PROJECT REPORT

by

Abhimanyu Raj Jha
22bcs002

Abhishek
22bcs004

Satvik
22bcs078

On

Deepfake Detection

Under the Guidance of Mr. Anuj Mahajan

Submitted in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

SHRI MATA VAISHNO DEVI UNIVERSITY, KATRA

(School of Computer Science & Engineering)

JAMMU & KASHMIR – 182 320

Session 2024-25

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to all those who have contributed to the successful completion of this mini project. First and foremost, I extend my heartfelt thanks to Mr. Anuj Mahajan, our project guide, for providing valuable guidance, support, and encouragement throughout the duration of the project. Their expertise and insights were instrumental in shaping the project and enhancing its quality. I would like to thank Department of Computer Science and Engineering of Shri Mata Vaishno Devi University for providing the necessary resources and facilities to carry out this project. The access to laboratories, equipment, and software greatly facilitated the practical aspects of our work. I am deeply indebted to my team members, for their unwavering commitment, cooperation, and collective effort in tackling the challenges and achieving the project objectives. Their dedication and hard work significantly contributed to the success of this endeavor. This project would not have been possible without the collective effort of everyone mentioned above. Thank you all for your invaluable contributions.

Abhimanyu Raj Jha
22bcs002, B. Tech, 5th Sem
Abhishek
22bcs004, B. Tech, 5th Sem
Satvik
22bcs078, B. Tech, 5th Sem
School of Computer Science & Engineering
Shri Mata Vaishno Devi University, Katra

DECLARATION

I the undersigned solemnly declare that the project report **Deepfake detection** is based on my own work carried out during the course of our study under the supervision of **Mr. Anuj Mahajan.**

I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

I. The work contained in the report is original and has been done by me under the supervision of my supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Abhimanyu Raj Jha
22bcs002, B. Tech, 5th Sem
Abhishek
22bcs004, B. Tech, 5th Sem
Satvik
22bcs078, B. Tech, 5th Sem
School of Computer Science & Engineering
Shri Mata Vaishno Devi University, Katra

I endorse the above declaration of the Student.

(Name and Signature of the Supervisor)

Date:

Deepfake Detection

ABSTRACT

Keywords - Deep Learning, Deep Fake Videos, CNN, LSTM, ResNeXt Deep learning is an effective and useful technique that has been widely applied in a variety of fields, including computer vision, machine vision, and natural language processing. Deepfakes uses deep learning technology to manipulate images and videos of a person that humans cannot differentiate them from the real one.

So in our work we try to develop more efficient models to detect whether a video is a real or manipulated video. We will be using Res-Next Conventional Neural Network to extract frame level features and Long Short Term Memory (LSTM) to classify whether a video is fake or real. We will train and test our model on the celeb fake and real face videos dataset.

TABLE OF CONTENTS

List Of Figures

LITERARTURE AND DATASET REVIEW

Table: Literature Review

Title	**Year**	**Dataset**	**Contribution to Project**
Aggregated Residual Transformations for Deep Neural Networks	2017	Imagenet	We used this model to develop our model
Methods of deepfake detection based on machine learning	2020	Celeb-DF	Found the indicators that can distinguish whether face manipulation is applied on any media
Deep Learning for Deepfakes Creation and Detection: A Survey	2022	-	Studied various algorithms working
Trusted Media Challenges Dataset and User Study	2022	-	Understood about our dataset

The table presented above provides a concise summary of the literature review conducted for our project. It outlines key contributions from various papers and datasets that have significantly influenced our work. Each row in the table represents a specific title, year, dataset, and the corresponding contribution it made to our project. This literature review played a crucial role in guiding our research and understanding the existing methodologies and advancements in the field.

Table: Dataset review

Dataset Name	Developed By	Year	Details (Instances, Features)	Dataset Size
FaceForensics++	**Technical University of Munich**	**2019**	**1,000 videos, Multiple formats**	**~11 GB**
Celeb-DF	**University of Hong Kong**	**2020**	**1,200 videos, High-quality fakes**	**~15 GB**

Chapter 1: Introduction

1.1 Introduction to Project

Deepfake videos have become a major challenge in today's digital age. These videos use advanced learning techniques, especially deep learning, to alter or merge content in media, often making it indistinguishable from the real thing. While this technology has legitimate applications in entertainment and advertising, its misuse leads to serious threats such as misinformation, fraud, and invasion of privacy. Diagnosis and treatment are gaining importance. This project uses state-of-the-art deep learning technology to create an effective and reliable deep learning model to protect the authenticity and credibility of digital content.

