is genuine or synthetically generated. It is particularly effective against voice cloning and text-to-speech attacks.

- **GUI Interface:** The system provides a user-friendly interface using NextJS for users to upload media files and view the classification results. Detected fake faces are visually displayed on the interface for transparency.
- **Integration and Execution:** Python serves as the main implementation language, with TensorFlow used for the video model and PyTorch for the audio model. All components are integrated seamlessly in the GUI for real-time analysis.

8.3.1 Working of the System

The Hybrid Deepfake Detection System is designed to analyze both audio and video files to determine their authenticity. The step-by-step workflow is outlined below:

1. **Media Upload:** The user begins by uploading a media file through the provided interface. The system supports both video and audio formats.
2. **Preprocessing:**
   - **For Video:** The system extracts a fixed number of frames from the uploaded video using OpenCV. Each frame is saved temporarily for further analysis.
   - **For Audio:** The system converts the uploaded audio to a standard format (.wav, 16kHz, mono) using Pydub and Torchaudio if it is not already in the required format.
3. **Face Detection (Video Only):** MTCNN is used to detect and crop human faces from each video frame. These cropped face images form the dataset for the video-based deepfake classification model.
4. **Feature Extraction and Classification:**
   - **Video-Based Detection:** The face images are passed through the InceptionV3 model to extract spatial features. These features are then sequentially fed into an LSTM layer, which captures temporal dependencies between frames. The final output is a classification score indicating whether the video is real or fake.
   - **Audio-Based Detection:** The converted audio waveform is passed through the Wav2Vec 2.0 processor to generate embeddings. These embeddings are classified by a fine-tuned Wav2Vec 2.0 model to determine if the voice is authentic or synthetically generated.

5. **Result Evaluation:**
   - If a majority of frames or audio segments are labeled as "Fake", the media is flagged as deepfake.
   - The number of real and fake detections is displayed to the user.
6. **Output Display:** The final result is shown to the user through the graphical interface, along with thumbnails of any detected fake faces in the case of video input.
7. **Post-Processing:** Temporary storage folders for frames and faces are cleaned up automatically after classification to maintain disk space and user privacy.

8.3.2 Algorithm 1/Pseudo Code

This algorithm outlines the step-by-step procedure followed by the hybrid deepfake detection system, which processes either video or audio input and classifies it as real or fake.

1. **Input:** User uploads a media file (either a video or an audio file).
2. **Check Media Type:**
   - If video, proceed with video analysis steps.
   - If audio, proceed with audio analysis steps.
3. **For Video Input:**
   (a) Extract key frames using OpenCV.
   (b) Detect faces in each frame using MTCNN.
   (c) Crop and resize the detected faces to 224×224 pixels.
   (d) Pass each face image through the InceptionV3 model for feature extraction.
   (e) Feed the extracted features into an LSTM network to capture temporal dependencies.
   (f) Apply a dense layer to classify the video as `Real` or `Fake`.
4. **For Audio Input:**
   (a) Convert to WAV format if necessary using Pydub.
   (b) Resample the audio to 16kHz using Torchaudio.
   (c) Extract features using the Wav2Vec 2.0 processor.
   (d) Use the Wav2Vec 2.0 classifier to determine authenticity.

5. **Output Results:**
    - Display the classification result to the user.
    - If video is classified as fake, also display the faces identified as fake.

Description

- **Step 1: Input Handling** — Accept audio or video files via a web interface.
- **Step 2: Preprocessing** — Extract frames and detect faces from videos or resample and normalize audio inputs.
- **Step 3: Feature Extraction** — Use InceptionV3 to obtain spatial features from faces and Wav2Vec 2.0 for audio embeddings.
- **Step 4: Classification** — LSTM is applied on video features to capture sequential manipulation; audio is classified directly via transformer outputs.
- **Step 5: Result Aggregation** — Results are displayed in realtime, and if fake faces are detected, they are visualized for user verification.

8.4 Verification and Validation For Acceptance

Verification and validation (V&V) are crucial processes to ensure that the system meets the specified requirements and performs as intended under real-world conditions. For the hybrid deepfake detection system, both processes were thoroughly applied at different development stages to ensure accuracy, usability, and reliability.

Verification

Verification focuses on ensuring that the system has been built correctly in accordance with the design and functional specifications.

- **Module Testing:** Each component such as frame extraction, face detection, InceptionV3+LSTM classification, and Wav2Vec 2.0 audio analysis was tested individually.
- **Integration Testing:** Modules were integrated and tested to verify smooth communication and data flow across components.
- **Interface Testing:** The user interface was verified for correct input-output interaction, error handling, and responsiveness.

- **Model Verification:** Pre-trained models were loaded and tested on known samples to verify expected behavior before deployment.

Validation

Validation ensures that the developed system fulfills the user needs and operates effectively in real-world scenarios.

- **Functional Validation:** The system was tested with diverse audio and video deepfake inputs to ensure that both modalities are accurately classified.
- **Dataset Evaluation:** The system was validated using benchmark datasets such as DFDC, Celeb-DF, and custom datasets to check its generalizability.
- **Performance Metrics:** Metrics like accuracy, precision, recall, and F1-score were calculated to validate the modelâs effectiveness.
- **User Feedback:** A sample group of users tested the interface and classification results to ensure the system is intuitive and informative.

The system passed all validation and verification criteria set at the beginning of the project, making it acceptable for the intended application domain of detecting deepfake media.

Chapter 9

Software Testing

9.1 Types of Testing

Testing is a critical phase in the development of the Hybrid Deepfake Detection System. It ensures that each component functions correctly in isolation and when integrated with others. The following testing techniques were employed:

9.1.1 Unit Testing

Unit testing was performed on individual components to verify their correctness in isolation. Some of the key units tested include:

- **Frame Extraction Module:** Verified that exactly N frames are extracted from videos and saved correctly.
- **Face Detection Module:** Tested with various frame inputs to ensure MTCNN reliably detects and crops facial regions.
- **Audio Preprocessing Module:** Checked the conversion of different audio formats to 16kHz mono WAV files.
- **Model Inference:** Tested the InceptionV3 + LSTM pipeline and Wav2Vec 2.0 independently with dummy inputs.

9.1.2 Integration Testing

Integration testing ensured that interconnected modules passed data correctly and functioned as a unified system. Key tests included:

- Feeding extracted faces into the InceptionV3-LSTM pipeline for real-time classification.
- Integration of video and audio classifiers into the NextJS user interface.
- Verifying that output from classification modules is correctly displayed in the GUI.

9.1.3 System Testing

System testing validated the entire end-to-end workflow, including:

- Uploading media files (video/audio) through the GUI.
- Preprocessing, classification, and result visualization.
- Detection of fake and real content and appropriate message/report generation.
- Cross-platform compatibility and responsiveness of the application.

9.1.4 Performance Testing

Performance testing was conducted to evaluate responsiveness, accuracy, and resource consumption:

- **Inference Time:** Measured the average time taken to process and classify media inputs.
- **Accuracy:** Evaluated using benchmark datasets such as DFDC and Celeb-DF.
- **Scalability:** Tested with multiple sequential inputs to evaluate system stability.
- **Memory Usage:** Profiled memory consumption during peak load using video and audio simultaneously.

9.1.5 Unit Testing Test Cases

tableUnit Test Cases with Input and Output

Test ID	**Module**	**Input**	**Expected Output**
TCU1	Frame Extraction	10s video file (.mp4)	7 image frames saved in `frames/` directory
TCU2	Face Detection (MTCNN)	Extracted frames with visible faces	Cropped face images saved in `faces/` directory
TCU3	Image Preprocessing	Cropped face image (any size)	Image resized to 224×224 with normalized pixel values
TCU4	Audio Conversion	Audio file (.wav)	Converted to 16kHz mono .wav file
TCU5	InceptionV3 Prediction	Resized image (224x224x3)	Probability score between 0 and 1 (e.g., 0.78)
TCU6	Wav2Vec2.0 Prediction	Processed 16kHz audio input	Label: `Real` or `Fake`

9.1.6 Integration Testing Test Cases

tableIntegration Test Cases with Input and Output

Test ID	**Module Integration**	**Input**	**Expected Output**
TCI1	Frame Extraction + Face Detection	10s video file (.mp4)	Detected faces saved as images in `faces/` directory
TCI2	Face Detection + InceptionV3 + LSTM	Cropped face images	Probability prediction for each face; sequence output fed to LSTM
TCI3	Video Pipeline + Classification UI	Video file uploaded through UI	Classification result displayed as `Real` or `Fake`
TCI4	Audio Preprocessing + Wav2Vec Model	Audio file (.mp3, .ogg, .wav) uploaded	Audio converted and classified correctly as `Real` or `Fake`
TCI5	End-to-End Video Path	Complete video input to GUI	Result summary with classification and detected fake faces shown

9.1.7 Performance Testing Test Cases

tablePerformance Test Cases with Input and Output

Test ID	**Performance Metric**	**Input**	**Expected Output**
TCP1	Inference Time (Video)	10-second video file with facial content	Classification result displayed in under 5 seconds
TCP2	Inference Time (Audio)	15-second audio file (.wav)	Audio classified as Real or Fake in under 3 seconds
TCP3	Memory Utilization	Simultaneous video and audio classification	RAM usage stays below 2 GB during execution
TCP4	Throughput	Batch of 5 video files uploaded sequentially	System processes all files without crash or delay
TCP5	Classification Accuracy	Benchmark dataset (FaceForensics++ or CelebDF)	Overall detection accuracy â¥ 90%
TCP6	CPU Utilization	Continuous classification workload (video + audio)	CPU usage remains under 70% on average
TCP7	File Size Handling	Video file size up to 500MB	System does not crash or timeout; classification completes successfully
TCP8	Concurrent Access	Multiple users uploading files simultaneously	System handles concurrent requests without errors or delays
TCP9	UI Response Time	Clicking classify button after upload	UI displays result in under 2 seconds post-processing
TCP10	Model Loading Time	Initial server startup with model weights	All models load and initialize in under 10 seconds

Chapter 10

Results

10.1 Screenshots

10.1.1 Web Interface

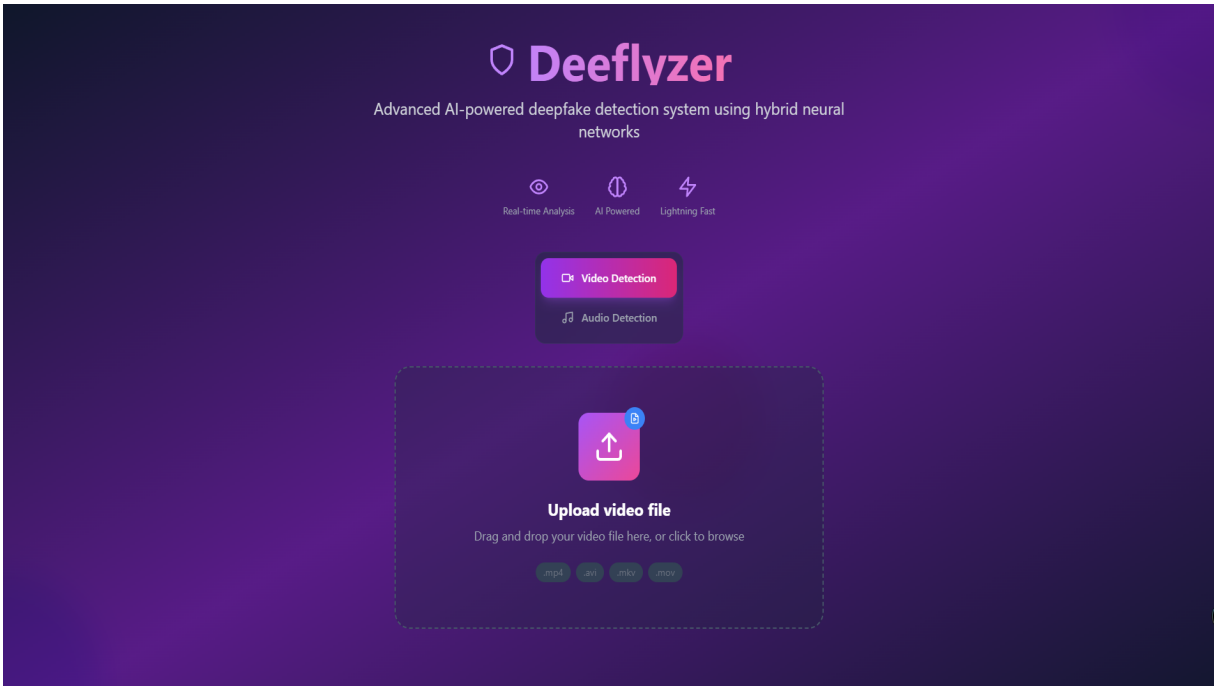
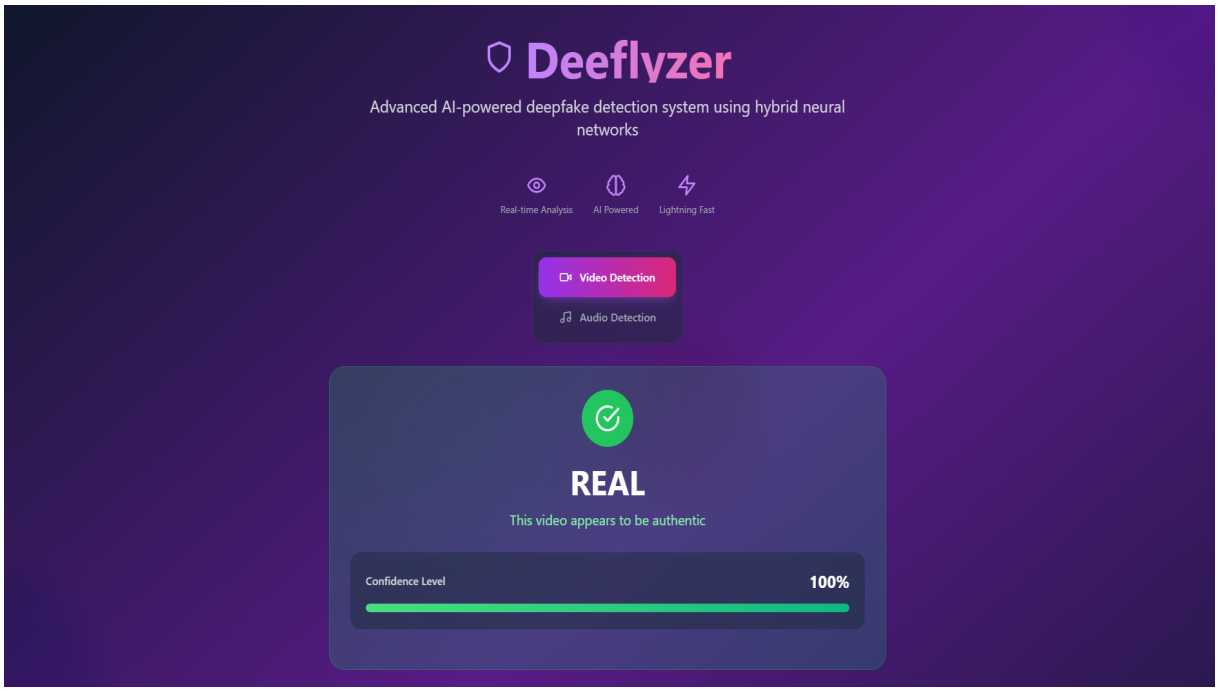
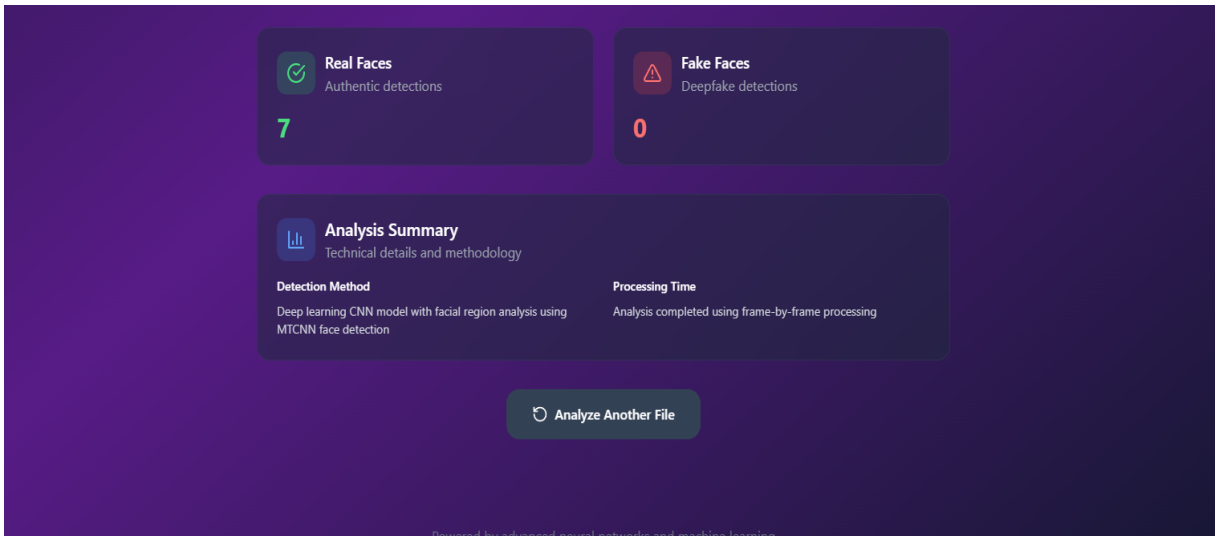
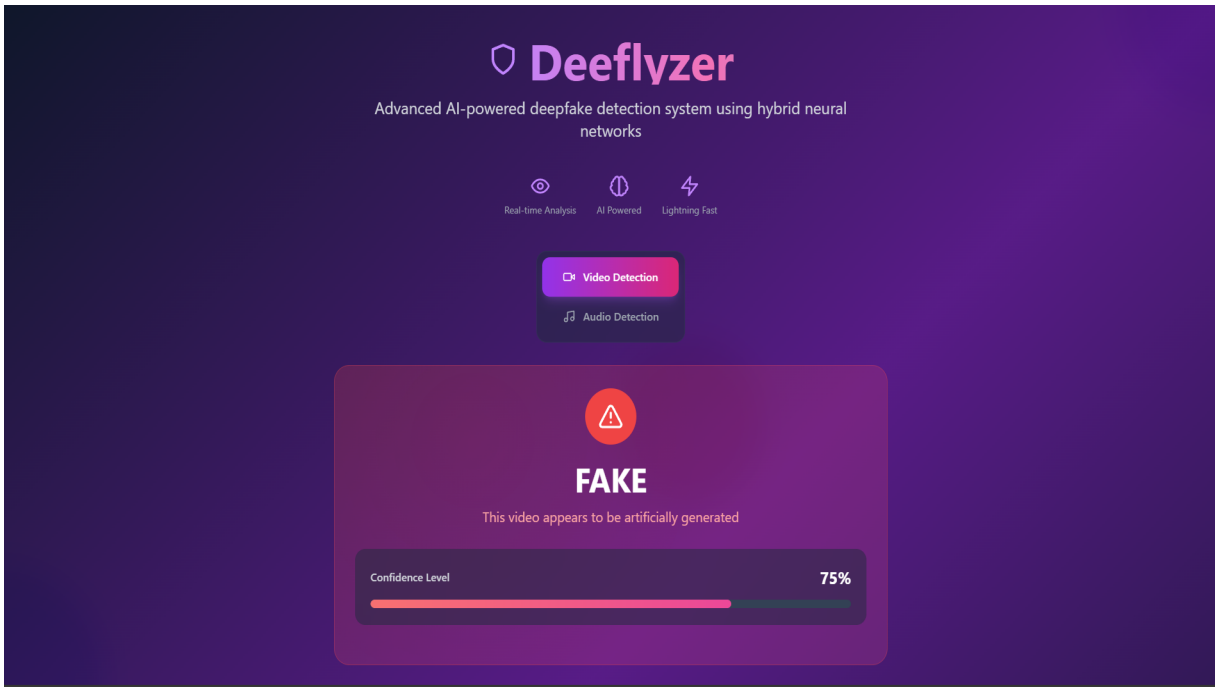


