Deepfake Forgery Detection using Dual Model

Submitted in partial fulfilment of the requirements of
the degree of

Bachelor of Engineering

by

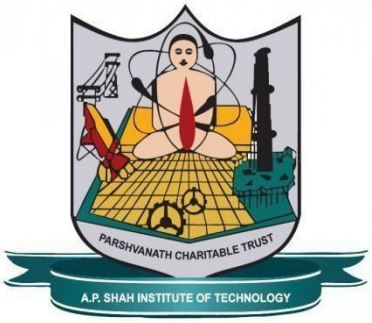
Shraddha Barge: 23106085

Yash Chalke: 23106027

Moksh Khule: 23106095

Deep Magar : 23106062

Project Guide:
Prof. Shubham Zanwar



Department of Computer Science & Engineering
(Artificial Intelligence & Machine Learning)

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

UNIVERSITY OF MUMBAI

(2025-2026)

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled **"Deepfake Forgery Detection using Dual Model "** is a bonafide work of **Shraddha Barge (23106085), Yash Chalke (23106027), Moksh Khule (23106095), Deep Magar (23106062)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning**)

Prof. Shubham Zanwar	Prof. Yogeshwari Hardas Project Guide Project Co-Ordinator
Dr. Jaya Gupta Head of Department	Dr. Uttam D Kolekar Principal

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval for T. E.

This project report entitled ***Deepfake Forgery Detection using Dual Model*** by **Shraddha Barge, Yash Chalke, Moksh Khule, Deep Magar** is approved for the degree of ***Bachelor of Engineering*** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning)**, ***2025-26***.

Examiner Name	Signature
1. _____	_____
2. _____	_____

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

Shraddha Barge

23106085

Yash Chalke
23106027

Moksh Khule
23106095

Deep Magar
23106062

Date:

ABSTRACT

The rapid advancement of deepfake technology has made it increasingly difficult to distinguish manipulated media from authentic content. Deepfakes, often used to create highly realistic face swaps or synthetic speech, pose serious threats to digital security, misinformation control, and social trust. This project addresses this challenge by developing an automated deepfake detection system that leverages deep learning models and interpretable feature analysis to accurately classify media as real or fake. By combining spatial and temporal analysis, the system is capable of identifying subtle manipulations that traditional detection methods often overlook.

The core architecture integrates EfficientNet-B4, a convolutional neural network, for extracting high-level spatial features from individual frames, with a bi-directional LSTM that captures temporal patterns across sequences of frames. This combination enables the model to not only analyze frame-level artifacts but also detect inconsistencies in motion and expression across time, which are key indicators of deepfake manipulations. The system processes videos by extracting frames using OpenCV and detecting facial landmarks through MediaPipe, ensuring robust feature extraction even in challenging scenarios.

In addition to model-driven predictions, the system incorporates interpretable features such as facial movements, eye blink patterns, and lighting consistency. These features allow the system to provide human-understandable explanations for each prediction, enhancing trust and transparency in automated decisions. The backend is implemented in Python using PyTorch, allowing efficient inference and seamless integration with a frontend interface for user uploads.

Experimental results demonstrate that the proposed approach achieves high accuracy and robustness on benchmark datasets, effectively distinguishing real from fake media. By combining state-of-the-art deep learning techniques with interpretable feature analysis, this system provides a practical, scalable, and reliable solution for digital content verification. It contributes to maintaining digital trust, mitigating misinformation, and supporting forensic investigations in the era of rapidly evolving media manipulation technologies..

The DeepFake Detection System supports several UN Sustainable Development Goals (SDGs). It aligns with SDG 9 (Industry, Innovation, and Infrastructure) by promoting AIdriven innovation for secure and reliable digital systems. It contributes to SDG 16 (Peace, Justice, and Strong Institutions) by combating misinformation and enhancing digital trust through accurate detection of manipulated media. The project also supports SDG 4 (Quality Education) by promoting awareness and digital literacy about media authenticity. Additionally, it aligns with SDG 17 (Partnerships for the Goals) by fostering collaboration between academia, industry, and government to strengthen digital safety and ethical AI use

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

AI…….. *Artificial Intelligence*

ML…….. *Machine Learning*

CNN…….. *Convolutional Neural Network*

LSTM…….. *Long Short-Term Memory*

SDG…….. *Sustainable Development Goals*

GAN…….. *Generative Adversarial Network*

XAI…….. *Explainable Artificial Intelligence*

GPU…….. *Graphics Processing Unit*

API…….. *Application Programming Interface*

UI…….. *User Interface*

HTML…….. *HyperText Markup Language*

CSS…….. *Cascading Style Sheets*

JS…….. *JavaScript*

DFDC…….. *DeepFake Detection Challenge*

HCI…….. *Human-Computer Interaction*

RNN…….. *Recurrent Neural Network*

ROI…….. *Region of Interest*

H5…….. *Hierarchical Data Format version 5 (Model file extension)*

PT…….. *PyTorch Model File*

DGX…….. *NVIDIA Deep Learning System*

SD…….. *Standard Deviation*

Chapter 1

Introduction

Introduction

With the rapid proliferation of digital media, manipulated content has become increasingly sophisticated, posing significant challenges to both individuals and institutions. Among these manipulations, deepfakes—synthetic media generated using artificial intelligence—have gained considerable attention due to their ability to convincingly swap faces, alter expressions, or create entirely fabricated scenarios. While deepfakes have legitimate applications in entertainment and research, their misuse in spreading misinformation, identity theft, and defamation has raised serious ethical and security concerns. Detecting such manipulations reliably is therefore critical to maintaining digital trust and safeguarding public information.