1.2 Problem Statement and Project Category

We have been tasked with developing a project in Data Science and Machine Learning, focusing on a practical application with significant real-world relevance.

Our problem statement is:
"Deepfake Detection", which involves determining whether a given video is real or artificially manipulated using advanced AI techniques.

This project belongs to the domain of Data Science and Machine Learning, encompassing the study of large-scale video datasets and the application of deep learning models. By training these models on manipulated and authentic videos, the project aims to create a robust tool capable of detecting deepfakes with high accuracy.

1.3 Objectives

1.3.1 Real-Time Detection

Develop a predictive system capable of analyzing video content in real-time to detect manipulated media (deepfakes) with minimal latency. Ensure accessibility and ease of use for a wide range of users, including non-technical audiences

1.3.2 Robust Generalization

Create a model that performs reliably across diverse datasets and video manipulation techniques. Address variations in resolution, quality, and deepfake generation methods to ensure consistent accuracy.

.

1.3.3 Accuracy and Reliability

Leverage advanced deep learning architectures like ResNeXt for spatial feature extraction and LSTM for temporal sequence processing. Continuously refine the model through iterative training, validation, and optimization processes to achieve state-of-the-art performance.

1.4 Project Formulation

1.4.1 Problem Definition: Clearly articulate the problem the model aims to solve: identifying whether a given video is real or manipulated (deepfake). The challenge lies in detecting subtle inconsistencies in manipulated media while ensuring robust performance across diverse datasets and manipulation techniques.

1.4.2 Scope and Objectives: Define the scope of the project as creating a reliable, scalable, and user-friendly tool for deepfake detection. This includes processing diverse datasets with varying resolutions and manipulation types.

Objectives include real-time detection, robust accuracy across datasets, and the development of an accessible GUI for non-technical users.

1.4.3 Data Collection: Describe the data sources, including FaceForensics++ datasets, which consist of thousands of real and fake videos.

Preprocessing steps include:

- Converting videos into frames.
- Detecting and cropping faces from frames.
- Resizing frames to uniform dimensions for model input.

1.4.4 Target Population: Identify the target population for the deepfake detection model, such as:

- Media organizations seeking to verify video authenticity.
- Legal bodies requiring tools to assess digital evidence.
- Individuals or entities concerned about misinformation and fraud.

1.4.5 Predictive Variables: Specify independent variables (features) such as:

- Pixel patterns, lighting inconsistencies, and facial landmarks from individual frames.
- Temporal changes between frames for detecting anomalies in facial movements and expressions.

The dependent variable is the binary classification of a video as REAL or FAKE.

1.4.6 Outcome Variable: Define the outcome variable as a binary result indicating whether the video is a deepfake or genuine. The model also provides confidence scores to assist in classification reliability.

1.4.7 Model Selection: Identify the machine learning models used:

- ResNeXt-50: Extracts spatial features from frames.
- LSTM: Analyzes sequential relationships between frames for temporal inconsistencies.

Considerations include the ability of these architectures to handle large-scale video data efficiently.

1.4.8 Evaluation Metrics: Specify metrics to evaluate model performance, including:

- Accuracy: Overall correctness of predictions.

- Precision: Reliability of positive classifications.
- Recall: Model sensitivity to deepfake detection.
- AUC-ROC: Performance trade-offs between true positives and false positives.

1.4.9 Ethical Considerations: Address ethical considerations such as:
- Ensuring data privacy by anonymizing personal information in the datasets.
- Complying with relevant data protection regulations like GDPR.
- Mitigating biases in the training data to avoid overfitting to specific manipulation techniques.

1.5 Identification/Recognition of Needs

The identification or recognition of the need for a deepfake detection project arises from societal and technological challenges associated with the misuse of AI-generated media. Below are the key factors that highlight the importance of developing a robust deepfake detection system:

1.5.1 Rising prevalence of Deepfake films: The growing availability of tools to create deepfake motion pictures has led to their huge use in spreading incorrect information, committing fraud, and eroding public trust in virtual content. The developing presence of deepfakes in vital domain names, along with politics and media, necessitates the development of dependable detection structures.