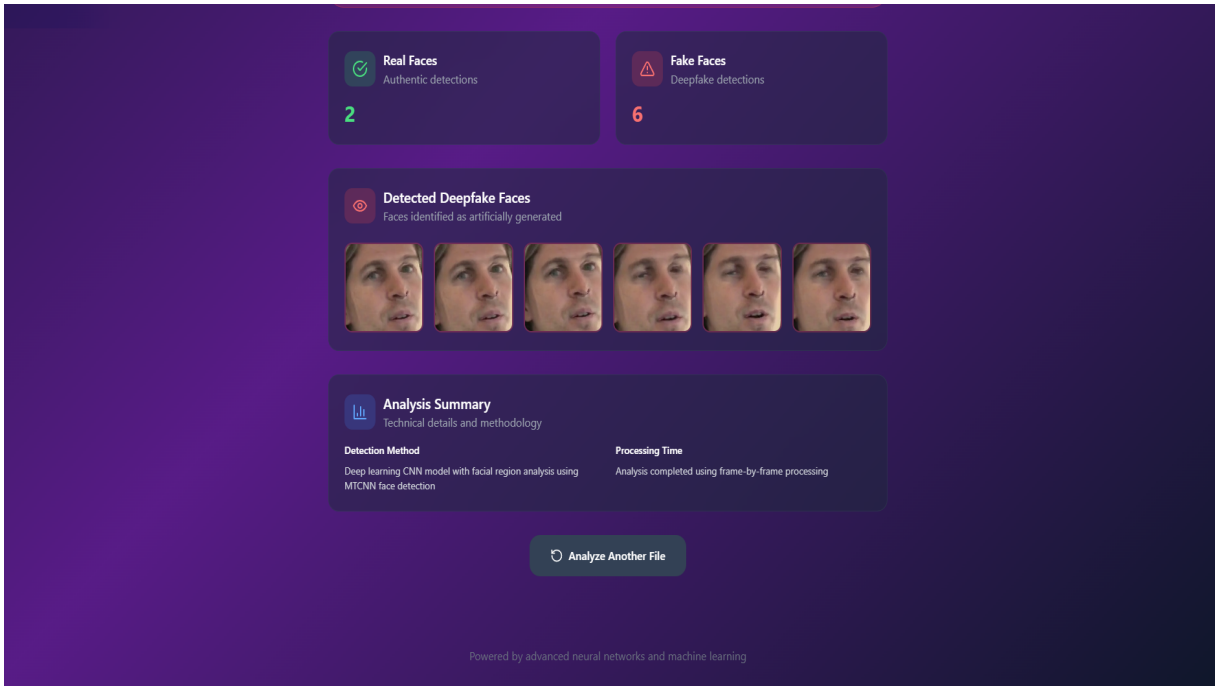
Figure 10.1: Web Interface

10.1.2 Video Deepfake Detection

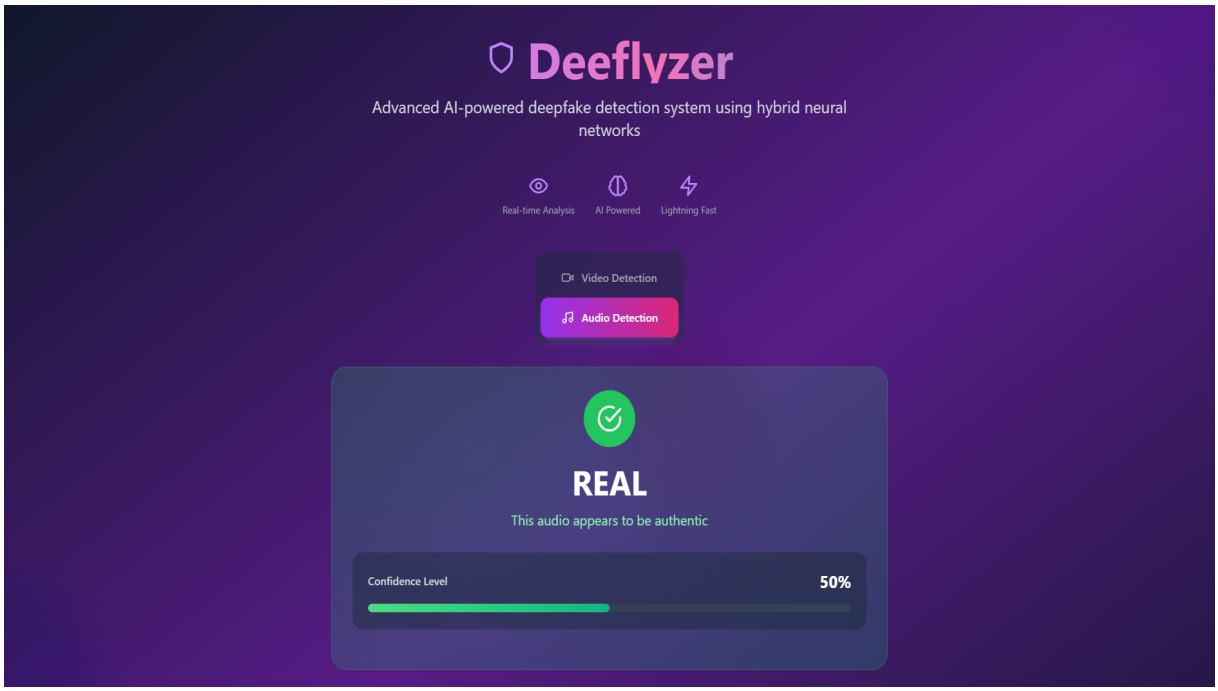


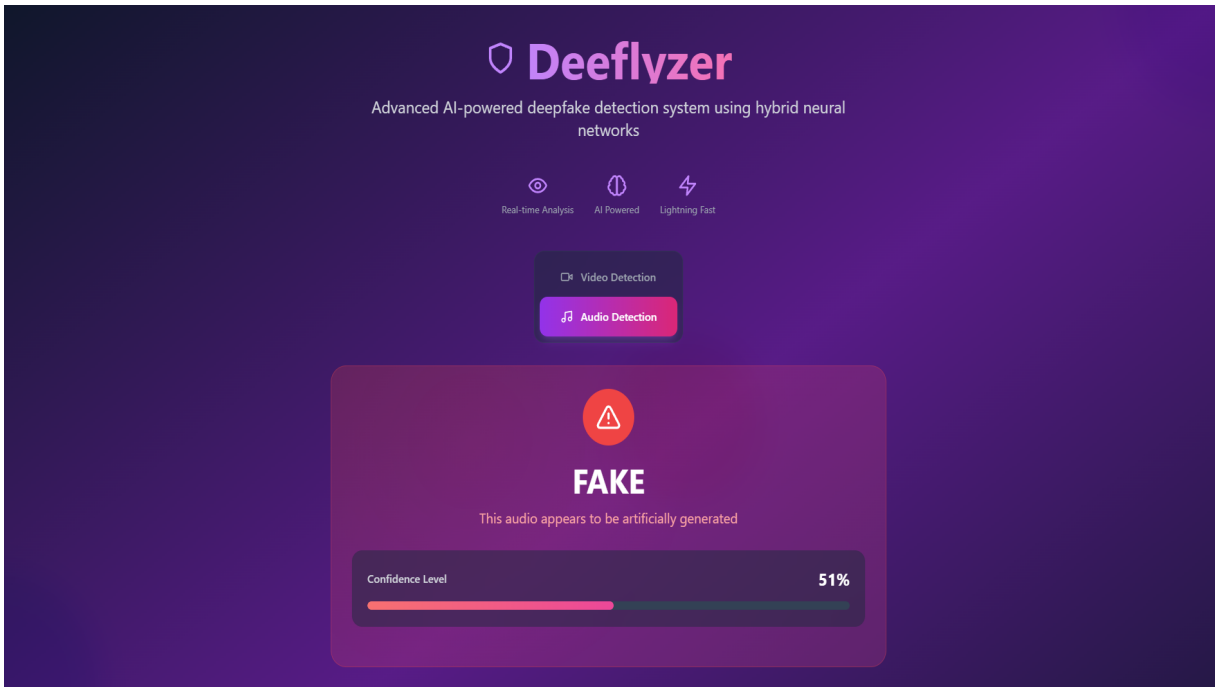



Deeflyzer
Advanced AI-powered deepfake detection system using hybrid neural networks
Real-time Analysis
AI Powered
Lightning Fast
Video Detection
Audio Detection
FAKE
This video appears to be artificially generated
Confidence Level
75%


Real Faces
Authentic detections
2
Fake Faces
Deepfake detections
6
Detected Deepfake Faces
Faces identified as artificially generated
Analysis Summary
Technical details and methodology
Detection Method
Deep learning CNN model with facial region analysis using MTCNN face detection
Processing Time
Analysis completed using frame-by-frame processing
Analyze Another File
Powered by advanced neural networks and machine learning

10.1.3 Audio Deepfake Detection





Chapter 11

Deployment and Maintenance

11.1 Installation and Uninstallation

Installation Procedure

To install and set up the Hybrid Deepfake Detection System, follow the steps below:

1. Ensure that Python (version 3.8 or above) is installed on the system.
2. Clone the project repository or download the source files to your local machine.
3. Navigate to the project directory using the command line.
4. Create a virtual environment (optional but recommended):

```
python -m venv venv
source venv/bin/activate    % On Linux/macOS
venv\Scripts\activate       % On Windows
```

5. Install the required dependencies using pip:

```
pip install -r requirements.txt
```

6. Launch the backend server:

```
python main.py
```

7. Launch the frontend server:

```
npm run dev
```

Uninstallation Procedure

To uninstall the system, follow these steps:

1. Deactivate the virtual environment (if used):

```
deactivate
```

2. Remove the virtual environment folder:

```
rm -r venv     % Linux/macOS
rmdir /S venv  % Windows
```

3. Optionally, delete the project directory and all associated media files:

```
rm -rf deepfake-detector/
```

This ensures a complete cleanup of installed components and dependencies from your system.

Chapter 12

Conclusion and Future Scope

Conclusion

In this project, a hybrid deepfake detection system was successfully developed, integrating both audio and video analysis for robust fake media identification. The system utilizes advanced deep learning models such as InceptionV3 with LSTM for video-based classification and Wav2Vec 2.0 for audio-based deepfake detection. By combining spatial, temporal, and speech features, the system demonstrates improved accuracy and generalizability across different types of manipulated media.

The project also incorporates real-time processing capabilities and a user-friendly interface using NextJS, making the solution accessible and interactive. The inclusion of face detection, frame extraction, and visualization of fake regions adds an interpretable layer to the classification process, enhancing user trust and transparency.Through rigorous testing and evaluation, the system proved effective in identifying deepfake content with considerable precision. It addresses growing concerns about the misuse of synthetic media in misinformation, fraud, and identity manipulation. This project serves as a foundational step toward more secure digital content authentication mechanisms.

Future Scope

The Hybrid Deepfake Detection System developed in this project provides a strong foundation for detecting synthetic media; however, there are several directions for future enhancement:

- **Real-Time Detection:** Integrating real-time streaming analysis to detect deepfakes in live video feeds and voice calls, which is crucial for surveillance and online conferencing platforms.
- **Multilingual Audio Support:** Expanding the system to handle and detect deepfake audio in multiple languages and accents to ensure global applicability.
- **Mobile and Web Deployment:** Optimizing the model for deployment on mobile devices and cloud platforms to enhance accessibility and usability for a wider audience.
- **Adversarial Robustness:** Incorporating adversarial training techniques to make the system more resilient to manipulation techniques designed to bypass deepfake detectors.
- **Explainability and Visualization:** Enhancing interpretability by visualizing manipulated regions in video frames or highlighting anomalies in audio patterns to build user trust.
- **Dataset Expansion:** Training the models on larger and more diverse datasets, including emerging deepfake generation techniques, to improve generalization performance.

- **Integration with Content Platforms:** Collaborating with social media and digital platforms to automate the flagging and reporting of synthetic or harmful media in real-time.
- **Legal and Ethical Frameworks:** Aligning the system with ethical guidelines and data protection laws to ensure responsible and lawful deployment.

Annexure A

References

1. Kuiyuan Zhang, Zeming Hou, Zhongyun Hua, Yifeng Zheng, Leo Yu Zhang, "Boosting Deepfake Detection Generalizability via Expansive Learning and Confidence Judgement," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

2. B. V. Chowdary, Marry Prabhakar, Mavoori Akhil, Komirishetty Pavan, B. Pavana Teja Reddy, "Deep Fake Detection using Adversarial Ensemble Techniques," *8th International Conference on Inventive Systems and Control (ICISC)*, 2024.

3. B. Sarada, T. V. S. Laxmi Sudha, Meghana Domakonda, B. Vasantha, "Audio Deepfake Detection and Classification," *Asia Pacific Conference on Innovation in Technology (APCIT)*, 2024.

4. Hao Teng, Chia-Yu Lin, "Dynamic and Static Features Extraction for Deep-fake Detection," *International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, 2024, DOI: 10.1109/ICCE-Taiwan62264.2024.10674402.

5. Haobo Liang, Yingxiong Leng, Jinman Luo, Jie Chen, Xiaoji Guo, "A Face Forgery Video Detection Model Based on Knowledge Distillation," *IEEE/ACIS 27th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 2024.

6. Amaan M. Kalemullah, Prakash P, Sakthivel V, "Deepfake Classification for Human Faces using Custom CNN," *7th International Conference on Circuit Power and Computing Technologies (ICCPCT)*, 2024.

7. Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, "Contrastive Learning for DeepFake Classification and Localization via Multi-Label Ranking," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

8. Li Lin, Xinan He, Yan Ju, Xin Wang, Feng Ding, Shu Hu, "Preserving Fairness Generalization in Deepfake Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, DOI: 10.1109/CVPR52733.2024.01591.

9. Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, Jong-won Choi, "Exploiting Style Latent Flows for Generalizing Deepfake Video Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

10. Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, Yunchao Wei, "Rethinking the Up-Sampling Operations in CNN-Based Generative Network for Generalizable Deepfake Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

11. Trevine Oorloff et al., "AVFF: Audio-Visual Feature Fusion for Video Deepfake Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, DOI: 10.1109/CVPR52733.2024.02559.

12. Siyou Guo et al., "Deepfake Detection via a Progressive Attention Network," *International Joint Conference on Neural Networks (IJCNN)*, 2024..

13. Prakash Raj S et al., "Deepfake Detection Using Deep Learning," *ICACCS, Coimbatore*, 2024, DOI: 10.1109/ICACCS60874.2024.10717155.

14. Pham Minh Thuan, Bui Thu Lam, Pham Duy Trung, "Spatial Vision Transformer: A Novel Approach to Deepfake Video Detection," *VCRIS, Hanoi*, 2024.

15. Aung Kyi Win et al., "A Novel Methodology for Deepfake Detection Using MesoNet and GAN-based Deepfake Creation," *ICAIT, Yangon*, 2024.

16. Sornavalli G, Priyanka Vijaybaskar, "DeepFake Detection by Prediction of Mismatch Between Audio and Video Lip Movement," *ADICS, Chennai*, 2024.

17. Naveed Ur Rehman Ahmed et al., "Visual Deepfake Detection: Review of Techniques, Tools, Limitations, and Future Prospects," *IEEE Access*, 2024.

18. Jitendra Chandrakant Musale, Anuj Kumar Singh, "Effective face recognition with hybrid distance key frame selection using tbo-unesamble model," *International Journal of Wavelets, Multiresolution and Information Processing*, 2024, DOI: 10.1142/S0219691323500443.

19. Jitendra Chandrakant Musale, Anuj Kumar Singh, Swati Shirke, "Tri bird technique for effective face recognition using Deep Convolutional Neural Network," *ICACCS 2023*, DOI: 10.2991/978-94-6463-314-6_33.

20. Alexandre Libourel et al., "A Case Study on how Beautification Filters Can Fool Deepfake Detectors," *IWBF*, 2024, DOI: 10.1109/IWBF62628.2024.10593932.

21. Tong Qiao et al., "Fully Unsupervised Deepfake Video Detection via Enhanced Contrastive Learning," *IEEE TPAMI*, vol. 46, no. 7, 2024, DOI: 10.1109/TPAMI.2024.3356814.

22. Yinlin Guo et al., "Audio Deepfake Detection with Self-Supervised Wavlm and Multi-Fusion Attentive Classifier," *ICASSP*, 2024, DOI: 10.1109/ICASSP48485.2024.10447923.

23. Manoj Kumar et al., "A Novel Approach for Detecting Deepfake Face Using Machine Learning Algorithms," *ICDT*, 2024, DOI: 10.1109/ICDT61202.2024.10489036.

24. Huimin She et al., "Using Graph Neural Networks to Improve Generalization Capability of the Models for Deepfake Detection," *IEEE TIFS*, 2024, DOI: 10.1109/TIFS.2024.3451356.

25. Lam Pham et al., "Deepfake Audio Detection Using Spectrogram-based Feature and Ensemble of Deep Learning Models," *IS2*, 2024, DOI: 10.1109/IS262782.2024.10704095.

26. M. Sivabalamurugan, T. R. Swapna, "Deepfake Detection and Classification Using Local Surface Geometrical Features," *CVMI*, 2024, DOI: 10.1109/CVMI61877.2024.10782175.

27. Amidela Anil Kumar et al., "XAI - Empowered Ensemble Deep Learning for Deepfake Detection," *ICCCNT*, 2024, DOI: 10.1109/ICCCNT61001.2024.10726125.

28. Daeun Song et al., "Anomaly Detection of Deepfake Audio Based on Real Audio Using GAN Model," *IEEE Open Access*, 2024.

29. Xiaoke Yang et al., "AdaForensics: Learning A Characteristic-aware Adaptive Deepfake Detector," *ICME*, 2024, DOI: 10.1109/ICME57554.2024.10687869.

30. Atharva Kohapare et al., "Implementation of Deep Learning Method for Forgery Detection on Social Media," *IDCIoT*, 2024, DOI: 10.1109/IDCIoT59759.2024.10467237.

31. Saima Waseem et al., "Attention-Guided Supervised Contrastive Learning for Deepfake Detection," *ICSIPA*, 2024, DOI: 10.1109/ICSIPA62061.2024.10687088.