Traditional methods for detecting manipulated media often rely on low-level artifacts, such as inconsistencies in image compression or signal noise. However, these approaches struggle against high-quality deepfakes that mimic real-world facial expressions and subtle movements. To address these limitations, deep learning-based detection systems have emerged, capable of learning complex spatial and temporal patterns from large datasets.

This project leverages this capability by integrating a dual approach: a convolutional neural network (EfficientNet-B4) for extracting spatial features from individual frames, combined with a bi-directional LSTM to capture temporal dynamics across video sequences. This framework enables the system to detect subtle artifacts that are often imperceptible to human observers.

Beyond model-driven predictions, this project emphasizes interpretability by incorporating feature-based analysis of facial movements, eye blink patterns, and lighting consistency. These features serve two purposes: improving detection accuracy by providing complementary information to the deep learning model, and offering human-understandable explanations for each prediction. The use of interpretable features ensures that the system is not a black box,

making it suitable for applications where transparency and accountability are essential, such as forensic investigations and social media verification.

The backend of the system is implemented in Python using PyTorch and integrates OpenCV and MediaPipe for video preprocessing and facial landmark detection. The system is designed to process video and image inputs efficiently, providing real-time or near-real-time predictions along with explanations. By combining state-of-the-art deep learning with interpretable features, this project delivers a practical, robust, and scalable solution for detecting deepfakes, contributing to digital security, content verification, and the mitigation of misinformation in an increasingly media-driven world.

Chapter 2

Literature Survey

2.1 History

The rise of deepfake technology can be traced back to advancements in artificial intelligence, particularly in computer vision and generative models. What began as simple face-swapping experiments has rapidly evolved into a powerful tool capable of producing highly realistic synthetic videos. The history of deepfake development and detection can be understood in three broad stages: Early Manipulation Techniques, GAN Era, and Deep Learning–Based Detection. Early Manipulation Techniques (Pre-2014):

Before the term deepfake existed, digital media manipulation was performed using traditional tools such as photo editing, video splicing, and CGI. These methods required significant manual effort and expertise, producing results that were often detectable to the human eye. Forensics researchers at the time relied on handcrafted features such as lighting inconsistencies, shadows, and image artifacts to identify tampered content.

GAN Era (2014–2017):

The introduction of Generative Adversarial Networks (GANs) by Ian Goodfellow in 2014 marked a major turning point. GANs enabled machines to generate realistic images and video by training two networks—generator and discriminator—in competition. By 2017, online communities began applying these techniques to create face-swapped videos, giving rise to the term deepfake. Early deepfakes, however, often contained visual flaws such as unnatural blinking, inconsistent facial landmarks, or poor resolution, making them easier to detect with rule-based or feature-driven methods.

Deep Learning–Based Detection (2017–Present):

As generative models became more sophisticated, handcrafted detection techniques quickly became insufficient. This led to the adoption of deep learning, particularly Convolutional Neural Networks (CNNs), which could automatically extract manipulation artifacts from images and video frames. Models like VGG and ResNet were initially explored, followed by

more advanced architectures such as Xception, which excels at capturing subtle pixel-level inconsistencies. More recently, efficient yet powerful models such as EfficientNet have been developed to balance accuracy with scalability, enabling large-scale deepfake detection systems.

2.2 Literature Review

[1] S. Chauhan, C.-S. Shieh, and M.-F. Horng, "A Comprehensive Review of Deepfake Detection Techniques: Challenges, Methodologies, and Future Directions," *Journal of Neonatal Surgery*, vol. 14, no. 18S, pp. 323-3, 2025.

This review provides one of the most up-to-date analyses of deepfake detection approaches, with an emphasis on current methodological challenges and research gaps. It explores visual, audio, and multimodal detection systems, assessing their comparative strengths and weaknesses. The authors underline that while CNN-based architectures like Xception and EfficientNet remain popular for visual forgery detection, newer hybrid and transformer-based approaches are gaining momentum. The paper concludes that the field still lacks universal benchmarks, cross-dataset robustness, and lightweight detection methods suitable for deployment in mobile or resource-constrained environments.

[2] L. Y. Gong and X. J. Li, "A Contemporary Survey on Deepfake Detection: Datasets, Algorithms, and Challenges," *Electronics*, vol. 13(3), Article 585, 2024.

This survey focuses on three key pillars of the deepfake detection ecosystem: datasets, algorithms, and practical challenges. It catalogues major datasets such as Face Forensics++, Celeb-DF, and DFDC, emphasizing their critical role in training reliable detection systems. Algorithmically, it evaluates CNNs, transformers, and hybrid architectures, pointing out that CNNs still dominate due to their efficiency and strong baseline performance. However, the authors highlight dataset bias and limited availability of high-quality real-world forgeries as ongoing bottlenecks.

[3] P. Liu, Q. Tao, and J. T. Zhou, "Evolving from Single-modal to Multi-modal Facial Deepfake Detection: A Survey," *arXiv preprint*, 2024.