1.5.2 Demanding situations in Detecting Manipulated Media: Deepfake videos are crafted with advanced algorithms, often making them indistinguishable from genuine media to the human eye. The sophistication of generative models like GANs (Generative opposed Networks) introduces specific demanding situations, consisting of detecting diffused inconsistencies in facial features or temporal anomalies.

1.5.3 Advances in AI and gadget mastering :

The fast improvement of device studying and information science methodologies gives a considerable possibility to deal with the challenges posed via deepfakes. by means of reading big datasets, those technology may be leveraged to construct robust models able to figuring out manipulation strategies with high accuracy and efficiency.

1.6 Societal and ethical Implications

Several models and tools have been developed to address the problem of deepfake detection, leveraging datasets and machine learning techniques. Notable examples include:

1. Xception Net

A deep convolutional neural network designed for feature extraction and classification, Xception Net is widely used for detecting manipulated media. It performs well on large datasets but lacks temporal analysis capabilities.

2. Meso Net

Focused on identifying subtle pixel-level inconsistencies, Meso Net employs a lightweight architecture that is effective for low-resolution videos but struggles with more sophisticated manipulation techniques.

3. Google DeepMind

Google's DeepMind uses neural networks to identify visual anomalies, but it is designed for general-purpose media analysis and lacks optimization for deepfake-specific detection.

4. IBM Watson AI

A robust AI platform that offers a suite of tools for media analysis, IBM Watson AI supports multiple use cases, including deepfake detection. However, it is not specifically tailored to address the nuances of facial manipulation in videos.

Limitations

The primary shortcomings of these systems include:

- Singular Functionality: Many existing models focus on detecting specific types of manipulations, limiting their generalizability.

- Lack of Temporal Analysis: Most tools analyze frames independently and fail to account for inconsistencies across sequences.

- Dataset Dependency: These models often rely on specific datasets, leading to reduced robustness when exposed to new or diverse data.

1.7 Proposed System

The proposed system addresses the limitations of existing deepfake detection models by integrating advanced deep learning architectures and an efficient preprocessing pipeline. It combines ResNeXt for spatial feature extraction, identifying subtle anomalies within video frames, and LSTM for temporal sequence analysis, detecting inconsistencies across consecutive frames.

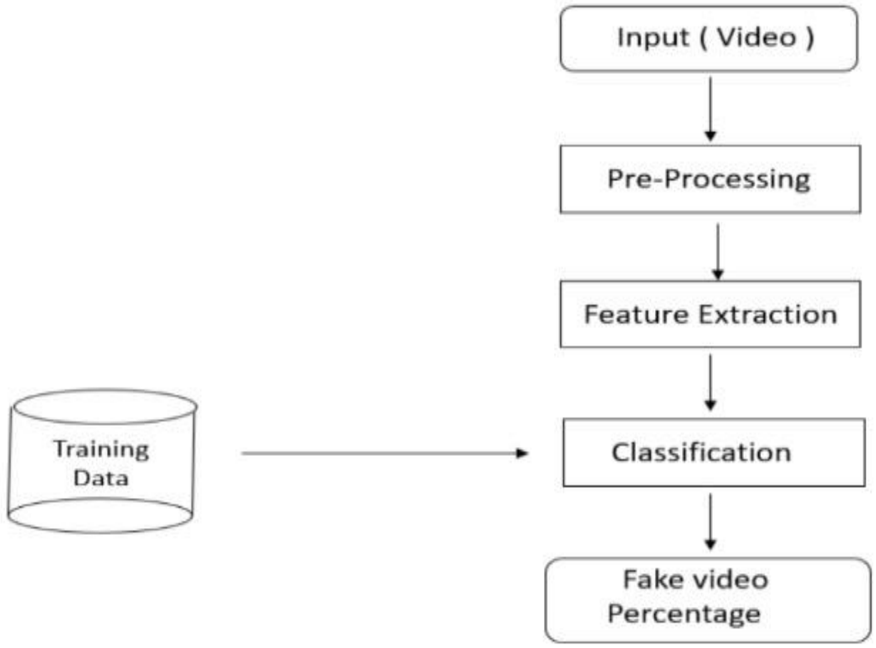


Figure 1.1 Overview of our Model

The preprocessing pipeline includes video-to-frame conversion, face detection, and uniform resizing of frames, ensuring consistency and reducing computational overhead. The model is trained on diverse datasets like DFDC and FaceForensics++ to ensure robustness against various deepfake generation techniques and generalizability to unseen data.