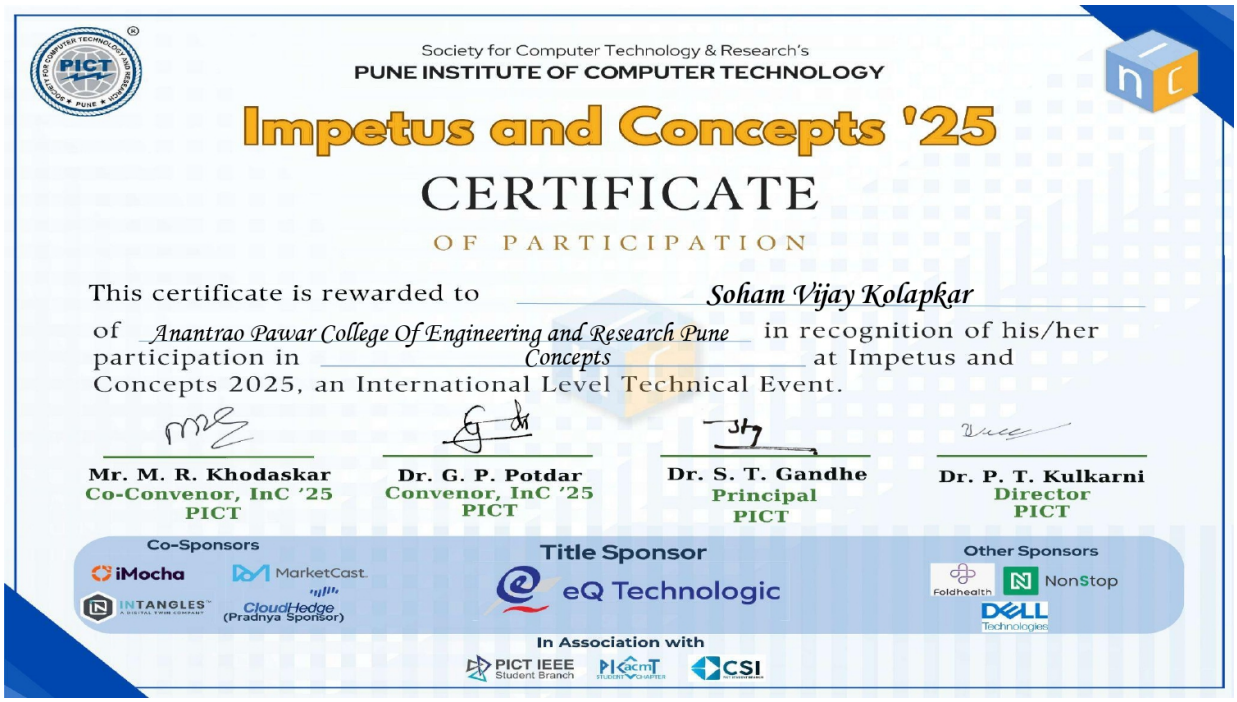
32. Yuran Qiu et al., "Analysis of Backdoor Attacks on Deepfake Detection," *IJCB*, 2024, DOI: 10.1109/IJCB62174.2024.10744504.

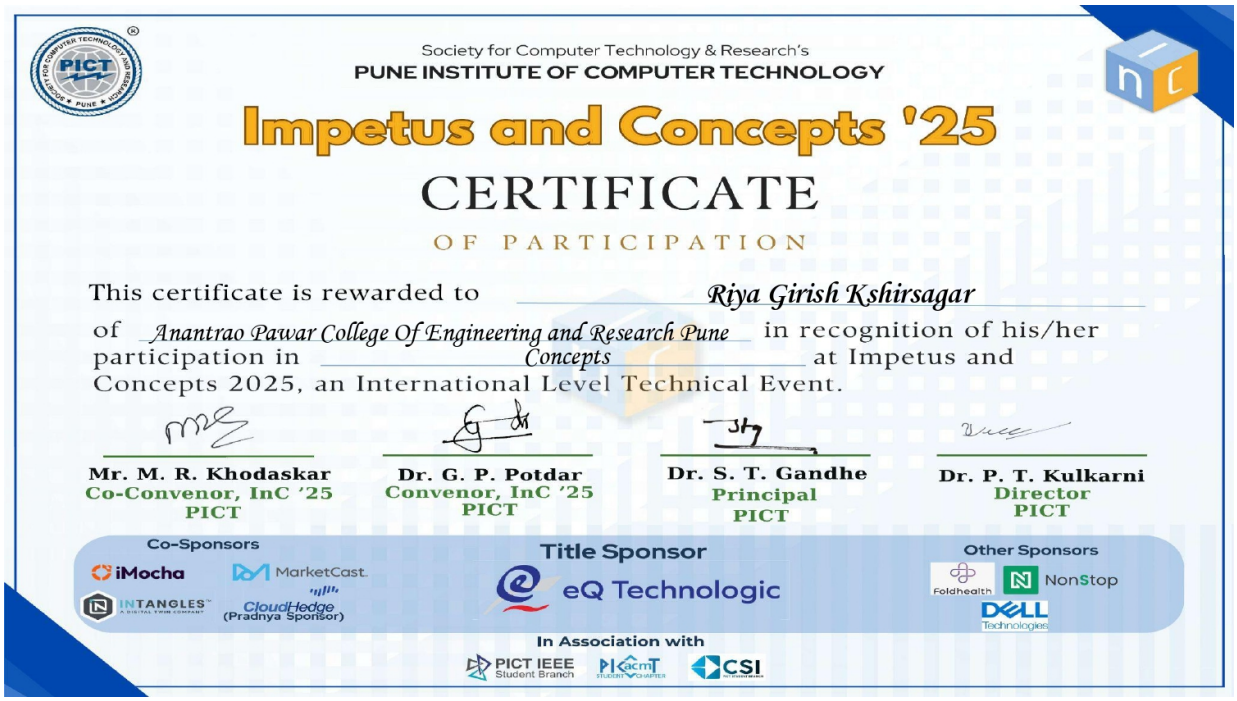
Annexure B

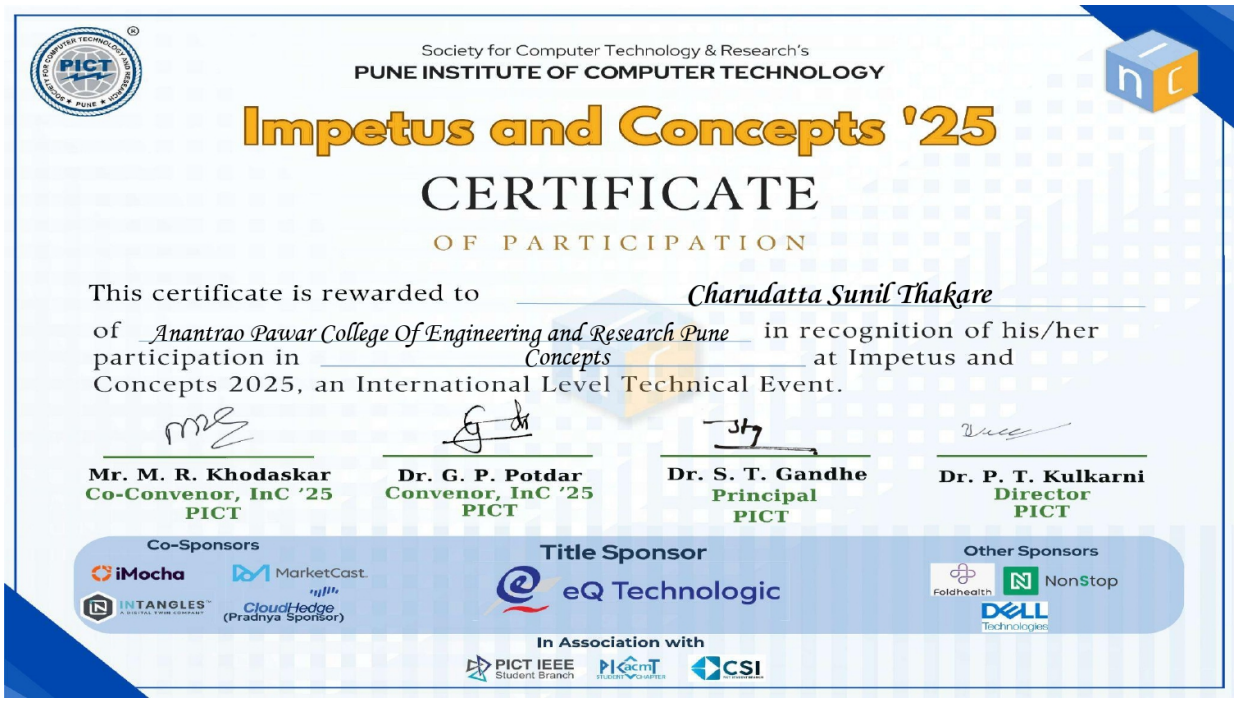
Competition participation Certificates

12.1 Competition Certificates

Competition Name: **PICT Impetus 2025**







Society for Computer Technology & Research's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Impetus and Concepts '25

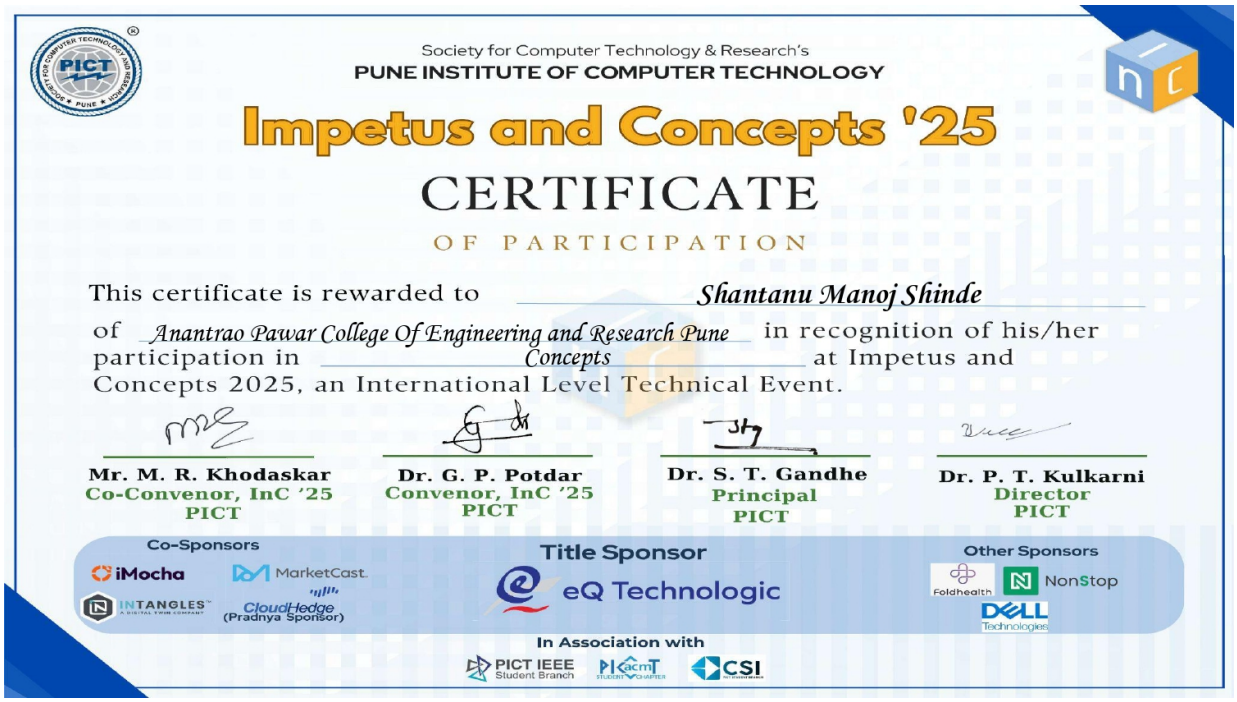
CERTIFICATE

OF PARTICIPATION

This certificate is rewarded to *Charudatta Sunil Thakare*
of *Anantrao Pawar College Of Engineering and Research Pune* in recognition of his/her participation in *Concepts* at Impetus and Concepts 2025, an International Level Technical Event.

Mr. M. R. Khodaskar, Co-Convenor, InC '25, PICT
Dr. G. P. Potdar, Convenor, InC '25, PICT
Dr. S. T. Gandhe, Principal, PICT
Dr. P. T. Kulkarni, Director, PICT

Co-Sponsors: iMocha, MarketCast, INTANGLES, CloudHedge (Pradnya Sponsor)
Title Sponsor: eQ Technologic
Other Sponsors: Foldhealth, NonStop, DELL Technologies
In Association with: PICT IEEE Student Branch, ACM Student Chapter, CSI



Society for Computer Technology & Research's
PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Impetus and Concepts '25

CERTIFICATE

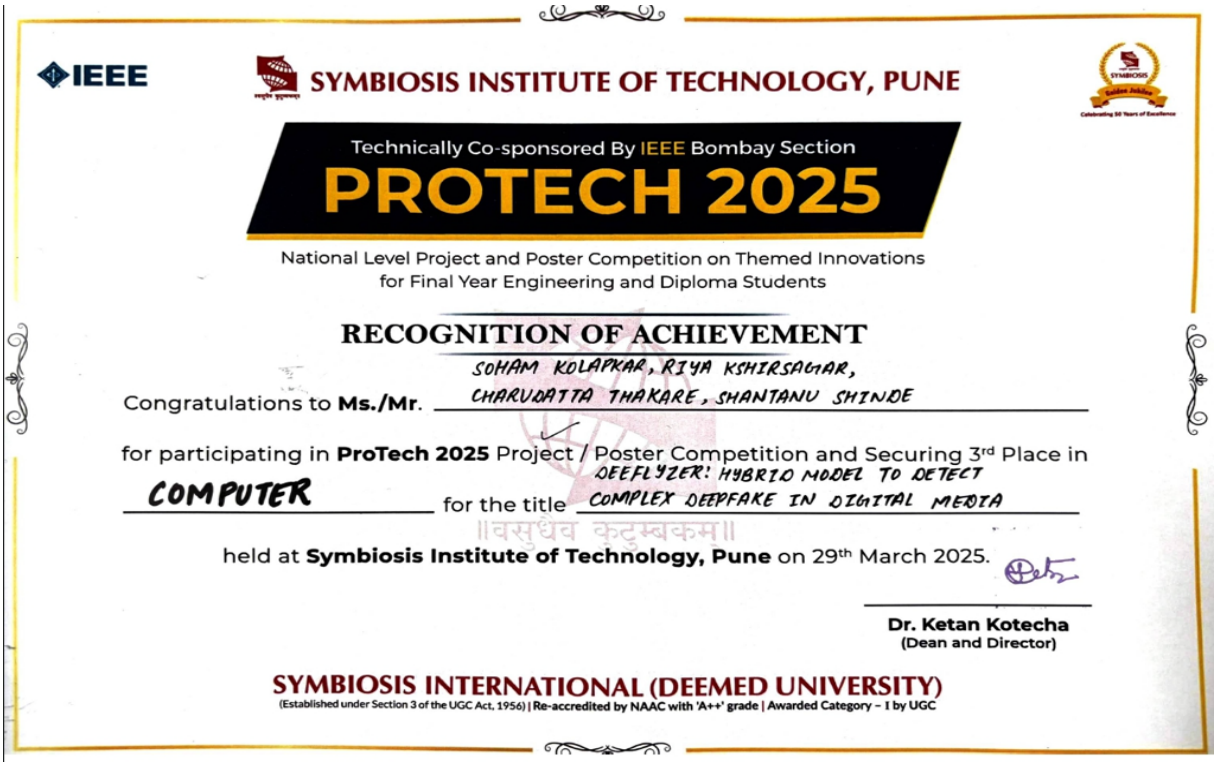
OF PARTICIPATION

This certificate is rewarded to *Shantanu Manoj Shinde*
of *Anantrao Pawar College Of Engineering and Research Pune* in recognition of his/her participation in *Concepts* at Impetus and Concepts 2025, an International Level Technical Event.

Mr. M. R. Khodaskar, Co-Convenor, InC '25, PICT
Dr. G. P. Potdar, Convenor, InC '25, PICT
Dr. S. T. Gandhe, Principal, PICT
Dr. P. T. Kulkarni, Director, PICT

Co-Sponsors: iMocha, MarketCast, INTANGLES, CloudHedge (Pradnya Sponsor)
Title Sponsor: eQ Technologic
Other Sponsors: Foldhealth, NonStop, DELL Technologies
In Association with: PICT IEEE Student Branch, ACM Student Chapter, CSI

Competition Name: **SIT Protech 2025**



IEEE

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Technically Co-sponsored By IEEE Bombay Section

PROTECH 2025

National Level Project and Poster Competition on Themed Innovations for Final Year Engineering and Diploma Students

RECOGNITION OF ACHIEVEMENT

Congratulations to Ms./Mr. SOHAM KOLAPKAR, RIYA KSHIRSAGAR, CHARUDATTA THAKARE, SHANTANU SHINDE

for participating in ProTech 2025 Project / Poster Competition and Securing 3rd Place in COMPUTER for the title DEEFLYZER: HYBRID MODEL TO DETECT COMPLEX DEEPFAKE IN DIGITAL MEDIA

॥वसुधैव कुटुम्बकम्॥

held at Symbiosis Institute of Technology, Pune on 29th March 2025.

Dr. Ketan Kotecha
(Dean and Director)

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC

Competition Name: **AVISHKAR 2025**

Pune District Education Association's

ANNASAHEB MAGAR MAHAVIDYALAYA

Hadapsar, Pune - 411 028

Affiliated to Savitribai Phule Pune University • Id No. : PU/PN/ASC/029/1971 • Jr.Coll.No. 11.15.005 (Mah.)

• AICTE and DTE Approved. Institute Id: 16332

AISHE code: C-41752 • Accrediated by NAAC : A+ (4th Cycle) • Best College Award by Savitribai Phule Pune University

SAVITRIBAI PHULE PUNE UNIVERSITY

Internal Quality Assurance Cell (IQAC)

AAVISHKAR-2024

ZONAL LEVEL RESEARCH PROJECT COMPETITION

Certificate of Participation

This is to certify that Mr./ Miss. ....KOLAPKAR SOHAM.....of

..APCOE R.....College/ Institute has participated in Zonal Level Research Project Competition for UG/PG/PPG (Ph. D.) students at "AAVISHKAR-2024" Research Festival, Under ..Engineering and Technology.... category, held at PDEA's Annasaheb Magar Mahavidyalaya, Hadapsar, Pune, on 27th November-2024

Dr. Sunita Danai-Tambhale
ARC Coordinator, AMM

Dr. Nitin Ghorpade
Principal & Member- Management Council, SPPU

Prof. Dr. Sanjay Dhole
Director, IQAC, SPPU

Pune District Education Association's

ANNASAHEB MAGAR MAHAVIDYALAYA

Hadapsar, Pune - 411 028

Affiliated to Savitribai Phule Pune University • Id No. : PU/PN/ASC/029/1971 • Jr.Coll.No. 11.15.005 (Mah.)