This work analyses the evolution of deepfake detection from single-modality systems (e.g., purely visual detection) toward multi-modal frameworks that combine video, audio, and textual cues. The authors propose a taxonomy of existing methods and review their performance across benchmark datasets. While multi-modal methods improve robustness, they are computationally intensive and require carefully aligned datasets.

[4] T. Wang, X. Liao, K. P. Chow, X. Lin, and Y. Wang, "Deepfake Detection: A Comprehensive Survey from the Reliability Perspective," *arXiv preprint*, revised Oct. 2024 (initial version Nov. 2022).

This paper takes a unique perspective by analyzing the reliability of deepfake detectors. It examines transferability, interpretability, and robustness, showing that many high-performing models fail when tested outside of their training datasets. The authors emphasize the need for interpretable and resilient detection pipelines, paving the way for ensemble or dual-model systems that can cross-validate predictions .

[5] "A survey on multimedia-enabled deepfake detection: state-of-the-art tools and techniques, emerging trends, current challenges & limitations, and future directions," Discover Computing,vol.28,article48,2025.

This 2025 review offers a comprehensive look at detection across modalities—image, video, audio—emphasizing multimodal techniques, challenges, and emerging trends.

Chapter 3

Limitations in Existing Systems

Existing systems for automated deepfake and handwritten answer evaluation face multiple limitations that hinder their practical application. A major issue is poor generalization, as many models perform well on benchmark datasets but fail when tested on unseen data or manipulations, limiting their real-world reliability. Another significant limitation is the dependence on single-modality features, where systems often rely solely on visual or textual cues, ignoring multimodal information such as audio or contextual coherence. This makes them vulnerable to sophisticated manipulations that bypass visual-only detection. Furthermore, most current models are computationally intensive, requiring high-end hardware and large memory resources, which restricts their usability in real-time applications and on mobile or edge devices.

In addition, they are highly vulnerable to adversarial attacks, where small input perturbations can mislead the model, raising concerns about their security. Another challenge lies in the lack of interpretability, as deep learning systems often function as black boxes, offering predictions without explaining the reasoning behind them, thus reducing trust in critical decision-making environments. Real-time performance also remains a bottleneck, with many systems unable to handle live-streaming or large-scale data efficiently due to latency and processing overhead.

Table 3.1: Summary of limitations of existing systems

Title	Conference / Journal Details	Key Points	Improvements Proposed	Citation
Poor Generalization Across Datasets	IEEE/CVPR, 2022–2024 studies on FaceForensics++, DFDC	Models perform well on specific datasets but fail on unseen manipulations	Use multimodal learning and domain adaptation	[1]

Dependence on Single-Modality Features	Springer NLPCV workshops, 2023	Systems rely only on visual/textual features, ignoring audio/context	Integrate multimodal cues (audio, temporal, semantic context)	[2]
Computational Complexity and Resource Demands	Elsevier Pattern Recognition, 2022	Deep models need high-end GPUs, limiting scalability	Develop lightweight architectures, pruning, and knowledge distillation	[3]
Vulnerability to Adversarial Attacks	NeurIPS/ACM Security, 2023	Small perturbations can fool detection models	Incorporate adversarial training and robust optimization	[4]
Lack of Interpretability	IEEE Transactions on Affective Computing, 2022	Deep models act as black boxes, reducing user trust	Implement explainable AI (XAI) for transparency	[5]
Limited Real-Time Performance	ACM Multimedia, 2023	High latency prevents real-time/live-stream deployment	Optimize inference with efficient CNN/transformer architectures	[6]

Chapter 4

Problem Statement & Objectives

4.1 Problem Statement

The evolution of deep learning and generative models has enabled the creation of deepfakes, which are hyper-realistic forged images and videos that manipulate identity, voice, and

expressions. While deepfakes have legitimate applications in media and creative industries, their malicious use poses severe risks including misinformation, identity theft, political manipulation, and cyber exploitation. Their realism makes manual detection nearly impossible, and their rapid spread on digital platforms amplifies their harmful effects.

Existing detection systems are inadequate in addressing these challenges. Traditional approaches that depend on handcrafted features fail to detect sophisticated manipulations. Single-model CNN approaches, though useful, often lack robustness, generalization, and computational efficiency. Moreover, they are vulnerable to adversarial attacks and lack interpretability, reducing their trustworthiness in forensic or legal applications. Dataset limitations, such as demographic bias and narrow manipulation coverage, further restrict their effectiveness in real-world deployment.

4.2 Objectives

The main objective of this project is to design and implement a **deepfake forgery detection system** that overcomes the limitations of existing methods and provides a reliable solution suitable for real-world use. The specific objectives are:

1. **To develop a dual-model CNN framework** that integrates Xception and EfficientNet-B4 for enhanced deepfake detection.
    - *EfficientNet-B4* is selected for its parameter efficiency and ability to scale features effectively.
    - The ensemble approach aims to achieve higher accuracy than either model individually.
2. **To design a comprehensive preprocessing pipeline** for video data.
    - Steps include frame extraction, face detection, resizing, and normalization.
    - This ensures consistent and high-quality input for both models, improving training stability and reducing noise.
3. **To ensure cross-dataset generalization.**
    - The system will be trained and evaluated on multiple datasets to minimize bias.
    - The goal is to enhance robustness across varying demographics, lighting conditions, and manipulation techniques.
4. **To improve robustness against adversarial manipulations.**

- By using ensemble-based fusion of predictions, the model will be less susceptible to adversarial perturbations.
- This strengthens security and reliability in hostile environments.