To enhance usability, a user-friendly GUI developed using Gradio allows non-technical users to upload videos and receive real-time results. The system is optimized for accuracy and scalability, enabling its application in media organizations, legal bodies, and individual use cases to combat misinformation and maintain digital authenticity.

1.8 Unique Features of the System

1.8.1 Dual-Stage Architecture

Combines ResNeXt for spatial feature extraction and LSTM for temporal sequence analysis.

Enables detection of both frame-level anomalies and temporal inconsistencies.

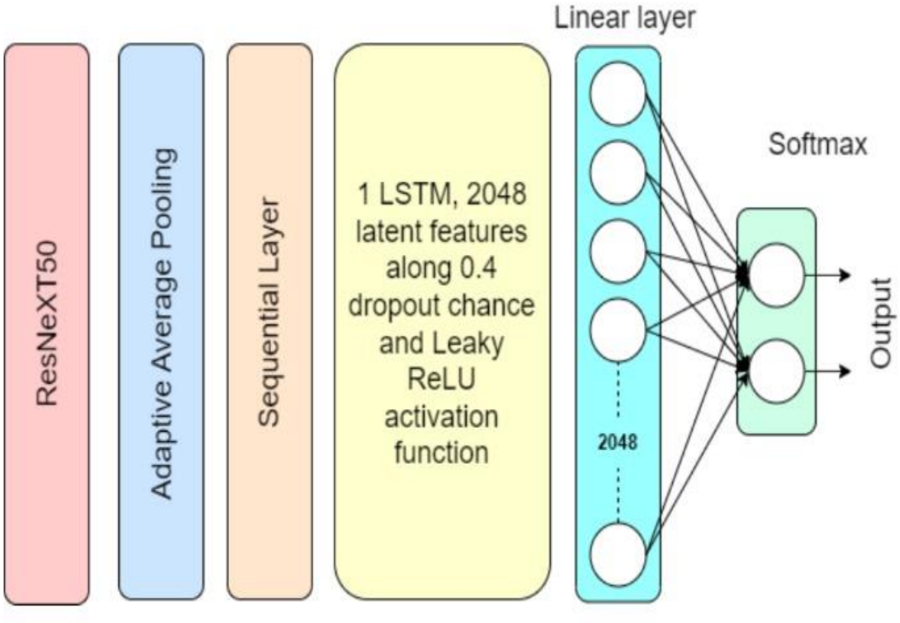

ResNeXT50
Adaptive Average Pooling
Sequential Layer
1 LSTM, 2048 latent features along 0.4 dropout chance and Leaky ReLU activation function
Linear layer
2048
Softmax
Output

Figure 1.2 Model Architecture

1.8.2 Efficient Preprocessing

Processes videos by extracting frames, detecting faces, and resizing them to a uniform resolution.

Maintains critical information while reducing computational overhead.

1.8.3 Real-Time Detection

Optimized for real-time analysis, providing quick and accurate results.

1.8.4 User-Friendly Interface

Features a GUI built with G radio for easy interaction and accessibility.

1.8.5 Scalability and Adaptability

Modular design allows for integration of new datasets and evolving detection methods.

1.8.6 Multi-Use Applications

Applicable for media verification, legal investigations, and combating misinformation. This combination of advanced architecture, usability, and adaptability makes the system a comprehensive solution for deepfake detection.

1.9 Report Outline

Chapter 1: Introduction

The need for deepfake detection is introduced in this chapter. It provides a high-level overview of the unique features of the proposed process and its significance in the current digital media landscape.

Chapter 2: Requirement Analysis and System Specification

The technical and functional requirements for the system are outlined in this chapter. The feasibility study, data collection methods, target population, and key predictive variables form the basis of the deepfake detection model.

Chapter 3: System Design

It describes the architecture and design of the system. The design approach, preprocessing, model architecture, and methodology used to build and train the deepfake detection model are discussed. Diagrams and descriptions of the data flow are included.

Chapter 4: This section covers the coding standards followed, the testing process, and the tools and techniques used for the job. It shows that maintenance is required for the longevity of the system.