• AICTE and DTE Approved. Institute Id: 16332

AISHE code: C-41752 • Accrediated by NAAC : A+ (4th Cycle) • Best College Award by Savitribai Phule Pune University

SAVITRIBAI PHULE PUNE UNIVERSITY

Internal Quality Assurance Cell (IQAC)

AAVISHKAR-2024

ZONAL LEVEL RESEARCH PROJECT COMPETITION

Certificate of Participation

This is to certify that Mr./ Miss. ..THAKRE CHARUDATTA.....of

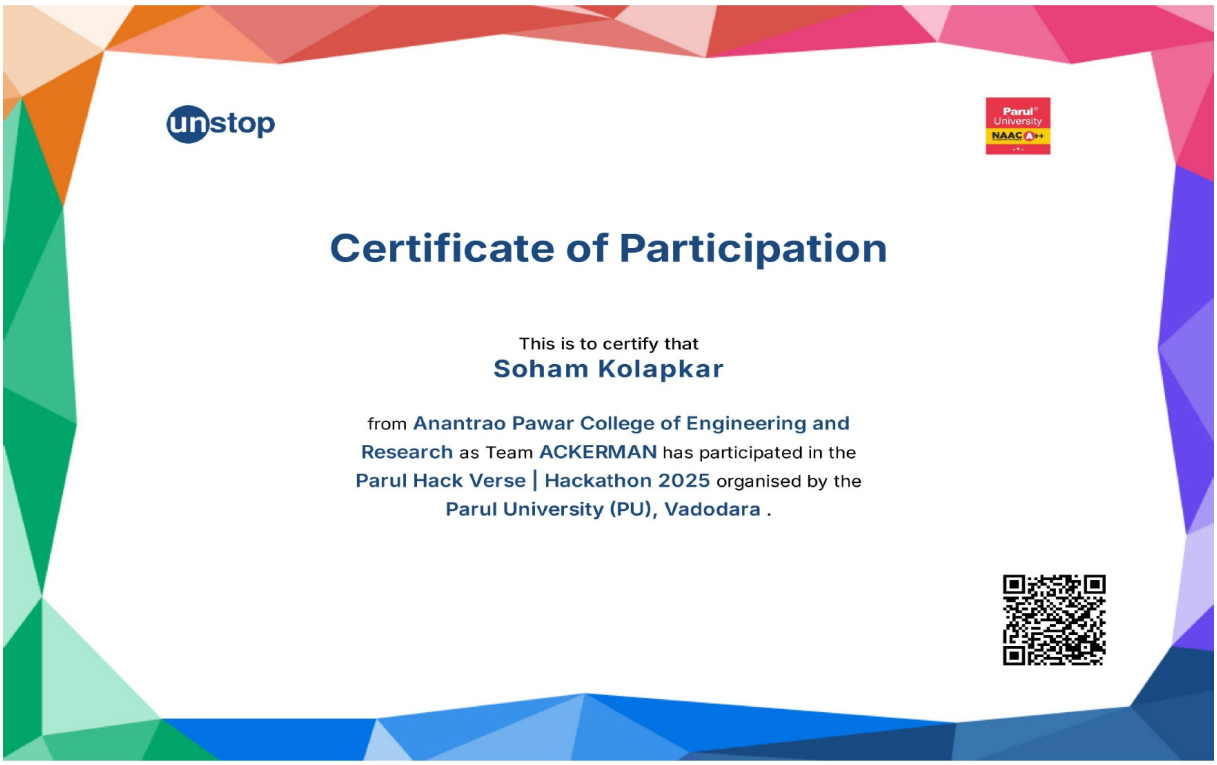
.....APCOER.....College/ Institute has participated in Zonal Level Research Project Competition for UG/PG/PPG (Ph. D.) students at "AAVISHKAR-2024" Research Festival, Under ..Engineering and Technology.... category, held at PDEA's Annasaheb Magar Mahavidyalaya, Hadapsar, Pune, on 27th November-2024

Dr. Sunita Danai-Tambhale
ARC Coordinator, AMM

Dr. Nitin Ghorpade
Principal & Member- Management Council, SPPU

Prof. Dr. Sanjay Dhole
Director, IQAC, SPPU

Competition Name: **Parul Hack Verse — Hackathon 2025 , Parul University, Vadodara**



unstop

Parul University NAAC A++

Certificate of Participation

This is to certify that

Charudatta Thakare

from **Anantrao Pawar College of Engineering and Research** as Team **ACKERMAN** has participated in the **Parul Hack Verse | Hackathon 2025** organised by the **Parul University (PU), Vadodara** .

unstop

Parul University NAAC A++

Certificate of Participation

This is to certify that

Shantanu Shinde

from **Anantrao Pawar College of Engineering and Research** as Team **ACKERMAN** has participated in the **Parul Hack Verse | Hackathon 2025** organised by the **Parul University (PU), Vadodara** .

Annexure C

Paper, Certificate, Reviewers Comments of Paper Submitted

Research Paper Title:

Deeflyzer: Enhancing Media Integrity Through Advanced Deepfake Detection

Conference Name:

2025 International Conference on Knowledge Engineering and Communication Systems (ICKECS)

Publication Certificate

|| Jai Sri Gurudev ||
Sri Adichunchanagiri Shikshana Trust (R.)

SJC INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU from 2024-25)

AICTE Approved, Accredited by NBA (CSE, ISE, ECE, ME, CV, AE) & NAAC with A+ Grade, QS I-Gauge Gold rated

www.sjcit.ac.in

Chikkaballapura - 562 101, Karnataka

Estd. 1986

IEEE Student Branch S J C Institute of Technology

IEEE BANGALORE SECTION

CERTIFICATE

This Certificate is awarded to Dr./Mr./Ms./Mrs. CHARUDATTA THAKARE from ANANTRAO PAWAR COLLEGE OF ENGINEERING & RESEARCH for having Participated and Presented the paper titled DEEFLYZER: ENHANCING MEDIA INTEGRITY THROUGH ADVANCED DEEPFAKE DETECTION Co-authored by 1. RIYA KSHIRSAGAR , 2. SOHAM KOLAPKAR , 3. SHANTANU SHINDE , 4. JITENDRA MUSALE , 5..... in "3rd IEEE International Conference on Knowledge Engineering and Communication Systems" held at SJC Institute of Technology in Association with IEEE Bangalore Section on April 28 - 29, 2025.

Dr. C Rangaswamy
HOD - ECE
Organizing Chair - ICKECS

Dr. Manjunatha Kumar B H
Dean - Academics & HOD - CSE
Technical Program Chair - ICKECS

Dr. G T Raju
Principal - SJCIT
General Chair - ICKECS

Deeflyzer: Enhancing Media Integrity Through Advanced Deepfake Detection

Charudatta Sunil Thakare
Department of Computer Engineering
ABMSPs Anantrao Pawar College of Engineering and Research
Pune, India
charudattathakare1010@gmail.com

Riya Girish Kshirsagar
Department of Computer Engineering
ABMSPs Anantrao Pawar College of Enginnering and Research
Pune, India
riya.kshirsagar@gmail.com

Soham Vijay Kolapkar
Department of Computer Engineering
ABMSPs Anantrao Pawar College of Engineering and Research
Pune, India
sohamkolapka4@gmail.com

Shantanu Manoj Shinde
Department of Computer Engineering
ABMSPs Anantrao Pawar College of Engineering and Research
Pune, India
shantanums840@gmail.com

Jitendra Musale
Department of Computer Engineering
ABMSP's Anantrao Pawar College of Engineering and Research
Pune, India
jitendra.musale@abmspecoerpune.org

***Abstract*— Deepfake technology has advanced significantly, leading to the creation of highly realistic yet artificially manipulated media. The rapid advancement of generative adversarial networks (GANs) and specialized deep learning techniques, including InceptionV3, XceptionNet, and EfficientNet, has further complicated the detection of deepfakes, necessitating the development of robust detection mechanisms. Datasets such as ASVspoof 2019, FaceForensics++, Celeb-DF, and the DeepFake Detection Challenge (DFDC) have significantly contributed to research by providing diverse benchmarks for training and evaluation. Detection methods span feature extraction, temporal consistency analysis, multi-modal learning, and adversarial training. Feature extraction-based approaches leverage CNNs to identify spatial artifacts, while temporal analysis captures inconsistencies in motion or lip synchronization. Multi-modal learning integrates audio and visual features to improve detection accuracy. After extensive testing and evaluation, our findings indicate that among the numerous models and datasets analyzed, InceptionV3 consistently outperforms others in identifying spatial anomalies in deepfake content. Similarly, DFDC emerges as the most comprehensive dataset for video deepfake detection, while ASVspoof 2019 remains unmatched for audio deepfake detection. These insights highlight critical tools for advancing the fight against deepfake threats and guiding future innovations in detection systems.**

Keywords*— *Deepfake, CNN, RNN, Spatio-Temporal.

I. INTRODUCTION

In recent years, deepfake technology has gained significant attention due to its ability to generate hyper-realistic yet artificially manipulated media. Deepfakes are created using advanced machine learning techniques, primarily generative adversarial networks (GANs), which produce synthetic content by training a generator to mimic real-world data and a discriminator to distinguish fake from real. These methods have demonstrated remarkable capabilities in creating fake images, videos, and audio that are nearly indistinguishable from authentic content [1].

While deepfake technology has introduced innovative applications in fields such as entertainment, gaming, education, and virtual reality, it also poses serious ethical and security challenges. Deepfakes are increasingly exploited for malicious purposes, such as spreading misinformation, committing identity theft, political propaganda, and financial fraud. These concerns have prompted significant interest in developing reliable and efficient detection mechanisms to counter the misuse of this technology [2] [3] [4].

The challenges of deepfake detection arise from the rapid evolution of generative techniques. Modern deepfake generators are capable of addressing common artifacts and irregularities that earlier systems could identify, making the detection process far more complex. To combat these challenges, advanced deep learning techniques, including InceptionV3, XceptionNet, and EfficientNet, have been employed. These models leverage their ability to extract intricate spatial features and detect subtle inconsistencies, improving detection performance [5] [6].

In addition to algorithmic advancements, the availability of high-quality datasets has been instrumental in fostering progress in the field. Datasets like ASVspoof 2019, FaceForensics++, Celeb-DF, and the DeepFake Detection Challenge (DFDC) provide benchmarks that simulate real-world scenarios, enabling researchers to test and refine their models under diverse conditions. ASVspoof 2019, for example, is widely recognized for its contributions to audio-based deepfake detection, while as shown in Fig. 1, DFDC offers one of the most comprehensive collections of manipulated videos [7] [8] [9]. These datasets facilitate the development of models capable of generalizing across various types of manipulations, a crucial requirement for practical deployment [10].

Detection methodologies employed in the field can be broadly categorized into feature extraction, temporal consistency analysis, multi-modal learning, and adversarial training [11]. Feature extraction methods utilize convolutional neural networks (CNNs) to detect spatial artifacts in individual frames, while temporal analysis identifies motion inconsistencies across video sequences [12]. Multi-modal approaches integrate visual and audio features to enhance detection accuracy, making them particularly effective in complex scenarios [13].

This paper provides a comprehensive survey of the state-of-the-art deepfake detection systems, evaluating their effectiveness and limitations. Based on extensive testing, this study identifies InceptionV3 as a leading model for spatial anomaly detection, and highlights DFDC and ASVspoof 2019 as exemplary datasets for video and audio deepfake

detection, respectively [14]. These findings aim to guide future research and development in building robust systems to combat the growing threats posed by deepfake technology [15].

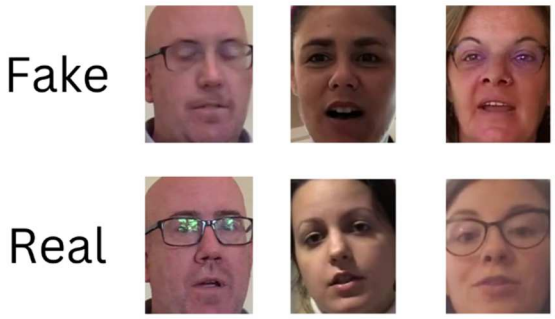


Fig. 1: Examples from dataset

II. Literature Survey

Tong Qiao [1] and associates proposed an unsupervised deepfake video detection system utilizing enhanced contrastive learning. They reviewed existing methods and identified the need for unsupervised learning approaches. Their system leverages feature representation learning without labeled data, significantly improving detection performance. The proposed method adapts to diverse datasets, ensuring robust and scalable detection of deepfake videos in various real-world scenarios.

Yinlin Guo [2] and their team proposed an audio deepfake detection framework com-bining self-supervised WavLM and a multi-fusion attentive classifier. They surveyed existing techniques and determined that integrating self-supervised learning with attention mechanisms improves accuracy. The system effectively captures subtle differences in audio signals, enhancing robustness against different manipulation methods. This innovative method demonstrates improved generalization and effectiveness across multiple datasets.

Manoj Kumar [3] and associates presented a machine learning-based deepfake detec-tion approach. They reviewed existing models and found that advanced machine learning algorithms, when applied to facial feature inconsistencies, yield better detection accuracy. The proposed system emphasizes lightweight algorithms for real-time applications and efficient detection, providing a practical solution to address deepfake prevalence on media platforms.

Huimin She [4] and their colleagues proposed a deepfake detection method leveraging graph neural networks (GNNs). Through their research, they identified that GNNs effectively model spatial relationships in visual data, improving generalization. Their system enhances cross-dataset adaptability by addressing the limitations of overfitting in traditional models. This approach ensures robustness against diverse deepfake types and formats, advancing detection capabilities.

Lam Pham [5] and associates introduced a deepfake audio detection system based on spectrogram features and ensemble deep learning models. Their analysis revealed that spectrograms capture subtle temporal and frequency patterns effectively. By integrating multiple architectures, the ensemble approach enhances robustness against various audio manipulations. This method outperforms traditional systems, making it a reliable tool for detecting audio deepfakes in real-world scenarios.

M Sivabalamurugan [6] and T R Swapna proposed a deepfake detection system fo-cusing on local surface geometrical features. They examined existing methods and identified that analyzing distortions in surface geometry significantly improves detection accuracy. The proposed system captures minute inconsistencies in facial structures, ensuring reliable detection of both known and novel deepfake formats. This approach offers a practical solution for combating digital face manipulation.

Amidela Anil Kumar [7] and colleagues developed an explainable AI-enabled ensemble deep learning system for detecting deepfakes. They reviewed existing methods and found that combining multiple models with interpretability improves both accuracy and trustworthiness. The system aggregates predictions from various architectures, providing robust detection against diverse fake formats. The explainability aspect enhances user confidence, making it suitable for critical applications like forensics and media monitoring.

Daeun Song [8] and their team proposed a GAN-based anomaly detection system for deepfake audio. Through their research, they identified that adversarial learning effectively identifies discrepancies between real and fake audio distributions. Their system adapts to evolving deepfake techniques and outperforms traditional methods in accuracy and generalization, offering a robust framework for audio deepfake detection in various applications.

Xiaoke Yang [9] and colleagues introduced AdaForensics, a dynamic deepfake detection framework. They identified that a characteristic-aware, adaptive approach improves accuracy across diverse datasets. AdaForensics adjusts detection strategies based on specific deepfake attributes, ensuring consistent performance against emerging manipulation techniques. Their results highlight the model's robustness, making it a promising solution for evolving deepfake technologies.

Atharva Kohapare [10] and their team proposed a deep learning-based forgery detec-tion system tailored for social media platforms. They analyzed existing methods and focused on developing techniques optimized for identifying manipulated content in low-quality uploads. The system integrates neural networks and preprocessing tech-niques to enhance detection accuracy, offering a practical solution for moderating social media content in real-time scenarios.

Saima Waseem [11] and associates presented a deepfake detection framework using attention-guided supervised contrastive learning. They determined that attention mechanisms effectively highlight critical regions in input data. The proposed system enhances feature extraction and class separation, leading to improved detection accuracy across datasets. This robust method demonstrates high scalability and performance in diverse deepfake detection applications.

Yuran Qiu [12] and their team analyzed the vulnerabilities of deepfake detection models to backdoor attacks. Their research identified common attack strategies and proposed defenses like adversarial training and data sanitization. The study emphasizes the importance of securing detection

systems to maintain their reliability and integrity in applications like forensic analysis and digital content verification.

Cheng-Yao Hong [13] and their associates proposed a new way of detecting and identifying deepfakes in images. Their approach to detecting deepfake is such that it divides the images into smaller parts called as patches and then it considers the whole image as a bag of patches. If one patch is manipulated then the whole image will be considered as fake and to do that, they used a unique way called the multiple instance learning (MIL). For identifying the specific part of the image which has been deepfake it used the multi-label ranking which label all parts of an images and returns the forged part.

Li Lin [14] along with their colleagues provided a solution to a problem is which occurs while detecting deepfakes. In their paper they addressed the issue of fairness generalization. Their study states that existing deepfake models are trained to detect manipulations but they are not efficient in detecting for people from different race and gender. The experiments conducted by them state that using this method the accuracy and effectiveness of the state-of-the-art methods can be easily surpassed.

Jongwook Choi [15] and their team presents a new approach in detecting fake videos based on style latent vectors. Their approach consists of targeting the temporal inconsistencies in fake videos. They have made use of the StyleGRU module which has been trained using contrastive learning used to represent the feature of style latent vectors. They also added a style attention module to detect visual artifacts. After testing this approach on multiple datasets, it is stated that it proves very effective.

III. Research Methodology

A. Literature Review:

An extensive review of academic publications and technical reports was conducted to explore advancements in deepfake detection techniques. The analysis focused on understanding the evolution of detection methods, the use of diverse datasets, and the metrics employed to evaluate model performance.

B. Model Selection:

Deep learning architectures, including InceptionV3, XceptionNet, and EfficientNet, were selected based on their prevalence in deepfake detection and ability to extract spatial and temporal features effectively.

C. Dataset Selection:

Benchmark datasets were selected based on their real-world applicability:

- **ASVspoof 2019** – Audio deepfake detection.
- **FaceForensics++** – Manipulated face dataset.
- **Celeb-DF** – High-quality deepfake dataset.
- **DFDC**– Comprehensive deepfake video dataset.

D. Preprocessing and Data Preparation:

Preprocessing included:

- **Frame Extraction:** Isolating key frames from video datasets.
- **Audio Cleaning:** Removing noise from deepfake audio samples.
- **Data Augmentation:** Enhancing model generalization using transformations like flipping, rotation, and noise injection.
- **Normalization:** Ensuring consistent feature scaling.
- **Handling Class Imbalance:** Applying synthetic oversampling and weighted loss functions.

E. Hyperparameter Tuning:

- The hyperparameter tuning process involved grid search and adaptive learning rate scheduling to optimize performance:
- **Batch Size:** 32–64 (chosen based on GPU memory constraints and model stability).
- **Learning Rate:** Initially set to 1e-4, with an adaptive scheduling strategy (ReduceLROnPlateau) to avoid overfitting.
- **Optimizer:** AdamW (Weight decay variation of Adam) was used for better generalization.
- **Loss Function:** Binary Cross-Entropy Loss, since deepfake detection is a binary classification problem.

F. Performance Evaluation:

Models were assessed using:

- Accuracy, Precision, Recall, and F1-score.
- Computational efficiency and real-time performance.
- Analysis of spatial artifacts, temporal inconsistencies, and multi-modal data fusion.