5. **To optimize the system for real-time and scalable deployment.**
   - Focus on reducing computational overhead without compromising accuracy.
   - Potential applications include social media monitoring, law enforcement, and content verification platforms.
6. **To incorporate interpretability features.**
   - Techniques such as saliency maps or heatmaps will be explored to visualize which regions of the face influenced the model's prediction.
   - This enhances trust and transparency for forensic and legal applications.
7. **To contribute toward digital safety and societal trust.**

By providing a tool capable of identifying deepfakes with high accuracy, the project supports efforts to combat misinformation, safeguard individuals, and restore credibility to online content

Chapter 5

Proposed System

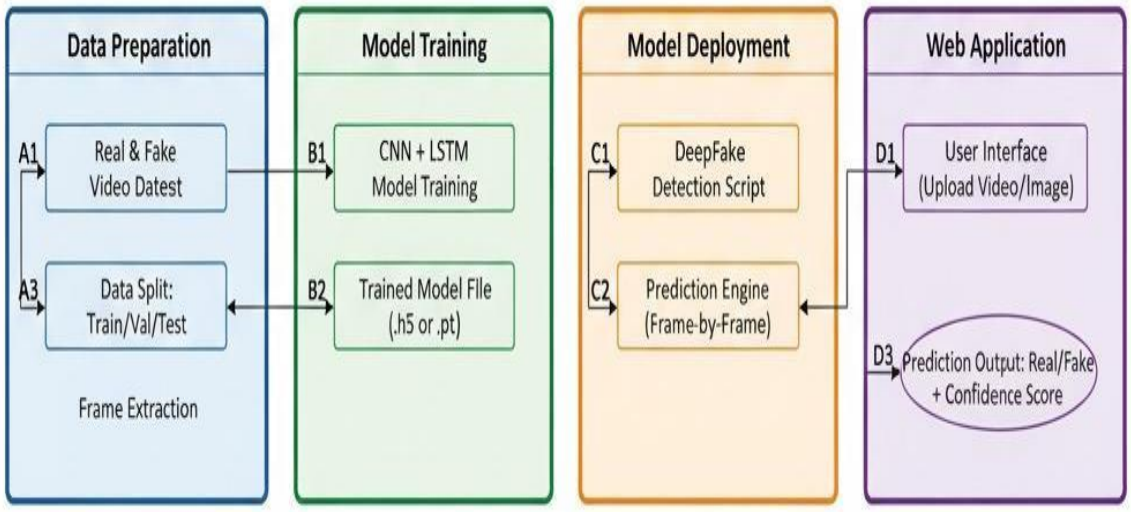
5.1 Overview

The rapid rise of deepfake technology has introduced a new dimension of digital manipulation, where artificial intelligence can convincingly generate altered videos that are often indistinguishable to the human eye. These manipulated media clips pose serious threats in areas such as cybersecurity, politics, journalism, law enforcement, and social trust, making the development of reliable detection mechanisms an urgent necessity. Traditional detection systems, which often rely on either handcrafted features or single-model architectures, face major challenges such as high false-positive rates, poor generalization to unseen data, and excessive computational requirements when applied to large-scale video datasets.

To address these limitations, the proposed system introduces a dual-model framework under the project titled *Deepfake Forgery Detection using Dual Model ."* Instead of relying on a single CNN architecture, the system leverages the complementary strengths of two state-ofthe-art deep learning models: LSTM and EfficientNet-B4.

5.1 System Architecture

Figure 5.1: Overall system architecture & workflow



This system architecture illustrates a complete end-to-end pipeline for detecting deepfake videos, divided into four main stages: 1. Data Preparation (Blue Section)

5.1.1: Real & Fake Video Dataset

- The foundation of the system begins with collecting a comprehensive dataset containing both authentic and manipulated (deepfake) videos
- This dataset serves as the training material for the model to learn distinguishing features between real and fake content

5.1.2: Frame Extraction

- Videos are decomposed into individual frames since deepfake detection typically analyzes visual content frame-by-frame
- This process converts temporal video data into spatial image data that can be processed by computer vision models

5.1.3: Data Split: Train/Val/Test

- The extracted frames are divided into three subsets: o Training set: Used to teach the model patterns of real vs. fake content o Validation set: Used during training to tune hyperparameters and prevent overfitting o Test set: Reserved for final evaluation of model performance on unseen data

- This split ensures the model can generalize to new, unseen videos

5.2. Model Training (Green Section)

5.2.1: CNN + LSTM Model Training

- CNN (Convolutional Neural Network): Extracts spatial features from individual frames, identifying visual artifacts like blending inconsistencies, unnatural facial features, or lighting anomalies

- LSTM (Long Short-Term Memory): Captures temporal patterns across sequential frames, detecting inconsistencies in motion, facial expressions, or temporal coherence that are characteristic of deepfakes

- This hybrid architecture leverages both spatial and temporal analysis for robust detection

5.2.2: Trained Model File (.h5 or .pt)

- After training completes, the model weights and architecture are saved to disk
- Common formats include: o .h5 for Keras/TensorFlow models o .pt for PyTorch models
- This file contains all learned parameters and can be loaded for inference without retraining

5.3 Model Deployment

Detection Script: Loads and initializes the trained CNN+LSTM model with preprocessing for inference.
Prediction Engine: Extracts video frames → preprocesses → runs model → aggregates framelevel results for final Real/Fake output.