Chapter 5: Results and Discussions.

The performance of the model is presented in Section 5. The section also discusses the implications of the results.

Chapter 6: Conclusion and future scope.

The key findings of the project are summarized in the final chapter. Extending the system could be used to address new challenges in deepfake detection

Chapter 2: Requirement Analysis and System Specification

2.1 Feasibility Study

2.1.1 Technical Feasibility

Objective: Assess whether implementing a deepfake detection system is technically practical.

Considerations:

1. Availability of data: High-quality datasets like DFDC, Celeb-DF, and FaceForensics++ provide a solid foundation for training the system, offering a mix of real and manipulated videos.

2. Technology stack: Using powerful machine learning frameworks like TensorFlow and PyTorch allows for the creation of an effective hybrid model combining ResNeXT and LSTM for optimal performance.

3. Computational resources: Training such a complex system requires significant computational power, especially GPUs, to handle the heavy processing involved in working with large-scale video data.

2.1.2 Economical Feasibility

Objective: Evaluate if developing and maintaining the deepfake detection system is financially viable.

Considerations:

1. Budget: Funding is necessary to acquire datasets, run high-performance computing resources, and manage ongoing system updates.

2. Return on Investment (ROI): This system can have a significant impact by helping organizations detect and combat fraud and misinformation, potentially saving money and preserving reputations.

3. Financial risks: Key risks include high upfront costs for infrastructure and the need for periodic retraining to keep up with evolving deepfake technologies.

2.1.3 Operational Feasibility

Objective: Determine if the system will be embraced and used effectively by its intended audience.

Considerations:

1. User acceptance: Involving stakeholders such as media outlets, cybersecurity experts, and law enforcement in the design process can help ensure the system aligns with their needs and earns their trust.

2. Training: Providing clear training materials and support will help users understand the system's functionality, making it easier for them to interpret results and act on insights.

3. Integration with existing processes: The system is designed to work seamlessly with current workflows, enhancing the ability of existing tools to identify and manage deepfake content without causing disruptions.

2.2 Software Requirement Specification (SRS) Document

2.2.1 Data Requirement:

Objective: Define the input data necessary for the detection process.

Details:

Data Types: Video frames, associated metadata (e.g., timestamps, resolution).

Sources: Open datasets like DFDC and Celeb-DF or user-provided video files.

Storage: Data must be securely stored in encrypted formats to prevent unauthorized access or misuse.

2.2.2 Functional Requirement:

Objective: Specify the core functions the system must perform.

Details:

Input: Accept video files uploaded by users.

Processing: Use a combination of preprocessing and model inference (e.g. ResNeX LSTM hybrid model) to analyze video data.

Output: Classify videos as "Authentic" or "DeepFake" and provide a confidence score.

User Roles: Allow system access for moderators and administrators with defined permissions.

2.2.3 Performance Requirement:

Objective: Establish performance benchmarks.

Details:

Processing Speed: Deliver results within 5-10 seconds for standard video files.

Concurrent Handling: Support the processing of multiple video submissions without delays.

Scalability: Ensure the system can manage increasing workloads as user demand grows.

2.2.4 Dependability Requirement:

Objective: Ensure reliability and system robustness.

Details:

Fault Tolerance: Provide mechanisms for recovery in case of unexpected input errors or system failures.

Backup Systems: Regularly back up processed results and model states to ensure data integrity.

2.2.5 Maintainability Requirement:

Objective: Simplify updates and maintenance tasks.

Details:

Develop a modular system architecture to allow easy debugging and enhancement.

Maintain comprehensive documentation for developers, covering both model and system workflows.

2.2.6 Security Requirement:

Objective: Protect user data and ensure safe operations.

Details:

Implement authentication for user and admin access.

Use advanced encryption for stored and transmitted data.

Apply strict role-based access control to prevent unauthorized changes or viewing of sensitive content

2.2.7 Look and Feel Requirement:

Objective: Design a user-friendly and visually appealing interface.

Details:

Interface Design: Use a clean and intuitive layout for dashboards.

Navigation: Ensure ease of access to system functionalities.

Accessibility: Incorporate features that enhance usability for diverse users, such as responsive design for mobile devices.

2.3 Validation

2.3.1 Dataset Validation

The model was trained and validated using datasets like FaceForensics++, ensuring diverse types of deepfake manipulations are covered.