G. Comparative Analysis:

A systematic comparison was conducted to assess the strengths and weaknesses of each model-dataset

combination, focusing on their effectiveness, generalization ability, and real-time performance.

H. Validation and Experimental Analysis:

To ensure the robustness of Deeflyzer, extensive validation experiments were conducted, including:

- **External Validation on Real-World Deepfake Media:**
  The model was tested on manipulated media from real-world social media platforms and forensic datasets to assess its effectiveness beyond standard benchmark datasets.
- **Cross-Validation & Generalization Testing:**
  K-Fold Cross-Validation (k=5) was performed to ensure the model's stability across different data distributions. External dataset validation was conducted using unseen datasets to measure generalizability.
- **Statistical Validation Techniques:**
  Confidence Intervals were computed to quantify the reliability of predictions.P-values were used to assess statistical significance in performance improvements.
- **Benchmarking Against State-of-the-Art Models:**
  Deeflyzer was compared with Vision Transformer-based models, adversarially trained networks, and multi-modal deepfake detection frameworks. Performance was evaluated based on accuracy, recall, F1-score, and computational efficiency.

I. Practical Feasibility and Deployment Considerations:

To assess Deeflyzer's usability in real-world applications, the following factors were analyzed:

- **Feasibility for Real-Time Applications:**
  The system was designed to be deployed in forensic investigations, social media content moderation, and live-streamed media verification. An API-based approach enables easy integration with YouTube, Facebook, and real-time video platforms for automated deepfake detection.
- **Computational Efficiency & Scalability:**
  Optimized using model pruning and quantization techniques to reduce processing overhead. The architecture allows parallel processing on cloud-based servers for large-scale deepfake monitoring, making it suitable for high-traffic environments.
- **Integration with Law Enforcement & Social Media Platforms:**
  Law enforcement agencies can utilize Deeflyzer for forensic verification of deepfake evidence. Secure data handling policies and compliance with privacy regulations ensure safe usage in forensic cases and large-scale content moderation.

J. Key Insights:

- **InceptionV3** was the most effective for **spatial anomaly detection**.
- **DFDC** emerged as the most comprehensive dataset for **video deepfake detection**.
- **ASVspoof 2019** was the best-performing dataset for **audio deepfake detection**.

K. Recommendations for Future Work:

- **Exploring ensemble learning** to enhance detection robustness.
- **Improving adversarial resistance** through refined training strategies.
- **Enhancing real-time detection capabilities** for deployment in forensic applications.
- **Integrating multi-modal approaches** to improve deepfake detection across different media formats.

IV. ALGORITHM

After the text edit has been completed, the paper is ready for the template.

A. *Input Acquisition:* Receive the input video file for processing.

B. *Pre-processing:* Extract audio and video components separately. Normalize video frames to ensure uniformity in resolution and eliminate artifacts caused by compression. Segment the audio into smaller chunks for detailed feature extraction.

C. *Feature Extraction:* Apply InceptionV3 to extract spatial features from video frames and detect anomalies. Use WAVLM to extract relevant features from segmented audio clips.

D. *Temporal Analysis:* Utilize LSTM to analyze sequential dependencies in video features and detect inconsistencies over time. Apply MFA on extracted audio features to identify irregularities in pitch, cadence, and tone.

E. *Fusion of Modalities:* Combine insights from both video and audio analyses to improve detection accuracy.

F. *Classification and Decision Making:* Use the extracted features to classify the input as real or deepfake using a trained classifier.

G. *Prediction Output:* Generate the final prediction, indicating the probability of the input being a deepfake. Provide confidence scores to support decision-making and further verification.

V. Flow Diagram of Proposed Work

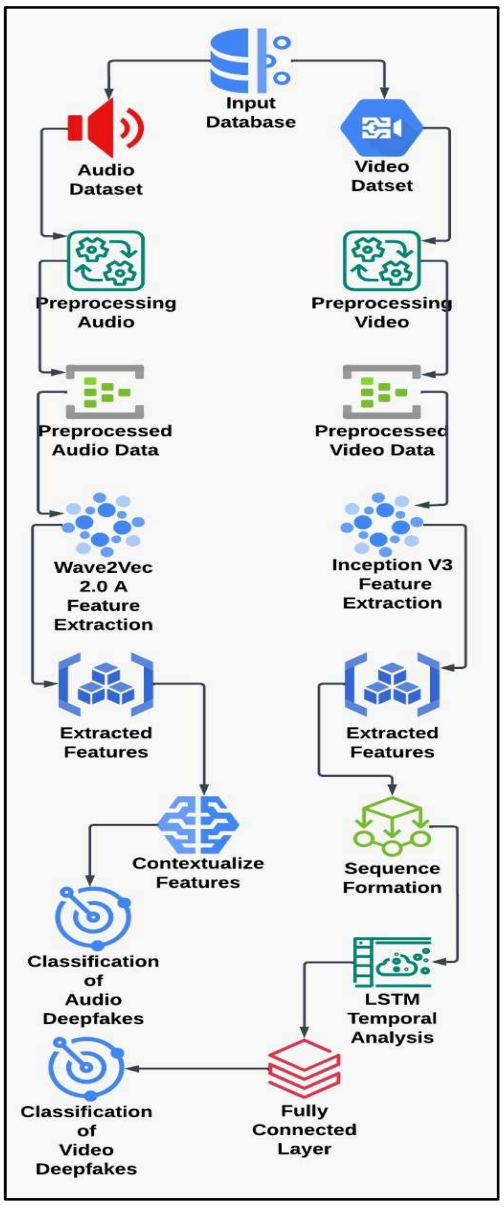


Fig. 2: Flow diagram of Proposed System

The above Fig. 2 demonstrates the actual flow of Deeflyzer system, where the user submits the media which can be either audio or video and receives the output as a classification in the form of 'Real' or 'Fake'. The working of each individual component can be understood in the above diagram.

VI. Experimentation Result

In this study, extensive experiments were conducted using various datasets and deep learning architectures to evaluate the effectiveness of different models in deepfake detection. The experimentation involved testing combinations of video and audio datasets, including DFDC, FaceForensics++, Celeb-DF for video and ASVspoof 2019 for audio, alongside different neural network architectures. For video-based detection, models such as XceptionNet, EfficientNet, and InceptionV3 were employed, while for audio deepfake detection, architectures like WAVLM and Wave2Vec 2.0 were analyzed.

LSTM (Long Short-Term Memory) is the most effective recurrent neural network (RNN) model for capturing temporal dependencies and sequential patterns in videos, making it the optimal choice for analyzing deepfake video frames over time; there-fore, we have selected LSTM for our deepfake detection system.

TABLE I. Comparison of Models and Dataset`

Models	Dataset Used	Metrics		
		Accuracy	Precision	Recall
InceptionV3	DFDC	96.18%	89.12%	91.13%
	FF++	95.19%	87.35%	86.51%
	Celeb-DF	95.46%	88.11%	84.11%
EfficientNet	DFDC	95.68%	82.42%	83.22%
	FF++	93.34%	83.13%	82.43%
	Celeb-DF	95.10%	84.72%	84.64%
XceptionNet	DFDC	91.79%	81.64%	87.86%
	FF++	89.44%	83.54%	82.78%
	Celeb-DF	93.72%	87.10%	89.40%

The performance of these models was assessed based on feature extraction capabilities, robustness to manipulations, and overall classification accuracy.

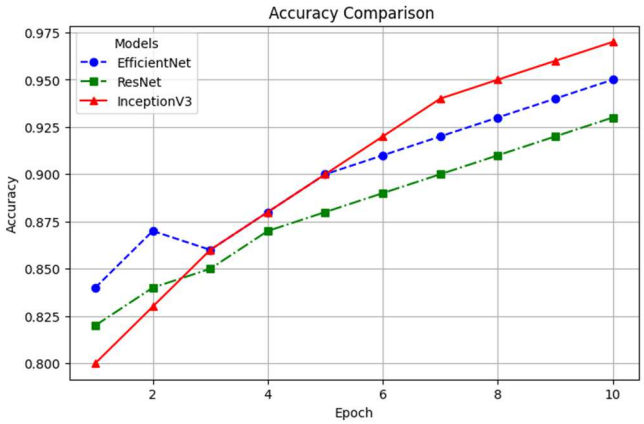


Fig. 3: Accuracy Comparison

The results of our experimentation as seen in Table 1, indicate that the DFDC dataset is the most com-prehensive and effective for video deepfake detection, as it provides a diverse range of manipulated media, making it ideal for training robust models. Similarly, ASVspoof 2019 was found to be the best dataset for detecting audio deepfakes due to its wide coverage of spoofing attacks and real-world scenarios. Among the models tested, InceptionV3 combined with LSTM demonstrated superior performance in identifying spatial and temporal inconsistencies in deepfake videos as demonstrated in Fig. 3, while Fig 4 demonstartes Accuracy comparison of Audio Models indicating Wave2Vec 2.0 outperformed other models in detecting anomalies in deepfake audio.

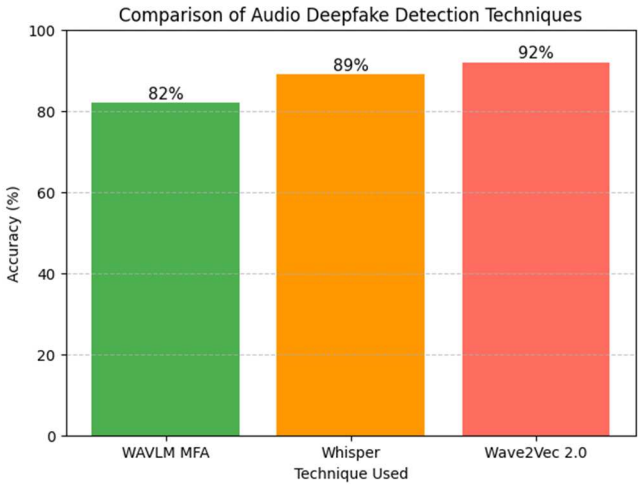


Fig 4: Accuracy comparison of Audio Models

These findings reinforce the significance of dataset diversity and model selection in developing effective deepfake detection systems.

VII. Advantages of Proposed Model

A. *Multi-Modal Analysis:*

The system integrates video, image, and audio modalities, ensuring a comprehensive analysis. By leveraging different data streams, it increases the accuracy and robust-ness of deepfake detection.

B. *Enhanced Spatial Feature Detection:*

Utilizing InceptionV3 for image and video frame processing allows the system to identify intricate spatial artifacts, improving its ability to detect subtle deepfake manipulations in visual content.

C. *Temporal Consistency Analysis:*

The incorporation of LSTM enables the system to analyze temporal patterns in video data, detecting inconsistencies such as unnatural frame transitions or mismatched lip synchronization.

D. *High-Performance Audio Analysis:*

Through WAVLM, the system extracts advanced audio features, making it adept at detecting audio deepfakes. MFA further refines the analysis by identifying inconsistencies in pitch, tone, and cadence.

E. *Generalization Across Diverse Data:*

By employing benchmark datasets like DFDC for video and ASVspoof 2019 for audio, the system is trained and tested on a wide variety of real-world scenarios, improving its generalization capabilities.

F. *Improved Accuracy and Reliability:*

The combination of cutting-edge models, such as InceptionV3 and WAVLM, along-side advanced techniques like temporal and multi-modal analysis, ensures high accuracy in detecting both audio and visual deepfakes.

G. *Scalability:*

The modular design of the system allows it to be easily scaled for new datasets or updated with emerging detection models, ensuring long-term adaptability to evolving deepfake technologies.

VIII. Challenges

A. *Data Privacy and Security:*

Challenge: Data Breaches are the concern in today's digital era, so the information should be kept secured from intruders.
Mitigation: As user data is our topmost priority we are bound to keep it protected from any unauthorized access and moreover by implementing strong encryption methods and rigid access controls we will safeguard the data.

B. *Integration with Existing Media Platforms:*

Challenge: Our detection system needs to support and work with a wide range of social media platforms, which can be a complex task given the variety of systems out there.
Mitigation: We have plans to build flexible APIs that can adapt to different platforms. By working closely with the intended media platform, we'll customize our system to fit their specific needs.

C. *Real-Time Detection and Processing:*

Challenge: Detecting deepfake media content in real-time, during live video streams or video calls requires our system to be both fast and accurate.
Mitigation: We will focus on optimizing our models to handle media quickly with intended accuracy. This means using cutting-edge algorithms we will meet all the demands of live detection.

D. *Model Robustness and Adaptability:*

Challenge: As recent and ever-rising techniques in the creation of deepfake content are improving our system needs to stay effective and updated all the time.
Mitigation: By gathering the feedback from the user based on their experience we will be able to adapt to the new changes which will also help to ensure that the committed system stays reliable and effective to detect the deepfakes.

E. *Handling of False Positives and False Negatives:*

Challenge: the optimal balance is required between fake content and falsely identified fake content, it’s our priority to avoid as much as possible to mark fake content as genuine and vice versa.
Mitigation: In our detection system we will be minimizing the false positives and negatives by better tuning our model and testing systems to constantly improve the accuracy.

F. *Scalability and Resource Management:*

Challenge: Scalability will always be the challenge as at every second thousands of user increase and with this increase in demands media files, content also increases leading to slowing down of the system.
Mitigation: Using the available cloud structure we can manage the workloads which will be helpful to make system run without any errors.

G. *Low quality dataset and high computational cost:*

Challenge: The substandard data set available on online platforms are mostly incomplete, mislabeled and moreover redundant, causing a lot of problems in training the model

accurately with precision. Also the high computational power leads to hard-ware limitations and may cause inefficient algorithms.

Mitigation: to null and void the impact we have used data cleaning methods and feature selection to reduce the incomplete and eliminate the redundancy and for the high computation we have found a novel approach that will optimize the findings.

IX. Future Scope

A. Integration with Real-Time Applications:

Future advancements can focus on deploying the system in real-time platforms such as video conferencing, live-streaming services, and social media monitoring to detect deepfakes instantaneously.

B. Low-Resource Optimization:

Optimizing the system for deployment on low-resource devices such as smartphones or edge computing platforms can make it accessible for broader use, especially in resource-constrained environments.

C. Multi-Language and Cross-Domain Support:

Expanding audio analysis to include multiple languages and accents, as well as sup-porting domain-specific applications like healthcare, legal documentation, and financial fraud prevention, will broaden the system's applicability.

D. Explainable AI (XAI) Integration:

Incorporating XAI techniques can provide transparency in predictions, enabling users to understand why a media sample is flagged as a deepfake. This is crucial for increasing trust in detection systems, especially in forensic and legal contexts.

E. Collaboration with Regulatory Bodies:

The system can serve as a cornerstone for collaborating with governments, law enforcement, and social media platforms to establish standard protocols for detecting and addressing deepfake threats.

X. Conclusion

Deepfake technology has emerged as both a marvel of modern AI and a significant societal challenge. While its creative applications in entertainment and education are notable, its misuse in spreading misinformation, identity theft, and digital fraud has raised critical concerns. The proposed deepfake detection system, integrating state-of-the-art models like InceptionV3, offers a robust multi-modal approach to counter these challenges. By leveraging benchmark datasets such as DFDC and ASVspoof 2019, the system demonstrates exceptional performance in detecting both visual and audio-deepfakes.

This survey consolidates existing research, highlighting the strengths and limitations of current methods while presenting a comprehensive framework for future innovations. With further advancements, such as dynamic dataset updates, low-resource optimization, and ethical detection practices, this system can become a cornerstone in combating deepfake threats, safeguarding digital integrity, and fostering trust in media authenticity.

References

[1] Tong Qiao, Shichuang Xie, Yanli Chen, Florent Retraint, Xiangyang Luo "Fully Unsupervised Deepfake Video Detection via Enhanced Contrastive Learning" IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 46, Issue: 7, July 2024) DOI: 10.1109/TPAMI.2024.3356814

[2] Yinlin Guo, Haofan Huang, Xi Chen, He Zhao, Yuehai Wang "Audio Deepfake Detection with Self-Supervised Wavlm and Multi-Fusion Attentive Classifier" 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) DOI: 10.1109/ICASSP48485.2024.10447923

[3] Manoj Kumar, Praveen Kumar Rai, Pankaj Kumar "A Novel Approach for Detecting Deepfake Face Using Machine Learning Algorithms" 2024 2nd International Conference on Disruptive Technologies (ICDT) DOI: 10.1109/ICDT61202.2024.10489036

[4] Huimin She, Yongjian Hu, Beibei Liu, Jicheng Li, Chang-Tsun Li "Using Graph Neural Networks to Improve Generalization Capability of the Models for Deepfake Detection" IEEE Transactions on Information Forensics and Security DOI: 10.1109/TIFS.2024.3451356

[5] Lam Pham, Phat Lam, Truong Nguyen, Huyen Nguyen, Alexander Schindler "Deepfake Audio Detection Using Spectrogram-based Feature and Ensemble of Deep Learning Models" 2024 IEEE 5th International Symposium on the Internet of Sounds (IS2) DOI: 10.1109/IS262782.2024.10704095

[6] M Sivabalamurugan, T R Swapna "Deepfake detection and classification using local surface geometrical features" 2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI) DOI: 10.1109/CVMI61877.2024.10782175

[7] Amidela Anil Kumar, S J Dheepthi Priyangha, P Meghana, Muppalla Dheeraj, R Aarthi "XAI – Empowered Ensemble Deep Learning for Deepfake Detection" 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) DOI: 10.1109/ICCCNT61001.2024.10726125

[8] Daeun Song, Nayoung Lee, Jiwon Kim, Eunjung Choi "Anomaly Detection of Deepfake Audio Based on Real Audio Using Generative Adversarial Network Model" Open Access IEEE

[9] Xiaoke Yang, Haixu Song, Xiangyu Lu, Shao-Lun Huang, Yueqi Duan "AdaForensics: Learning A Characteristic-aware Adaptive Deepfake Detector" 2024 IEEE International Conference on Multimedia and Expo (ICME) DOI: 10.1109/ICME57554.2024.10687869

[10] Atharva Kohapare, Karan Dhongade, Rahul Sukare, Priya Maidamwar "Implementation of Deep Learning Method for Forgery Detection on Social Media" 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) DOI: 10.1109/IDCIoT59759.2024.10467237

[11] Saima Waseem, Syed Abdul Rahman Bin Syed Abu Bakar, Bilal Ashfaq Ahmed "Attention-Guided Supervised Contrastive Learning for Deepfake Detection" 2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA) DOI: 10.1109/ICSIPA62061.2024.10687088

[12] Yuran Qiu, Huy H. Nguyen, Qingyao Liao, Chun-Shien Lu, Isao Echizen "Analysis of Backdoor Attacks on Deepfake Detection" 2024 IEEE International Joint Conference on Biometrics (IJCB) DOI: 10.1109/IJCB62174.2024.10744504

[13] Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu "Contrastive Learning for DeepFake Classification and Localization via Multi-Label Ranking" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) @ IEEE

[14] Li Lin, Xinan He, Yan Ju, Xin Wang, Feng Ding, Shu Hu "Preserving Fair-ness Generalization in Deepfake Detection" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 10.1109/CVPR52733.2024.01591 @ IEEE

[15] Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, Jong-won Choi "Exploiting Style Latent Flows for Generalizing Deepfake Video Detection" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 10.1109/CVPR52733.2024.00114 @ IEEE

Implementation Paper Title:

Deeflyzer: Hybrid Model to Detect Complex Deepfake in Digital Media

Conference Name:

International Conference on Advances and Applications in Artificial Intelligence (ICAAAI2025)

Publication Certificate

Deeflyzer: Hybrid Model to Detect Complex Deepfake in Digital Media

Soham Kolapkar[1][0009-0002-4123-9870], Riya Kshirsagar[2][0009-0008-5893-8022], Charudatta Thakare[3][0009-0009-9504-0681], Shantanu Shinde[4][0009-0003-9195-9937] and Dr. Jitendra Musale[5][0000-0003-0273-4828]

Department of Computer Engineering
ABMSP's Anantrao Pawar College of Engineering and Research, Pune, India.
sohamkolapka4@gmail.com
riya.kshirsagar@gmail.com
charudattathakare1010@gmail.com
shantanums840@gmail.com
jitendra.musale@abmspcoerpune.org

Abstract. The credibility of digital media has been significantly threatened due to advanced Artificial Intelligence techniques which are generally known as Deepfake, endanger the authenticity of audio-visual media. This paper presents an enhanced approach for identifying deepfake. This project utilizes a detection method that compares the area of manipulated face and surrounding areas with InceptionV3 and LSTM which are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) respectively. Our experiments showcased that Wav2Vec 2.0 proved to be optimal for audio spoofing and artificially generated voice. The feature of localization of the fabricated media is implemented to visualize the manipulated portions. For video, datasets like DFDC, FaceForensics++ and Celeb-DF were used to train the model out of which DFDC proved to be compatible. For audio, ASVSpoof 2019, ADD 2022, FAD etc. datasets were used for training and it was clear that the ASVSpoof 2019 was the most suitable. The combination of Inception V3 and LSTM gave the accuracy of 96.17% and Wav2Vec 2.0 would offer a solution with strong potential for high accuracy of 96% in detecting sophisticated audio deepfakes. This method leverages a novel integration of CNN, RNN architecture and achieves state-of-the-art performance in detecting audio and video deepfake.