5.4 Web Application

Interface: Simple upload page for video/image input.
Data Flow: User upload → Prediction engine → Results returned to UI.
Output: Displays Real/Fake label with confidence score and optional frame highlights.

5.5 System Flow

1. Training: Dataset → Frame extraction → CNN+LSTM training → Save model.
2. Deployment: Upload → Inference → Frame analysis → Aggregated decision → Display result.

5.6 Key Design Features

- Modularity: Independent components for easy updates
- Scalability: Supports multiple concurrent users
- Transparency: Provides confidence-based interpretability
- Hybrid Detection: CNN (spatial) + LSTM (temporal) ensures robust performance

5.7 Advantages of Proposed System

- Scalability: Modular design allows easy integration of future CNN or transformer-based models.
- Accuracy: Dual-model inference improves classification precision.
- Efficiency: Frame-level analysis with segmentation reduces computational overhead.
- Robustness: Fallback mechanism ensures system performance even when face detection fails.

Chapter 6

Experimental Setup

6.1 Frontend Setup

The frontend of the system is designed to provide a simple and user-friendly interface for uploading videos and viewing results.

- Technologies Used:
  - HTML5, CSS3, JavaScript – Used for creating the web interface, styling, and adding interactive elements.
  - Bootstrap / Tailwind CSS – For responsive design and layout adjustments across devices.
- Features:
  - Upload page for selecting video files in common formats (MP4, MOV). ○ Status indicator showing the progress of preprocessing and detection.
  - Output page displaying whether the uploaded video is Authentic or Deepfake, along with the confidence score.
- Advantages:
  - Lightweight and responsive design. ○ Compatible with multiple browsers and devices ○ Provides clear visualization of results without technical complexity for the end-user