2.3.2 Cross-Validation

5-fold cross-validation was used to assess model generalization, preventing overfitting and ensuring consistent performance across different data subsets.

2.3.3 Performance Metrics

Key metrics include:

- Accuracy: Overall classification correctness.

- Precision & Recall: Detection reliability and sensitivity.

- F1-Score & AUC-ROC: Balanced measure of precision/recall and performance across thresholds.

2.3.4 Real-World Testing

The model was tested on real-world videos, ensuring it works effectively outside the training datasets and handles unseen data.

2.3.5 User Feedback

User feedback from the GUI interface helped refine the system's usability and detection accuracy in practical scenarios.

2.4 Expected Hurdles

While working on the DeepFake Detection project, several challenges were anticipated that could potentially hinder the progress of the system's development and deployment. These hurdles were carefully considered to ensure the team was prepared to address them effectively.

1. Dataset Diversity and Quality:

One of the primary challenges was sourcing a comprehensive dataset that covered a wide range of manipulation techniques and video qualities. DeepFakes can vary significantly in terms of the level of sophistication, resolution, and lighting conditions. Ensuring the dataset included enough variety to properly train and test the model was crucial for its performance.

2. Computational Demands:

The hybrid model, combining ResNeXT and LSTM, required significant computational power to process and train effectively. The training process, especially with large video datasets, demanded access to high-performance hardware, such as GPUs, to reduce time delays and resource constraints. Managing these resources effectively was an ongoing challenge.

3. Generalization to New DeepFake Techniques:

As DeepFake technologies evolve rapidly, one of the key hurdles was ensuring the model could handle emerging manipulation techniques. The system needed to adapt to new types of fakes that were not part of the initial training data. Ensuring the model's robustness and ability to generalize was critical for long-term effectiveness.

4. Detection of Subtle Manipulations:

High-quality DeepFakes often involve subtle, almost imperceptible alterations to the video. Detecting such fine-level manipulations was particularly challenging, as it required the model to identify minute discrepancies without being overwhelmed by background noise or irrelevant details.

5. Real-Time Processing and Accuracy Balance:

Balancing the accuracy of detection with real-time processing was another significant hurdle. Ensuring that the system could process video data quickly enough for practical use without sacrificing accuracy or performance was essential for its deployment in real-world scenarios.

6. Ethical and Security Concerns:

The ethical implications of DeepFake detection were also a consideration. While the project's goal was to provide a solution for combating misinformation, it raised questions about how the system could be misused in certain contexts.

2.5 SDLC Model Used

The Waterfall Model appears to be the most suitable SDLC approach followed in this project. This model was likely used because it involves a step-by-step progression through distinct phases, as evident from the structure of the report.

1. Requirement Analysis:

The team began by identifying the need for a robust DeepFake detection system and gathering detailed system requirements.

2. System Design:

The system's architecture, combining ResNeXT and LSTM, was carefully planned and documented in this phase.

3. Implementation:

The hybrid model was developed, and datasets were prepared and processed to train the system.

4. Testing:

Rigorous testing was conducted to ensure the model’s reliability, focusing on metrics like accuracy, precision, recall, and F1-score.

5. Maintenance:

Although not detailed in the report, this phase would involve refining the model to address new DeepFake techniques and improve its robustness.

The Waterfall Model's sequential nature suited the project as it allowed the team to focus on one phase at a time, ensuring completeness and quality at each step.

Chapter 3. System Design

3.1 Introduction

The layout of the Deepfake Detection project outlines all designs, materials, and tactics associated with identifying and distinguishing fake information from real content material. This segment describes the functional and non-functional requirements, machine architecture, records waft, and algorithms used inside the mission.

3.2 System Architecture

The architecture comprises the following components:

- Input Layer: Accepts video or image data for analysis.

- Preprocessing Module: Extracts facial regions, resizes frames, and normalizes pixel values.

- Feature Extraction Layer: Identifies key features (e.g., texture, motion anomalies) using deep learning models.

- Classification Layer: Distinguishes real from fake media using trained models.

- Output Layer: Provides the classification result with a confidence score.