Keywords: Deepfake, InceptionV3, Wav2Vec 2.0, LSTM, Hybrid-Model.

1 Introduction

Information is highly valued in this modern world. Whether it is an image, text, video, or audio, it gives us knowledge that we utilize in life [1]. So, information is relevant, and so is the authenticity of such information, as false information can put us in a loss position either socially or economically. Hence a deepfake is a technique wherein images, videos, and audio records are modified using a deep learning model [2,3].

Such manipulated media can be used to cause false information in the society. It can be used to damage any reputed personality's reputation. Media, through various methods, can be deepfake, such as face swapping, face synthesis, lip syncing [4,5,6]. Hence it is very crucial to identify those kinds of manufactured media [6]. To do so, a combination of deep learning techniques like Convolutional Neural Network and Recurrent Neural Network can be implemented to identify the deepfakes in digital media [7]. Convolutional Neural Network in short CNN, one of the variants of artificial neural network architecture, is utilized mainly in computer vision and image processing [8].

This makes CNN extremely good for pattern recognition and objects in images [9]. Heavily used in applications of image classification, object detection, and image segmentation [10,11]. It being experimented on all the possible CNNs-RNNs models, Inception-V3 and Long-Short-Term-Memory (LSTM) turned out to be highly interesting combinations to pursue together as a combinatory model altogether [12].

The InceptionV3 is a much more complex model of deep learning for image recognition. It makes use of inception modules, factorization, and batch normalization for the optimization of efficiency and accuracy [13,14]. This algorithm is used widely in image classification, object detection, and segmentation among others. Thus, it turned out to be efficient for extracting spatial features from images [15]. Spatial feature extraction means discovering meaningful information that emerges from images. Meaningful information includes examples such as recognizing shapes, patterns, or textures. This is one of the initial steps that, in most computer vision tasks such as image classification and object detection, is often taken to be a prerequisite [16]. Techniques such as edge detection and corner detection can be used to obtain spatial features [17]. RNN is a type of neural network, particularly designed to process sequential data with a hidden state that feeds on information about previous inputs to understand the context [18].

These have diverse applications such as natural language processing, and time series analysis. Broadly speaking, there are three types of RNNs: simple RNN, LSTM, and GRU [19]. Long-Short-Term Memory or brief LSTM is one specific version of a type of recurrent neural network, an RNN that is designed to avoid vanishing gradients inherent to traditional RNNs [18,19]. It shines to capture long dependencies in sequential data, which makes it very applicable for tasks like NLP, Time Series Analysis & Speech Recognition [20]. LSTMs have a unique architecture with gates that control information flow, making it possible to memorize and process information over long periods; they are, therefore, highly effective in tasks that require the understanding and prediction of patterns in sequential data. Further, the feature which is extracted can be used to the problem of time sequencing in case of LSTM. To this end, there are ample datasets which can be used to train the deep fake detection system. For instance, some of these datasets include: CelebDF, Faceforensics++, DFDC, Google's Deepfake-detection etc. The Deepfake Detection Challenge (DFDC) dataset

consists of multitudes of videos which contain both real and fake samples.

Fig.1. Examples from DFDC dataset

2 Literature Survey

Kuiyuan Zhang [1] and their team members studied about improving the generalization in deepfake detection. In their paper they addressed the issue of catastrophic forgetting which is a concept where a model when learns about new deepfake techniques, it tends to forget its existing or prior knowledge about classical deep-fake techniques. This phenomenon degrades the model's quality significantly.

B.V. Chowdary [2] along with their associates proposed an effective deepfake detection system. They surveyed all the existing model and found out that the combi-nation of CNN and RNN proves to be the most efficient in detecting deepfakes. The proposed system makes use of the ResNet which is a CNN and LSTM which a RNN architecture.

B. Sarada [3] with their colleagues conducted a study on the topic of audio deepfake detection. In their study they proposed a solution in detecting AI cloned voices. They made use of the Generative Adversarial Network (GAN) along with Random Forest which is a machine learning algorithm usually used for classification.

Hao Teng [4] along with their team members proposed a solution on the problem of cross forgery which occurs when a model tries to detect a type of deepfake for which it was not originally trained. To overcome this problem, they suggested the use of extraction of static and dynamic features. Static features will be used to detect the old techniques of deepfake and dynamic features will be used to detect new techniques of deepfake. As per their experiments it is observed that this approach proves to be four times more accurate than single feature extraction models.
Haobo Liang, Yingxiong Leng [5] and their associates studied deeply in the field of face forgery. Their study indicates that the traditional methods for detecting for-gery cannot detect the latest forging techniques. They introduce a novel approach in detecting forgery with knowledge distillation and DCT. It will achieve high amount of accuracy and precision and will be able to detect subtle changes done on the face.

Amaan M. Kalemullah [6] with their co-authors conducted a study in the field of deepfake detection. They mainly focused on detection of deepfake in Human Faces. Their research proposes a comprehensive approach of using Convolutional Neural Network along with two Transfer Learning models ResNet-50 and EfficientNet B7 for detecting inconsistencies in human faces. These models when evaluated gave out the best accuracy in detecting manipulated facial content.

Cheng-Yao Hong [7] and their associates proposed a new way of detecting and identifying deepfakes in images. Their approach to detecting deepfake is such that it divides the images into smaller parts called as patches and then it considers the whole image as a bag of patches. If one patch is manipulated then the whole im-age will be considered as fake and to do that, they used a unique way called the multiple instance learning (MIL). For identifying the specific part of the im-age which has been deepfake it used the multi-label ranking which label all parts of an images and returns the forged part.

Li Lin [8] along with their colleagues provided a solution to a problem is which occurs while detecting deepfakes. In their paper they addressed the issue of fairness generalization. Their study states that existing deepfake models are trained to detect manipulations but they are not efficient in detecting for people from different race and gender. The experiments conducted by them state that using this method the accuracy and effectiveness of the state-of-the-art methods can be easily surpassed.

Jongwook Choi [9] and their team presents a new approach in detecting fake videos based on style latent vectors. Their approach consists of targeting the temporal inconsistencies in fake videos. They have made use of the StyleGRU module which has been trained using contrastive learning used to represent the feature of style latent vectors. They also added a style attention module to detect visual artifacts. After testing this approach on multiple datasets, it is stated that it proves very effective.

Chuangchuang Tan [10] along with their colleagues studied the problems occurring in existing deepfake detection models. These differences are present in the images manipulated or generated using GAN. To tackle this, they introduced a new method Neighboring Pixel Relationships (NPR) which is used to identify these differences. This method showcases a 12.8% increase in the efficient against the state-of-the-art methods.

Trevine Oorloff [11] with their associates introduce a new method called audio visual feature fusion which is learning method divided in two stages and they detect the differences between the audio and visual modalities. In the second stage the representations of the features are tuned and actual deepfake classification is done. This approach deals 98.6% accuracy on the FakeAVCeleb dataset which is a 14.9% gain over the state-of-the-art methods.

Siyou Guo [12] and their team members studied the existing techniques used for deep-fake detection. In their study, they encountered a problem. The existing models used for deepfake detection ignore the subtle variation in the media. To over-come this issue, they proposed a progressive attention network which incorporates two attention modules which are Efficient Multi-Scale Attention Module and Spatial and Channel Attention Module.

3 Research Methodology

3.1. Literature Review:

Existing literature is reviewed extensively to evaluate the current state of deepfake detection systems, and their differences with prior work are identified. While the geometry studied in, and closely related to it, has been well-studied both theoretically and practically, here we want to pro-vide an even more detailed insight into existing models and approaches like those of by defining them via an explicit exponential representation. These gaps will be working as a guiding pathway during the optimization of the proposed system, which uses the inception V3 model combined with LSTM.

3.2. Dataset collection and selection:

Today in a digital era many videos and pictures over the internet are easily manipulated to tessellate it into the form of deep fakes. Deepfake detection systems use deep learning approaches to flag fake content. In this approach, publicly available datasets are used to train a strong detector DFDC, Face Forensics++, Celeb-DF, Google Deep Fake Detection were chosen as the training and testing datasets. They have real media as well as deepfake kind. It also includes multiple sets of manipulated videos/images to train/test and validate the deepfake detection model.

3.3. Model Selection:

The architecture of the model adopted here is hybrid with InceptionV3 utilized to extract spatial features and LSTM used to analyze temporal effects. InceptionV3 is good at capturing fine-grained spatial details in the image frames; LSTM models can effectively model sequential data, thus improving the capacity of our network to recognize anomalies over time in video. That hybrid approach that we just mentioned is thought to be superior to classic single-model architectures.

3.4. Data Preprocessing:

In the case of video-based detection, each video is split up into individual frames which are then standardized and resized to keep a consistent input for our model. Steps like frame extraction, normalization, and resizing that occur due to preprocessing only guarantee high quality data in a consistent format for input into the model thereby increasing accuracy at training time.

3.5. Feature Extraction:

We use InceptionV3 to extract the spatial features in the frames that are important for

detecting artifact manipulation. The features extracted using the STN and its Spatial Transformer Layer are then pushed into the LSTM layer for similar time-based feature analysis to map temporal inconsistency. Wav2Vec 2.0 features are extracted from the raw audio by feeding the waveform through a convolutional neural network (CNN) to extract local patterns and short-term dependencies. At this first step, the audio is represented at a low level and quantized to discrete codebook representations.

3.6. Evaluation of Model:

Metrics like accuracy, precision, recall and F1-score derived from a validation dataset will be finally used to evaluate the performance of the hybrid model (InceptionV3 + LSTM). Since the performance of the system will be monitored at each stage to tune it in, so that it achieves a desired accuracy of detection as designed. After fine-tuning, evaluate model performance on the test data using evaluation metrics that were chosen.

4 Algorithm

I. **Input:** To start with, the video is received as an input.
II. **Separation of Audio and Video:** In the culmination of the process, audio and video are extracted from the video.
III. **Preprocessing of Video:** Moving the video and getting the video frames out of each frame. The video frames are then normalized so; this is simply the length caused by the video codec i.e. in the same length of the video process.
IV. **Preprocessing of Audio:** The audio is split into smaller pieces which facil-itate audio measurement.
V. **Feature Extraction:** The video frames are processed through InceptionV3 for feature extraction. Wav2Vec 2.0 is implemented on the audio clips to extract the features.
VI. **Temporal Analysis and Inconsistencies**: LSTM is implemented to inspect the video features extracted from the time perspective. The MFA is implemented for the analysis of the extracted features of the audio to review in-consistencies.
VII. **Deepfake Classification:** Once the video and audio analyses are conclude-ed, the processes check whether the face is a deepfake or a real person.
VIII. **Final Prediction:** The final state prediction is offered as the first suggestion that the input video clip has a high probability of being a deepfake.

5 Flow Diagram of Proposed Work

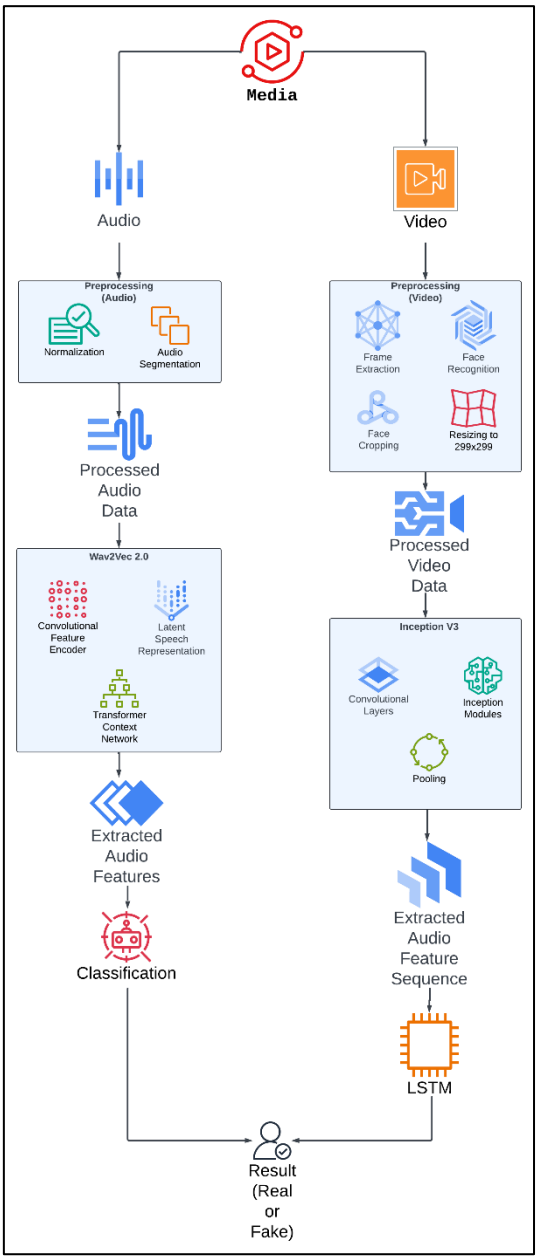


Fig.2. Flow diagram of Proposed System

6 Proposed System

Models such as InceptionV3 and LSTM along with Wave2Vec 2.0 are used in the proposed system as they proved to be the most efficient when combined. For deepfake detection in video and images, InceptionV3 which is a CNN and LSTM which is a RNN are used as well as for audio Wave2Vec 2.0 is used which detects AI cloned voices as well.

7.1. Video Deepfake Detection:

For detecting deepfakes in video, the dataset used to train the model is the DFDC dataset by Facebook (now Meta) which contains more than 10000+ real and fake videos. Although to train the model using this dataset, the data needs to be preprocessed due to the immense amount of time that it would take to train without processing. There are certain steps carried out in the preprocessing stage.

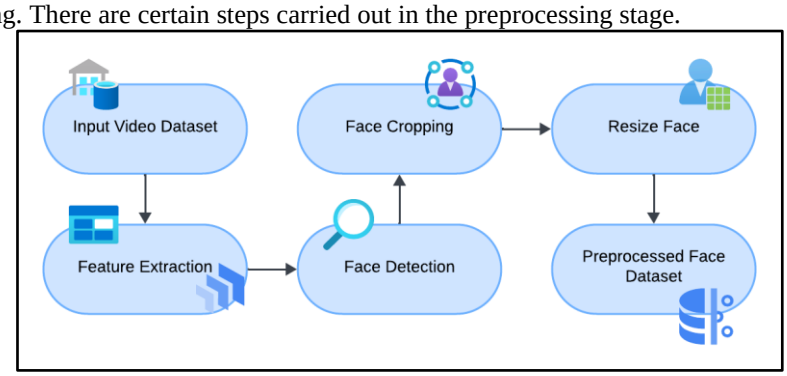


Fig.3. Pre-processing

Frame Extraction: Frames from videos are extracted and stored in the form of images to simplify the learning process.

Face Detection: Faces from those images are detected using computer vision which is a crucial step.

Face cropping and resizing: The face part of the image is cropped and resized to 299x299 which is an ideal size for InceptionV3 to be able to learn.

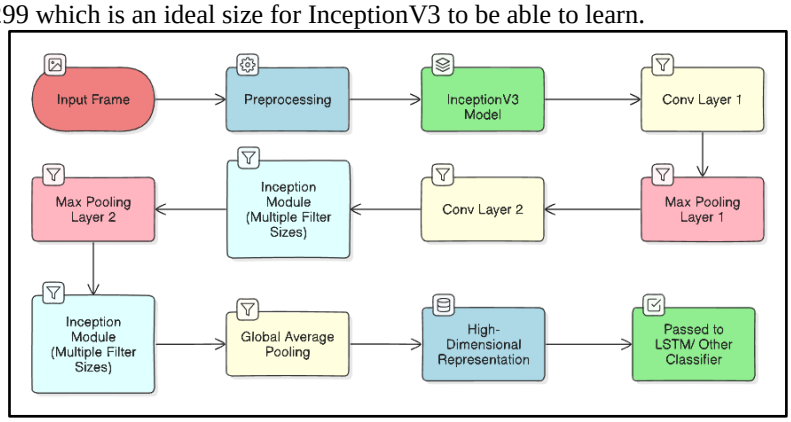


Fig.4. Working of InceptionV3

A Convolutional Neural Network, InceptionV3 — this is widely used for feature extraction on images. To detecting deepfakes, it learns spatial features such as textures, expressions, and skin colors which help in capturing even least noticeable artefacts left by deepfakes.

Conv Layer 1: In this layer various filters are applied to the image to capture features like curves, edges etc. and creates a feature map.

Max Pooling Layer 1: This layer performs the task of dimensionality reduction for the feature map decreasing the computational cost and making the network less susceptible to minute spatial changes.

Inception Module*:* It is the main component of InceptionV3 which uses filters to process the input image and captures feature from it. These three tasks are repeated as per the requirement constructing the neural network.

Global Average Pooling*:* Here the average of all the feature maps is taken which considerable reduces the size of feature maps and only important features are restored. The output is the form of a feature vector which has high dimensions and contains the most crucial features.