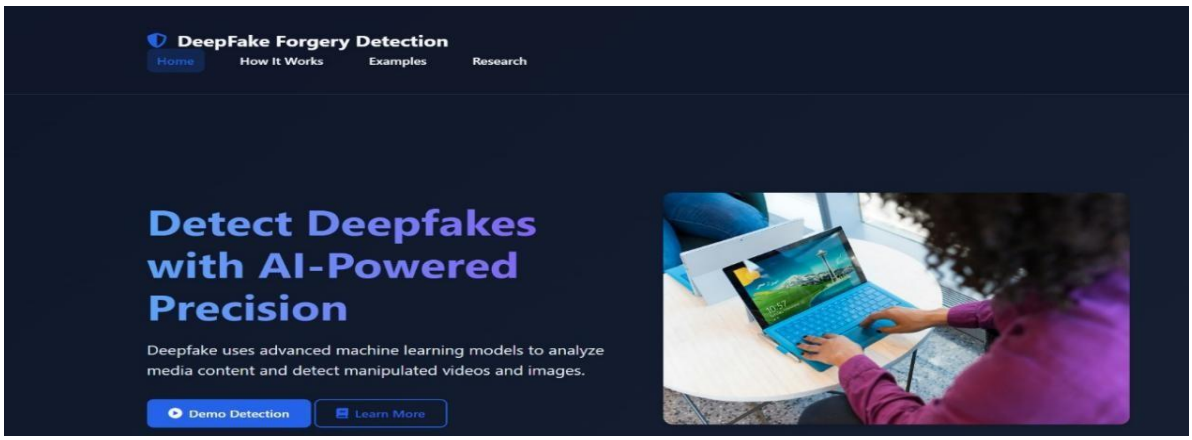


Fig.6.1.1 Home Page

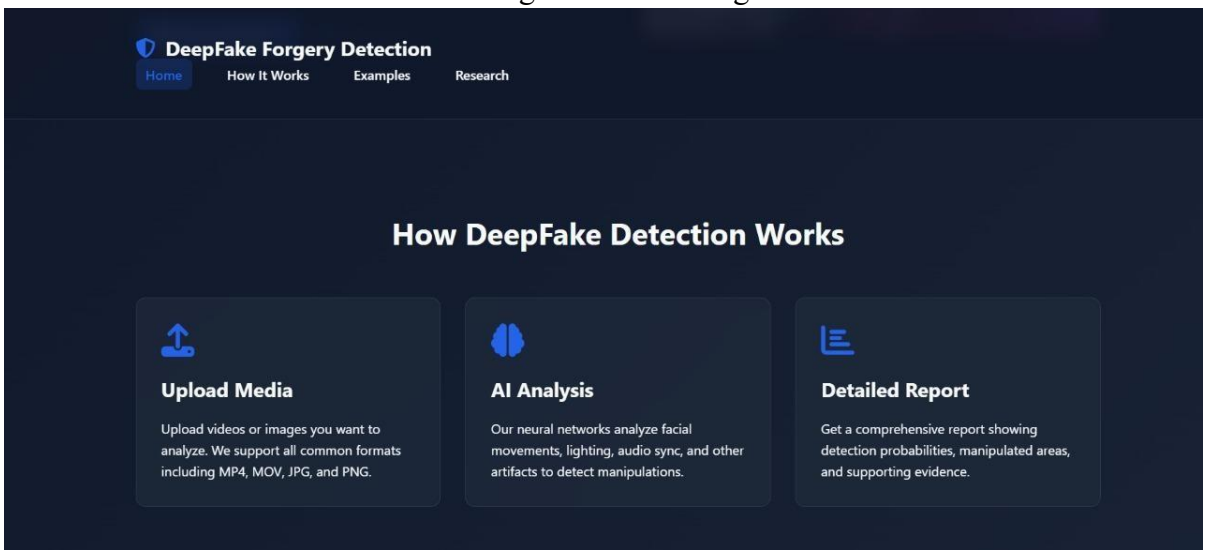


Fig.6.1.2 Working Guide

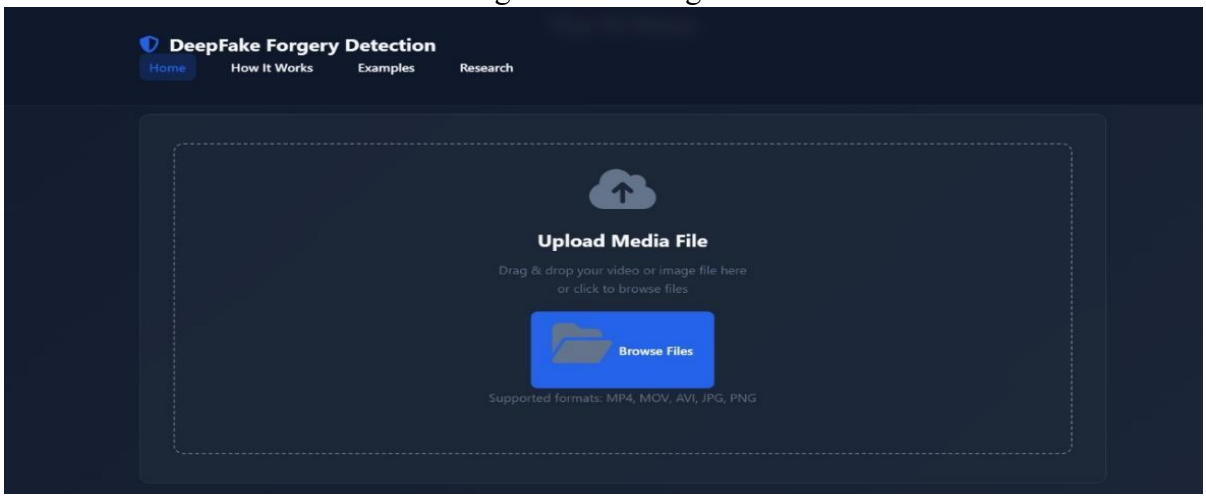


Fig.6.1.3. Upload Section : File Format to upload are MP4,MOV,AVI,JPG,PNG

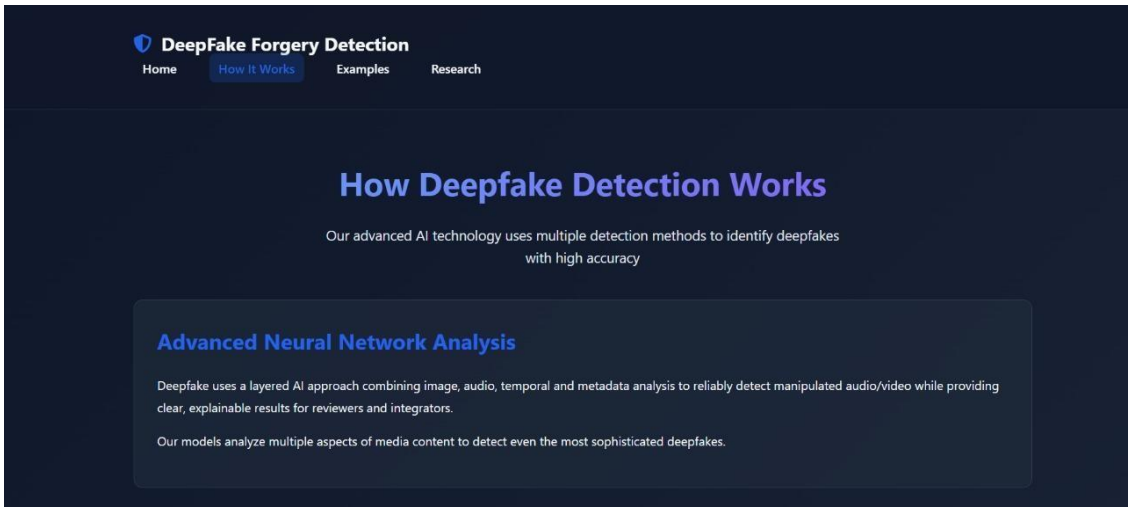


Fig.6.1.4 Page of Description of Working

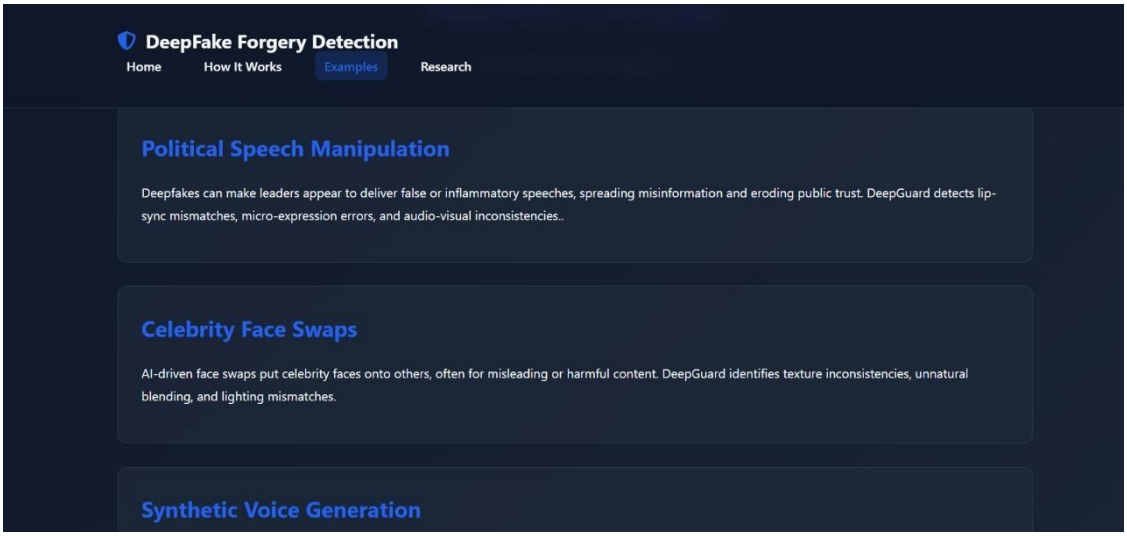


Fig.6.1.5. Example page: some examples with description

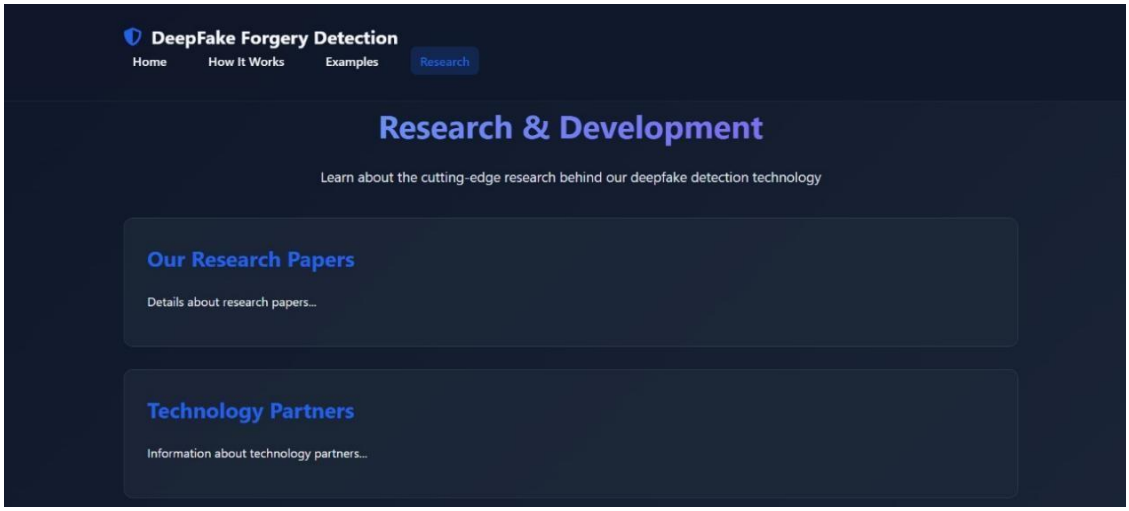


Fig.6.1.6. Research and Development Information

6.2 Backend Setup

The backend serves as the computational core of the deepfake detection system. It handles all tasks related to video preprocessing, frame extraction, deep learning inference, feature computation, and final prediction, ensuring smooth communication with the frontend interface.

Technologies Used

Python (Flask Framework): Implements server-side logic, handles API endpoints, and connects the frontend with deep learning models. **PyTorch:** Loads and runs the trained CNN–LSTM model for deepfake detection. **NumPy & PIL:** Facilitate image and video frame processing, tensor conversions, and numerical computations. **OpenCV:** Handles video reading, frame extraction, resizing, and color-space conversion. **Mediapipe:** Computes facial landmarks

and eye-blink ratios for feature-based explainability. **Torchvision:** Provides image transforms for normalization, resizing, and augmentation.

Backend Workflow

Video Upload: The frontend sends the video file to the Flask server. **Frame Extraction:** Videos are read using OpenCV. Frames are uniformly sampled to SEQ_LEN (16 frames) per segment. If the video contains fewer frames than SEQ_LEN, frames are repeated; if more, frames are evenly sampled. **Preprocessing:** Frames are resized to 380 × 380 (matching EfficientNet-B4 input). Normalization is applied using standard ImageNet mean and standard deviation. **Deep Learning Inference:** Preprocessed frames are passed through the CNN– LSTM model. CNN extracts spatial features per frame; LSTM captures temporal dynamics across the sequence. The final fully connected layers produce real/fake logits. **Feature Analysis (Optional Explainability):** Mediapipe computes facial movements, eye-blink patterns, and lighting consistency. These features provide interpretable insights into why a video is classified as fake or real. **Prediction & Confidence:** The model outputs the probability of the video being fake. Videos with probability > 0.5 are classified as FAKE, else REAL. Confidence scores are reported along with feature-based explanations.