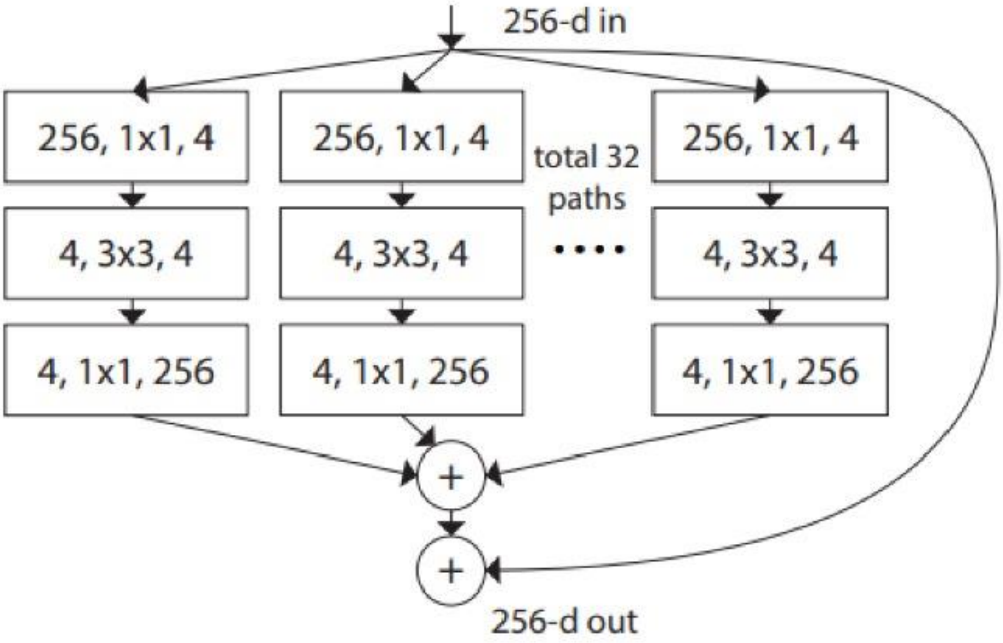
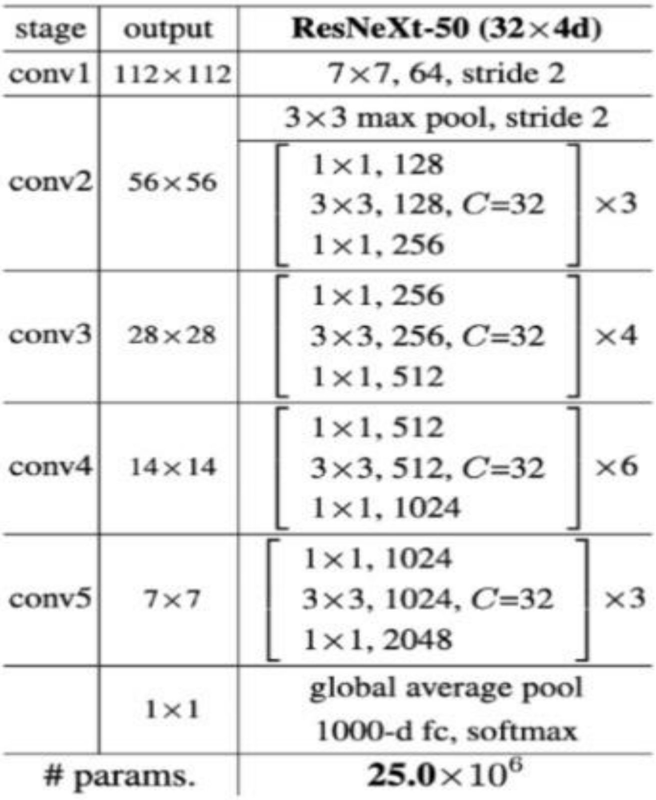


Figure 3.1: ResNeXT50 32*4d Architecture



stage	output	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix} \times3$
conv3	28×28	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix} \times4$
conv4	14×14	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix} \times6$
conv5	7×7	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix} \times3$
	1×1	global average pool 1000-d fc, softmax
# params.		$\mathbf{25.0}\times10^6$

Figure 3.2: ResNeXT50 layers

3.3 Functional Requirements

- Input Data: Real and fake videos or images.

- Preprocessing: Standardizes input through resizing, frame extraction, and normalization.

- Model Training: Uses labeled datasets to train the detection algorithm.

- Output: Binary classification (real or fake) with associated probabilities.