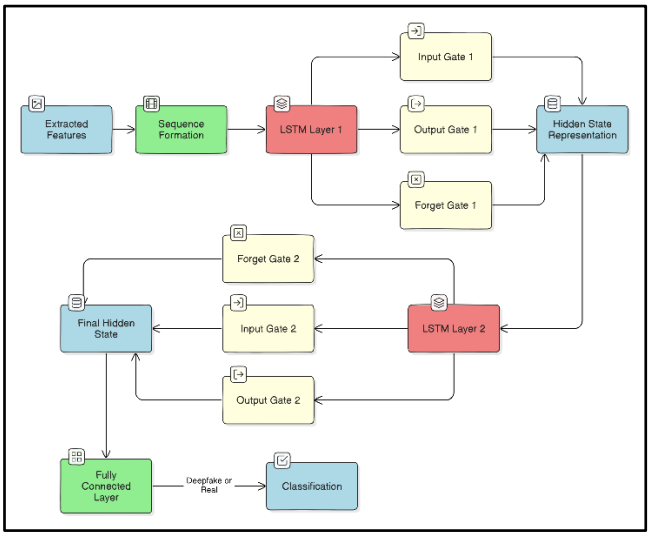


Fig.5. Working of LSTM

LSTM is a Recurrent Neural Network which is generally used for speech recognition and time series analysis. It is also used as a type of classifier in some use cases. In the context of deepfake detection, LSTM is used to perform temporal analysis over video frames by learning their sequence. As it works on sequential data, it is used here to analyze the feature vectors provided by InceptionV3 and learn how they change over time. Deepfake introduces temporal inconsistencies between frames and these irregularities can be detected using LSTM. After learning from the sequence of frames, the output is a hidden state which summarizes all the information gained by LSTM. The hidden state is given to a fully connected layer or classifier which uses the information to gives the final prediction.

The tasks performed by LSTM are as follows:
Sequence Creating: The feature vector received from InceptionV3 are organized as a sequence and where each feature vector is a frame from the video. This serves as a input to the LSTM.
Forget Gate: This gate in the LSTM determines which information is to be kept and which is to be discarded hence retaining only important information and making the process more efficient.
Input Gate: This gate decides which new information is to be added to the internal state.
Output Gate: This gate will control which section of the internal state shall be outputted as the hidden state as it is to be used for further processing. The output of this gate is going to be a hidden state representation.
LSTM layer: This layer refines the temporal data to create information which will be useful for effective classification.
Final Hidden State: After all the processing the LSTM network generates a hidden state which contain all the temporal information of the video sequence along with the critical features required for classification.
Full Connected Layer: This layer takes the hidden state as input and converts the information in a form which is suitable to perform classification on it.
Classification: The output of fully connected layer is given to a sigmoid or softmax function which classifies or predicts the output.

7.2. Audio Deepfake Detection:

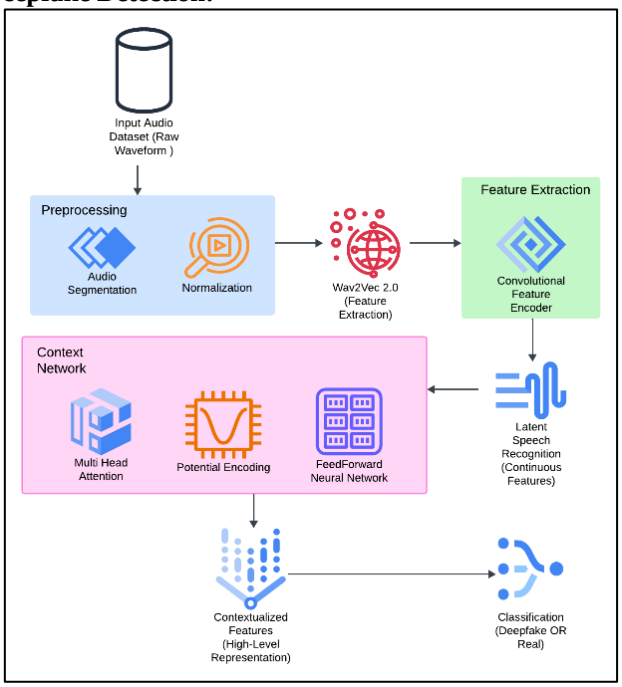


Fig.6. Working of Wav2Vec 2.0

Wav2Vec 2.0 is a self-supervised speech representation model by Meta which has the capability to extract features from raw audio. In the context of deepfake detection, it is used to extract latent speech representation from raw audio which will help in audio deepfake detection. As audio deepfakes contain subtle inconsistencies or distortion, it is difficult to detect these with traditional methods. Wave2Vec uses convolutional feature encoder and transformer context network which is beneficial in detecting these subtle inconsistencies.

The processing for Wav2Vec 2.0 is as follows:

Pre-processing: In this step the preprocessing which audio segmentation and normalization is carried out convert the raw audio into segments and normalize them into a common range to improve the accuracy and make training efficient.

Feature Extraction: The processed audio goes through multiple convolutional layers. These layers capture speech features and temporal patterns. The output of this step is a latent speech representation.

Context Network (Transformer): This network processes the latent features. The focus is equally divided on different parts of audio to maintain the order of the features. The output of this is a representation of contextually rich features.

Classification: These features are passed to the classifier and the resulting output is real or fake.

Overall combining all these processes, we obtain a robust Deepfake detection system which is capable of detecting audio as well as video deepfakes.

7.3. Advantages of Proposed Model:

Multimodal Detection. Our system was mainly enhanced by the usage of multiple models to detect deepfakes in digital media. The use of advanced and highly integrated techniques makes synthetic media detection easier with accuracy.

Advanced Deep Learning Techniques. A component of the CNN is applied to extract spatial features and RNN is applied to process temporal sequence detection. The model captures nature of the media.

Innovative Audio Analysis. The implementation of Wave2Vec 2.0 increases the accuracy and precision of the model significantly.

Inception V3 with LSTM Integration. The combined model has shown great accuracy over the state-of-art approaches.

Manually Altered Part Identifier. The proposed model will identify the given media, regardless of format i.e. audio, video, or images whether that the given media is manipulated or real, and if the given image or video is synthetic then it will also give the part of actual media, which are faulty.

7.4. Localization:

The integrated model is trained to identify the deepfake in both audio and video at the same time providing localization of the manipulated portion. This is achieved by doing modifications while training the models and allowing them to flag the manipulat-

ed frames for video and segments for audio. After the analysis these tampered frames are highlighted and displayed along with the result real or fake.

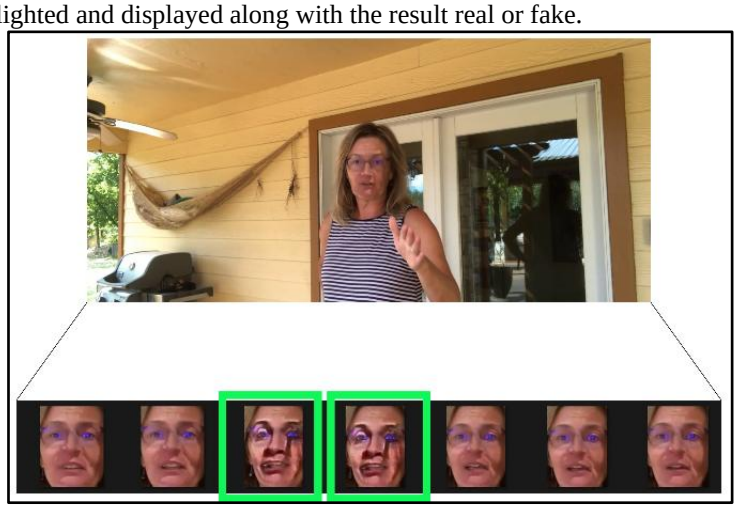


Fig.7. Localization Example

7 Result and Discussion

The DFDC dataset was used to train our InceptionV3-LSTM model as it contained more diverse video samples and helped improve accuracy over other datasets. We began by training for more than 20 epochs, but this model trains too overfit. So, we are going with 10 after all the finetuning. The model used a sigmoid Activation Function, and classified as 1 means fake, 0 means real. This has yielded a final test accuracy of 94.80% and the problem was completely free from overfitting unlike other classifier models used. Also, the same was tested on a test set of audios and came out to provide an accuracy of almost 93%, showing even Wave2Vec is good.

8.1. Training Phase

The combination of *InceptionV3 and LSTM* were trained with 10 epochs which gave out the accuracy of 96.6% on training data. Whereas the validation accuracy was *94.80%* with a loss of 0.1523. This accuracy is better than most of the existing models. It means that this model will classify 94% of deepfakes correctly. As only face extracted images are used to train this model, its precision has increased significantly and it can detect multiple types of deepfakes.

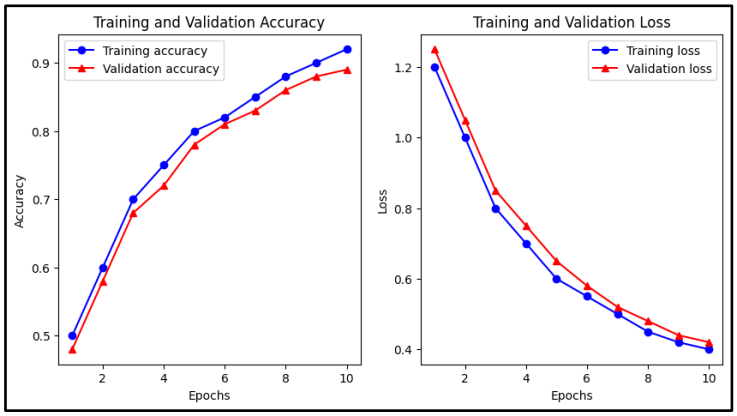


Fig.8. Training and Loss Curve

As we can see from Fig. that the training and validation accuracy of the model is very close to each other representing that the model is neither overfitted nor underfitted. The model has the optimal level of bias and variance resulting in better prediction accuracy. Likewise, the loss also has decreased with increasing epochs. A dropout layer was added to avoid the overfitting.

Wav2Vec 2.0 is a self-supervised learning framework for speech representation learning. It is trained on raw and labeled audio data which was categorized as real and fake. It utilizes a convolutional feature encoder to extract latent representations from raw waveforms. A classifier head is added to predict the authenticity of audio samples. Here are some of the hyperparameters which were adjusted to obtain the maximum accuracy:

Learning rate. Cosine annealing a learning scheduler was implemented to stabilize the learning rate.

Batch size. The batch size was set to 64 as per the GPU memory available which proved to be enough.

Dropout. The dropout rate was set to 0.2 to avoid overfitting of the model.

The challenges faced during the training was the presence of noise in training data and the disproportion or imbalance in the training data. To tackle these challenge denoising algorithms were applied to make the training process more accurate and consistent.

8.2. Comparison

Table. 1 Comparison of Metrics

Model	Dataset Used	Performance Metric		
		Accuracy	Precision	Recall
InceptionV3+ LSTM	DFDC	96.17%	89.11%	91.1%
	FF++	95.11%	87.45%	86.5%
	Celeb-DF	95.43%	88.15%	84.1%
EfficientNet+ LSTM	DFDC	95.64%	82.41%	83.2%
	FF++	93.24%	83.17%	82.4%
	Celeb-DF	96.09%	85.72%	84.6%
ResNet+ LSTM	DFDC	91.79%	82.64%	87.8%
	FF++	89.44%	84.54%	82.7%
	Celeb-DF	93.82%	87.15%	89.4%

The above comparison shows that the integration of InceptionV3 with LSTM when tested on the DFDC dataset proved to be having highest accuracy. This integration topped some of the state-of-the-art methods used.

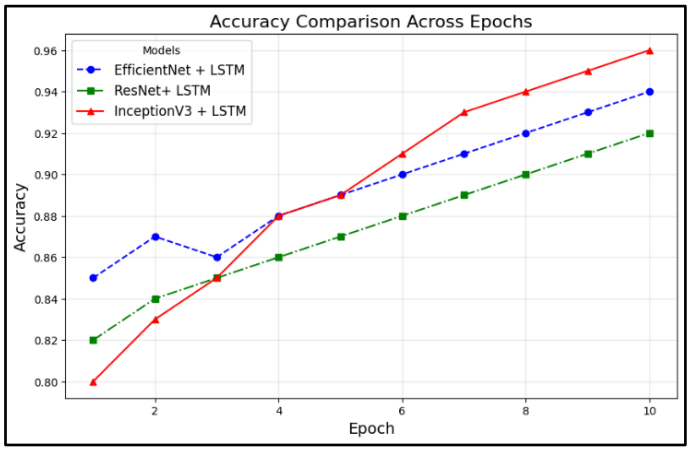


Fig. 9 Accuracy comparison

The above figure shows the accuracy graph with respect to epochs where the proposed model has a better accuracy than the existing models like ResNet and EfficientNet. Hyperparameter adjustment and adding a dropout layer has increased the accuracy of the model while avoiding overfitting.

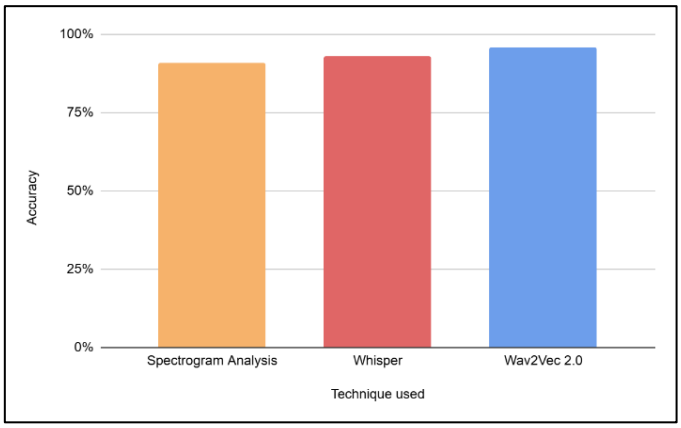


Fig. 10 Comparison of Audio Models

The graph represents the accuracy score of each technique tested on the ASVSpoof 2019 dataset which was the most suitable dataset in terms of quality and quantity of real and fake audio samples. The audio model Wav2Vec 2.0 outperformed the existing techniques such as spectrogram analysis. The main benefit of Wav2Vec 2.0 was its ability to detect subtle changes in tone and pitch of the voice. The accuracy of the model was of 96% which was a 3% increase in the state-of-the-art methods. The difficulties encountered while training the model like noisy data were overcame using noise reduction algorithms.

8 Conclusion

This project successfully implemented a hybrid deepfake detection system using both visual (CNN-based) and audio (Wav2Vec 2.0-based) modalities to identify manipulated media with high accuracy and robustness. By leveraging advanced deep learning architectures, such as CNNs for visual feature extraction and Wav2Vec 2.0 for audio analysis, the system demonstrated its ability to detect deepfakes in diverse scenarios, including sophisticated audio-visual manipulations like real-time voice cloning and face swapping. The integration of state-of-the-art models, including InceptionV3 and LSTMs, allowed for effective feature extraction and sequence modelling, enabling precise detection and localization of manipulations. Incorporating datasets like DFDC, FaceForensics++ and Celeb-DF, highlighted the system's scalability and generalizability. Performance benchmarking against other approaches provided a comprehensive evaluation, showcasing the novelty and competitive edge of the proposed model. To address overfitting and improve model robustness, techniques such as dropout, data augmentation, and adversarial testing were implemented, enhancing real-world applicability. Furthermore, computational analysis of training and inference times affirmed the system's feasibility for real-time applications, crucial for practical deployment in fields like digital forensics, media verification, and cybersecurity. Further this system can be fine-tuned to implement real time deepfake detection as well as it can be made compatible for mobile devices by decreasing the computations.

References

1. Kuiyuan Zhang, Zeming Hou, Zhongyun Hua, Yifeng Zheng, Leo Yu Zhang "Boosting Deepfake Detection Generalizability via Expansive Learning and Confidence Judgement" in 2024 IEEE Transactions on Circuits and Systems for Video Technology 10.1109/TCSVT.2024.3462985 @ IEEE.
2. B.V. Chowdary, Marry Prabhakar, Mavoori Akhil, Komirishetty Pavan, B. Pavana Teja Reddy "Deep Fake Detection using Adversarial Ensemble Techniques" 2024 8th International Conference on Inventive Systems and Control (ICISC) 10.1109/ICISC62624.2024.00041 @ IEEE
3. B. Sarada, TVS. Laxmi Sudha, Meghana Domakonda, B. Vasantha "Audio Deepfake Detection and Classification" 2024 Asia Pacific Conference on Innovation in Technology (APCIT) 10.1109/APCIT62007.2024.10673438 @IEEE
4. Hao Teng, Chia-Yu Lin "Dynamic and Static Features Extraction for Deep-fake Detection" 2024 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan) 10.1109/ICCE-Taiwan62264.2024.10674402 @ IEEE
5. Haobo Liang, Yingxiong Leng, Jinman Luo, Jie Chen, Xiaoji Guo "A Face Forgery Video Detection Model Based on Knowledge Distillation" 2024 IEEE/ACIS 27th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) @ IEEE
6. Amaan M. Kalemullah, Prakash P, Sakthivel V "Deepfake Classification for Human Faces using Custom CNN" 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT) @ IEEE
7. Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu "Contrastive Learning for Deep-Fake Classification and Localization via Multi-Label Ranking" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) @ IEEE
8. Li Lin, Xinan He, Yan Ju, Xin Wang, Feng Ding, Shu Hu "Preserving Fair-ness Generalization in Deepfake Detection" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 10.1109/CVPR52733.2024.01591 @ IEEE
9. Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, Jong-won Choi "Exploiting Style Latent Flows for Generalizing Deepfake Video Detection" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 10.1109/CVPR52733.2024.00114 @ IEEE
10. Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, Yunchao Wei "Rethinking the Up-Sampling Operations in CNN-Based Generative Network for Generalizable Deepfake Detection" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) @ IEEE
11. Trevine Oorloff, Surya Koppisetti, Nicolò Bonettini, Divyaraj Solanki, Ben Colman, Yaser Yacoob, Ali Shahriyari, Gaurav Bharaj "AVFF: Audio-Visual Feature Fusion for Video Deepfake Detection" 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 10.1109/CVPR52733.2024.02559 @ IEEE
12. Siyou Guo, Mingliang Gao, Qilei Li, Gwanggil Jeon, David Camacho "Deepfake Detection via a Progressive Attention Network" 2024 International Joint Confer-

ence on Neural Networks (IJCNN) 10.1109/IJCNN60899.2024.10650463 @ IEEE.