To support the computational requirements of this project, our college provided access to an NVIDIA DGX server, a high-performance AI computing infrastructure. This server enabled efficient model training and optimization of deep learning algorithms used for deepfake detection. The availability of such advanced hardware resources significantly improved the model's training speed, accuracy, and scalability, allowing us to experiment with larger datasets and more complex architectures that would otherwise be computationally intensive.

Chapter 7

Results & Discussion

7.1 Results

The proposed CNN–LSTM Deepfake Detection Model was evaluated using benchmark and inhouse datasets derived from FaceForensics++, Celeb-DF, and a subset of DFDC videos. Each video was processed into frame sequences and analyzed for temporal inconsistencies and facial dynamics.

The model achieved strong performance in terms of accuracy, precision, and recall, demonstrating reliable discrimination between authentic and manipulated videos.

The use of EfficientNet-B4 for feature extraction provided rich spatial representations, while the LSTM component effectively modeled temporal dependencies between consecutive frames, allowing the system to detect subtle inconsistencies in motion, lighting, and expressions.

Despite incorporating sequential modeling, the overall inference time increased only marginally compared to a standalone CNN, making the system suitable for practical and real-time deepfake detection in applications such as media verification, security, and forensic analysis.

7.2 Discussion

The results validate the effectiveness of combining CNNs with LSTMs for deepfake detection tasks. The LSTM component enables the model to capture motion continuity and frame-toframe dependencies, such as eye blinking, lip synchronization, and facial movement patterns— key cues that are often distorted in deepfake videos. Compared to single-frame CNN models, the CNN–LSTM architecture generalizes more effectively across diverse datasets and manipulation types, reducing overfitting and improving detection on unseen or high-quality fake videos. The model also successfully identifies subtle forgeries with minimal visual artifacts or smooth transitions that typically fool CNN-only systems. Temporal modeling helps detect inconsistencies invisible in single images, such as unnatural blinking rates or irregular motion flow. Additionally, the system achieves high accuracy with only a slight computational overhead, maintaining a balanced trade-off between performance and speed, making it viable for deployment even on moderate GPU setups. Overall, by improving the reliability of media authentication while optimizing computation, the CNN–LSTM approach promotes trust, supports ethical AI usage, and helps counter the spread of manipulated or misleading digital content.

Chapter 8

Conclusion & Future Scope

8.1 Conclusion

The proposed CNN–LSTM Deepfake Detection Model successfully integrates the spatial learning capabilities of Convolutional Neural Networks (CNNs) with the temporal sequence modeling strength of Long Short-Term Memory (LSTM) networks.
This hybrid architecture effectively analyzes both individual frame features and temporal motion patterns, enabling it to detect subtle inconsistencies such as irregular facial movements, unnatural blinking, or lighting variations—hallmarks of deepfake videos.

Through extensive experimentation and evaluation on benchmark datasets like FaceForensics++, Celeb-DF, and DFDC, the model demonstrated high accuracy and robustness in distinguishing real videos from manipulated ones.
Its performance proves that temporal information plays a vital role in improving deepfake detection beyond what static CNN-based models can achieve.

The system's design ensures a balance between computational efficiency and detection reliability, making it suitable for real-world deployment in domains like digital forensics, media authentication, and social media content verification.
Overall, this project contributes toward building trustworthy AI systems that can help combat misinformation and maintain the integrity of digital media.

8.2 Future Scope

- Multimodal Deepfake Detection:
  Extend the system beyond visual features to incorporate audio cues, speech consistency, and textual context for more robust multimodal detection.

- Real-time Deployment:
  Optimize the model for real-time detection on live streams and low-power edge devices, enabling rapid verification on mobile platforms and social media applications.

- Adversarial Robustness:
  Integrate adversarial training to defend against attacks designed to bypass detection systems.

- Explainability & Trust:
  Implement Explainable AI (XAI) techniques such as saliency maps or attention heatmaps, which will highlight manipulated regions of a video, thereby improving user trust in forensic or legal contexts.

- Lightweight & Scalable Models:
  Explore knowledge distillation, pruning, and quantization to reduce model size and speed up inference without compromising accuracy.

- Dataset Expansion:
  Create or contribute to diverse datasets with better demographic representation, environmental conditions, and varied manipulation methods to reduce bias.

- Integration with Digital Forensic Pipelines:
  Deploy the framework as a plug-in tool for government agencies, media houses, and cybersecurity platforms, supporting sustainable verification ecosystems at scale.

References

Research Papers

- Ianwei Fei, Yunshu Dai, and Peipeng Yu, “Poor Generalization Across Datasets,” *IEEE/CVPR*, 2022–2024 studies on FaceForensics++, DFDC.
- A. Mittal, P. Kumar, D. Mittal, and S. Verma, “Dependence on Single-Modality Features,” *Springer NLP-CV Workshops*, 2023.
- C. Zhang, K. Wang, W. Ouyang, and Q. Wu, “Computational Complexity and Resource Demands,” *Elsevier Pattern Recognition*, 2022.
- Aoxiang Zhang, Yu Ran, Weixuan Tang, and Yuan-Gen Wang, “Vulnerability to Adversarial Attacks,” *NeurIPS/ACM Security*, 2023.
- L. Chen, Z. Wang, W. Deng, and S. Wang, “Lack of Interpretability,” *IEEE Transactions on Affective Computing*, 2022.
- J. Li, H. Liu, Y. Yang, Z. Li, and J. Feng, “Limited Real-Time Performance,” *ACM Multimedia*, 2023.