13. Prakash Raj S, Pravin D, Sabareeswaran G, Sanjith R K; Gomathi B, “Deepfake Detection Using Deep Learning” in ICACCS, Coimbatore, 2024, DOI: 10.1109/ICACCS60874.2024.10717155
14. Pham Minh Thuan, Bui Thu Lam, Pham Duy Trung, “Spatial Vision Transformer: A Novel Approach to Deepfake Video Detection” in VCRIS, Hanoi, 2024, DOI: 10.1109/VCRIS63677.2024.10813391
15. Aung Kyi Win, Myo Min Hein, Chit Htay Lwin, Aung Myo Thu, Myo Myat Thu, Nu Yin Khaing, “A Novel Methodology for Deepfake Detection Using MesoNet and GAN-based Deepfake Creation” in ICAIT, Yangon, 2024, DOI: 10.1109/ICAIT65209.2024.10754912
16. Sornavalli G, Priyanka Vijaybaskar, “DeepFake Detection by Prediction of Mismatch Between Audio and Video Lip Movement” in ADICS, Chennai, 2024, DOI: 10.1109/ADICS58448.2024.10533515
17. Naveed Ur Rehman Ahmed, Afzal Badshah, Hanan Adeel, Ayesha Tajammul Ali Duad, Tariq Alsahfi, “Visual Deepfake Detection: Review of Techniques, Tools, Limitations, and Future Prospects” in IEEE Access, 2024.
18. Jitendra Chandrakant Musale, Anuj Kumar Singh “Effective face recognition with hybrid distance key frame selection using tbo-unesamble model” International Journal of Wavelets, Multiresolution and Information Processing 2024-03 (IJWMIP). 2024, 10.1142/S0219691323500443
19. Jitendra Chandrakant Musale, Anuj Kumar Singh, Swati Shirke “Tri bird technique for effective face recognition using Deep Convolutional Neural Network” Atlantis Highlights in Computer Sciences; Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACCS 2023). 2023, 10.2991/978-94-6463-314-6_33
20. Alexandre Libourel, Sahar Husseini, Nelida Mirabet-Herranz, Jean-Luc Dugelay “A Case Study on how Beautification Filters Can Fool Deepfake Detectors” in IWBF, Enschede, 2024, DOI: 10.1109/IWBF62628.2024.10593932

Reviewer Comments

Reviewer #1

Comment:

1. Include at least 20 references and ensure all are cited in the text and in sequence.

Reviewer #2

Comment:

1. Abstract should be within 150–200 words and include objectives, methodology, key findings, and significance.
2. Expand the Introduction section with detailed background, significance, and clear objectives.
3. Provide more comprehensive and relevant content in the Literature Review. Add a methodology diagram for better visualization.
4. Include tables and graphs in the Results section to present findings effectively.
5. Conclusion should be concise and within 150–200 words, summarizing findings and future work.
6. Include at least 20 references and ensure all are cited in the text.
7. Cite all tables and figures appropriately in the text.
8. Format the paper according to the template.
9. Ensure the paper is a minimum of 10 pages.

Reviewer #3

Comment:

1. Citations are not in the sequence.

Annexure D

Plagiarism Report

Tanaji Mali

859_250602_153006.pdf

01

Turnitin 01

Politeknik Manufaktur Negeri Bangka Belitung

Document Details

Submission ID
trn:oid:::1:3266729839

Submission Date
Jun 2, 2025, 5:04 PM GMT+7

Download Date
Jun 2, 2025, 5:06 PM GMT+7

File Name
859_250602_153006.pdf

File Size
370.4 KB

7 Pages

4,453 Words

28,629 Characters

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

18 Not Cited or Quoted 5%
Matches with neither in-text citation nor quotation marks

3 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

4% Internet sources
5% Publications
2% Submitted works (Student Papers)

Match Groups

- **18** Not Cited or Quoted 5%
  Matches with neither in-text citation nor quotation marks
- **3** Missing Quotations 1%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 4% Internet sources
- 5% Publications
- 2% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
Henderson State University		1%
2	Internet	
frcrce.ac.in		<1%
3	Publication	
Thangaprakash Sengodan, Sanjay Misra, M Murugappan. "Advances in Electrical ...		<1%
4	Publication	
Fatma M. Talaat, Ahmed R. Elnaggar, Warda M. Shaban, Mohamed Shehata, Most...		<1%
5	Internet	
www.sciencepg.org		<1%
6	Internet	
www.jetir.org		<1%
7	Student papers	
University of Arizona		<1%
8	Publication	
Lam Pham, Phat Lam, Dat Tran, Hieu Tang, Tin Nguyen, Alexander Schindler, Flori...		<1%
9	Internet	
bmva-archive.org.uk		<1%
10	Publication	
"Information and Communication Technology", Springer Science and Business M...		<1%

11 Publication

Yang Yu, Rongrong Ni, Siyuan Yang, Yu Ni, Yao Zhao, Alex C. Kot. "Mining General... <1%

12 Internet

arxiv.org <1%

13 Internet

largeaudiomodel.com <1%

14 Internet

www.frontiersin.org <1%

15 Publication

Khanh-Duy Cao-Phan, Quan Tran Dinh Dai, Van-Linh Nguyen. "Vdd: voice deepfak... <1%

16 Publication

V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challeng... <1%

Tanaji Mali

ICAAAI.pdf

02

Class N

Politeknik Manufaktur Negeri Bangka Belitung

Document Details

Submission ID
trn:oid:::1:3133190661

Submission Date
Jan 20, 2025, 1:53 PM GMT+7

Download Date
Jan 20, 2025, 1:54 PM GMT+7

File Name
ICAAAI.pdf

File Size
679.4 KB

17 Pages

5,311 Words

28,826 Characters

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

11 Not Cited or Quoted 2%
Matches with neither in-text citation nor quotation marks

6 Missing Quotations 1%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
2% Publications
1% Submitted works (Student Papers)

Match Groups

- **11** Not Cited or Quoted 2%
  Matches with neither in-text citation nor quotation marks
- **6** Missing Quotations 1%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
2% Publications
1% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
Universitat Politècnica de València		<1%
2	Internet	
ebin.pub		<1%
3	Internet	
www.nature.com		<1%
4	Internet	
ijrpr.com		<1%
5	Publication	
Rajendra Rana Bhat, Rodrigo Daniel Trevizan, Rahul Sengupta, Xiaolin Li, Arturo B...		<1%
6	Internet	
www.ijitee.org		<1%
7	Internet	
patents.glgoo.top		<1%
8	Publication	
Xinyuan Song, Keyu Chen, Ziqian Bi, Qian Niu, Junyu Liu, Benji Peng, Sen Zhang, ...		<1%
9	Publication	
Yang Yu, Rongrong Ni, Siyuan Yang, Yu Ni, Yao Zhao, Alex C. Kot. "Mining General...		<1%
10	Internet	
arxiv.org		<1%

11 Internet

dokumen.pub <1%

12 Internet

link.springer.com <1%

13 Publication

"Computer Vision – ECCV 2022", Springer Science and Business Media LLC, 2022 <1%

14 Publication

"Computer Vision – ECCV 2024", Springer Science and Business Media LLC, 2025 <1%

Annexure E

Information of Project Group Members

Information about Group Members

- Name : **Soham Vijay Kolapkar**
- Date of Birth : 01/12/2003
- Gender : Male
- Email-id : sohamkolapka4@gmail.com
- Address : Gokhalenagar, Senapati Bapat Road , Pune
- Mobile Number : 9307296260

- Name : **Riya Girish Kshirsagar**
- Date of Birth : 26/11/2003
- Gender : Female
- Email-id : riya.kshirsagar@gmail.com
- Address : Laxminagar, Parvati, Pune
- Mobile Number : 9822668997

- Name : **Charudatta Sunil Thakare**
- Date of Birth : 10/10/2003
- Gender : Male
- Email-id : charudattathakare1010@gmail.com
- Address : Manjari , BK , Pune
- Mobile Number : 8237398490

- Name : **Shantanu Manoj Shinde**
- Date of Birth : 17/03/2003
- Gender : Male
- Email-id : shantanums840@gmail.com
- Address : Dhankawadi, Pune
- Mobile Number : 9607878383

ANNEXURE F

Project Review PPT

10-06-2025

Deeflyzer: Hybrid Model to Detect Complex Deepfake in Digital Media

Name	Exam No.
SOHAM VIJAY KOLAPKAR	B401100154
RIYA GIRISH KSHIRSAGAR	B401100155
CHARUDATTA SUNIL THAKARE	B401100190
SHANTANU MANOJ SHINDE	B401100178

Guide: Dr. Jitendra Musale

Agenda:

- Introduction
- Problem Definition
- Literature Survey
- Solution
- Methodology
- Proposed Method
- Mathematical Model
- Results
- Conclusion

Introduction:

What are Deepfakes?

- AI-generated media that looks real but is artificially manipulated.

How are Deepfakes Created?

- Generative Adversarial Networks (GANs) train **generator** (creates fakes) **discriminator** (detects fakes).
- Raises ethical and security concerns like misinformation, identity theft, and fraud.

Need for Deepfake Detection

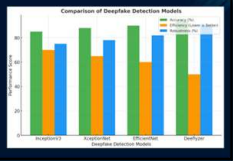
- Advanced generative techniques make deepfakes harder to detect.
- Researchers are developing AI-based detection methods to combat misuse.

Problem Definition

- Deepfake technology, powered by **Generative Adversarial Networks (GANs)**, has advanced significantly, making it increasingly difficult to distinguish between real and fake media.
- While it has innovative applications in entertainment, education, and virtual reality, its misuse presents serious threats, including:

1. **Misinformation & Fake News** – Spreading false information with realistic visuals.
2. **Identity Theft & Fraud** – Manipulating voices and faces for scams.
3. **Political Manipulation** – Altering videos to mislead the public.
4. **Cybersecurity Risks** – Exploiting AI to bypass security measures.

- Advanced deepfake techniques outpace traditional detection, requiring robust AI-based solutions for effective identification.



1

Literature Survey:

Sr. No.	Paper Name	Year	Scopus	Challenges
1.	Boosting Deepfake Detection Generalizability via Expansive Learning and Confidence Judgement	2024	IEEE	Not Suitable for all Deepfake Generation Techniques
2.	Deep Fake Detection using Adversarial Ensemble Techniques	2024	IEEE	Prone to Adversarial Attacks
3.	Audio Deepfake Detection and Classification	2024	IEEE	Cannot Classify AI Generated Voice
4.	Dynamic and Static Features Extraction for Deepfake Detection	2024	IEEE	Feature Extraction Varies due to Video Condition
5.	A Face Forgery Video Detection Model Based on Knowledge Distillation	2024	IEEE	Computational Cost reduction affecting accuracy
6.	Deepfake Classification for Human Faces using Custom CNN	2024	IEEE	Inefficiency of datasets

Literature Survey:

Sr. No.	Paper Name	Year	Scopus	Challenges
7.	Contrastive Learning for DeepFake Classification and Localization via Multi-Label Ranking	2024	IEEE	Localization causes excessive computational cost
8.	Preserving Fairness Generalization in Deepfake Detection	2024	IEEE	Bias in deepfake detection models with unfair performance
9.	Exploiting Style Latent Flows for Generalizing Deepfake Video Detection	2024	IEEE	Poor adaption to unseen deepfake techniques
10.	Rethinking the Up-Sampling Operations in CNN-Based Generative Network for Generalizable Deepfake Detection	2024	IEEE	CNN-based up-sampling introduces artifacts that may affect detection models
11.	AVFF: Audio-Visual Feature Fusion for Video Deepfake Detection	2024	IEEE	Cannot synchronize audio visual detection
12.	Spatial Vision Transformer: A Novel Approach to Deepfake Video Detection	2024	IEEE	High Computational Costs

Solution

- A system which is capable of classifying audio as well as video media as Real or Fake using state-of-the-art deep learning techniques.
- We have combined the **InceptionV3 (CNN)** and **LSTM (RNN)** for Video Deepfake Detection whereas **Wav2Vec 2.0** was implemented of Audio Detection.

Audio Detection

- Model Used: Wav2Vec 2.0
- Dataset Used: ASVspoof

Video Detection

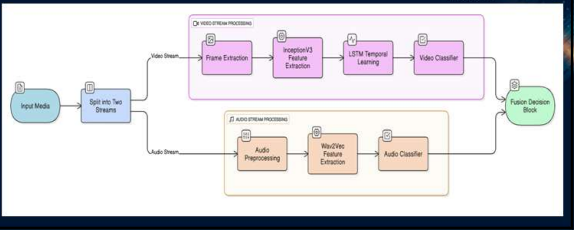
- CNN: InceptionV3
- RNN: LSTM
- Dataset Used: DFDC

Methodology:

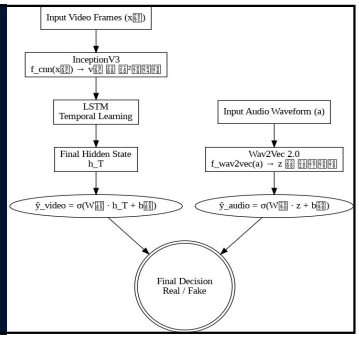
The methodology integrates deep learning architectures to classify media as either real or fake, focusing on both video and audio inputs.
The approach consists of:

- Dataset Selection:
  Video: DFDC (Deepfake Detection Challenge)
  Audio: DFDC & ASVSpoof 2019
- Model Design:
  Video Detection: Combines InceptionV3 (CNN for feature extraction) and LSTM (for sequential frame learning).
  Audio Detection: Utilizes Wav2Vec 2.0 model, which is a self-supervised model for audio feature extraction and classification.
- Training:
  Models are trained separately for audio and video.
  Evaluation conducted on unseen data to assess generalization.
- Localization: The system highlights manipulated regions in video/audio data for forensic insights.

Proposed Method



Mathematical Model

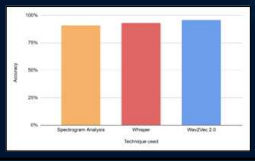


Experimentation Results:

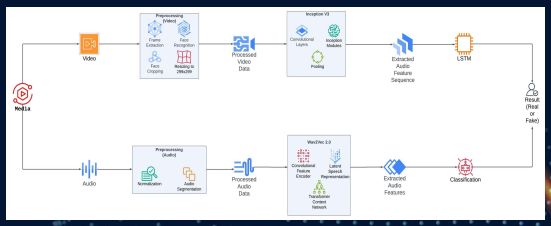
Multiple Models for audio as well as video Deepfake Detection were tested across multiple datasets.

Out of which the combination of **InceptionV3** and **LSTM** performed the best on **DFDC** dataset whereas **Wav2Vec 2.0** had the most accuracy when trained with **ASVSpoof 2019** dataset

Model	Dataset Used	Performance Metric		
		Accuracy	Precision	Recall
InceptionV3+ LSTM	DFDC	96.17%	89.11%	91.1%
	FF++	95.11%	87.45%	86.5%
	Celeb-DF	95.43%	88.15%	84.1%
EfficientNet+ LSTM	DFDC	95.64%	82.41%	83.2%
	FF++	93.24%	83.17%	82.4%
	Celeb-DF	96.09%	85.72%	84.6%
ResNet+ LSTM	DFDC	91.79%	82.64%	87.8%
	FF++	89.44%	84.54%	82.7%
	Celeb-DF	93.82%	87.15%	89.4%

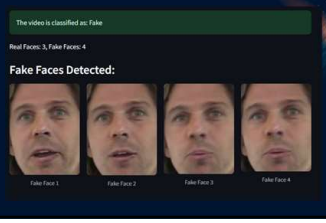


Flow Diagram:



Metrics and Localization:

- The Hybrid Video Model had the accuracy of **96.17%** whereas the Audio Model gave out the accuracy of **96%** on unseen data. The feature of Localization has also been added .
- Localization refers to the process of highlighting the manipulated area in an image or audio.



Results

Conclusion

- A **hybrid deepfake detection framework** combining state-of-the-art models for both audio and video has been presented.
- The **dual-modality** detection increases robustness and accuracy.
- **Localization** allows forensic insight by visually or acoustically marking fake segments.
- The model demonstrated high accuracy on real-world datasets like DFDC and ASVSpoof 2019, showing its applicability in practical security systems.

THANK YOU

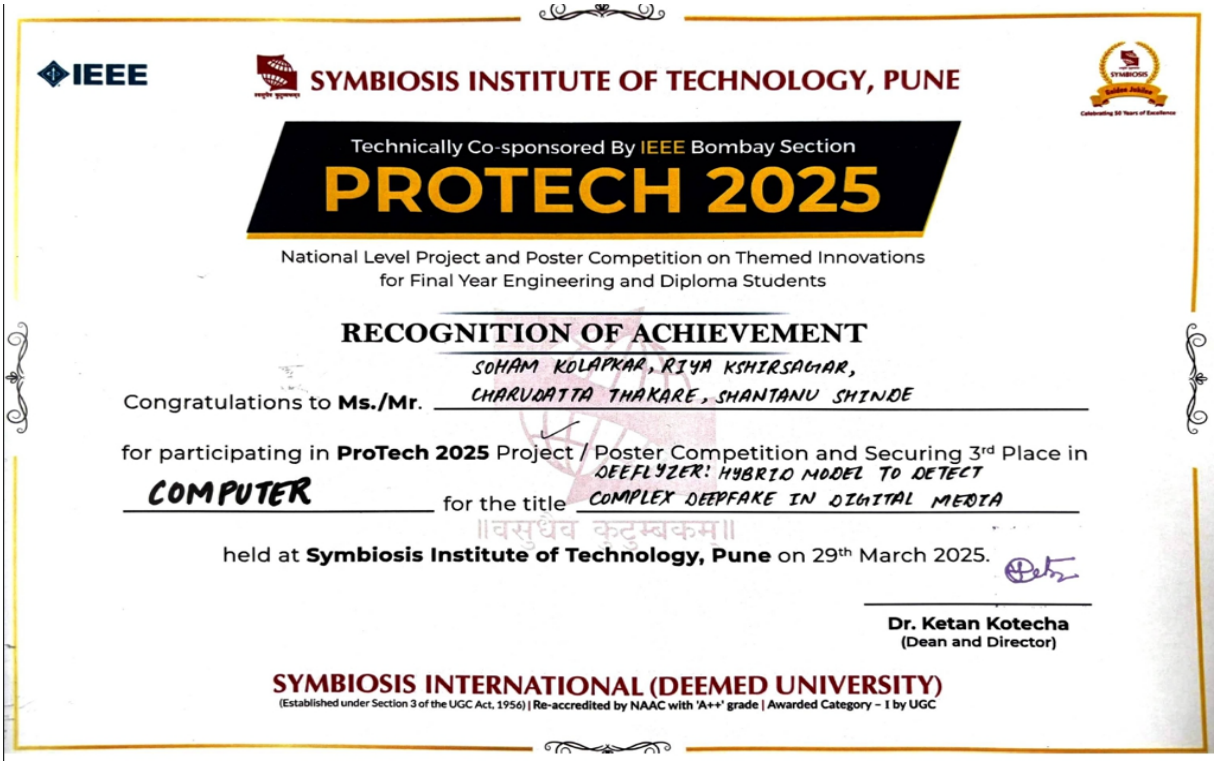
4

ANNEXURE G

Project Achievements

Competition Name: **SIT Protech 2025**

2nd Runner-Up



IEEE

SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE

Technically Co-sponsored By IEEE Bombay Section

PROTECH 2025

National Level Project and Poster Competition on Themed Innovations for Final Year Engineering and Diploma Students

RECOGNITION OF ACHIEVEMENT

Congratulations to Ms./Mr. SOHAM KOLAPKAR, RIYA KSHIRSAGAR, CHARUDATTA THAKARE, SHANTANU SHINDE

for participating in ProTech 2025 Project / Poster Competition and Securing 3rd Place in COMPUTER for the title DEEFLYZER: HYBRID MODEL TO DETECT COMPLEX DEEPFAKE IN DIGITAL MEDIA

॥वसुधैव कुटुम्बकम॥

held at Symbiosis Institute of Technology, Pune on 29th March 2025.

Dr. Ketan Kotecha
(Dean and Director)

SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)
(Established under Section 3 of the UGC Act, 1956) | Re-accredited by NAAC with 'A++' grade | Awarded Category – I by UGC