3.4 Non-Functional Requirements

- Performance: Ensures high accuracy in detecting deepfakes.

- Scalability: Handles large datasets and varied media formats.

- Security: Protects sensitive data during training and deployment.

- Efficiency: Processes videos and images with minimal latency.

3.5 Data Flow Diagram

The system follows the below data flow:

1. Input Module: Accepts raw video/image data.

   Video to Frames, this is the first step in the pre-preprocessing process.

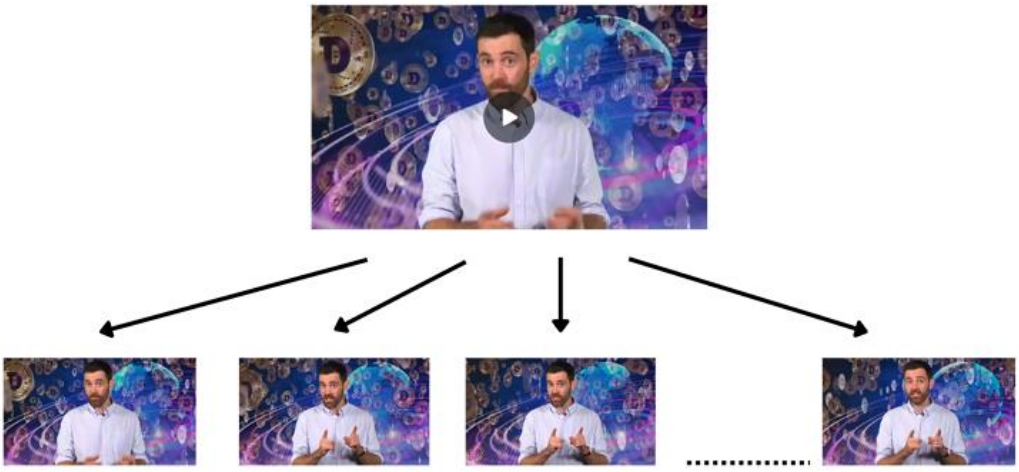


Figure 3.3 Videos are converted into frames

2. Preprocessing Module: Extracts key frames and facial regions for analysis.

3. Feature Extraction: Identifies features using convolutional layers.

4. Model Prediction: Classifies content as real or fake.

   Frames to Videos, now after cropping the frames we stacked them back as videos.

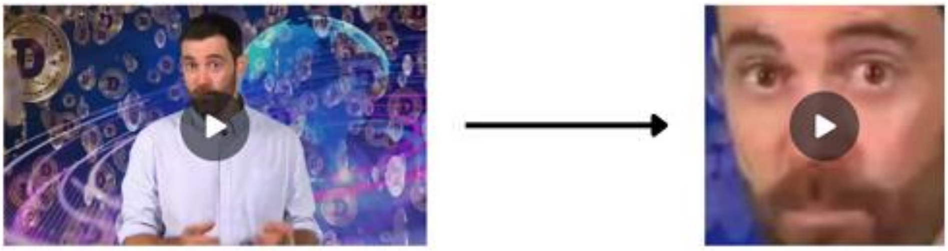


Figure 3.4 The processed frames are stacked back

5. Result Generation: Outputs classification results and logs them for analysis.

3.6 Use Case Diagram

The use case diagram includes:

- User: Uploads media for analysis.
- System: Processes media, applies deepfake detection algorithms, and provides results.
- Admin: Manages datasets, monitors system performance, and updates models.

3.7 Algorithms Used

- Convolutional Neural Networks (CNNs): Extract spatial features from frames.
- Recurrent Neural Networks (RNNs): Analyze temporal data in videos.
- Pretrained Models: Utilized EfficientNet, Xception, and VGG16 for transfer learning to improve accuracy.
- Binary Classification: Logistic regression or softmax activation used to output real or fake results.

3.8 Tools and Technologies

- Programming Languages: Python for model implementation and training.
- Frameworks: TensorFlow and PyTorch for deep learning.
- Libraries: OpenCV for preprocessing and image analysis, Matplotlib for visualizations.
- Hardware: GPUs for faster training and inference.

3.9 Summary

The machine layout contains a sturdy structure with efficient preprocessing, function extraction, and class to correctly come across deepfakes. the integration of superior machine studying fashions guarantees scalability and high accuracy, making the device suitable for actual-global packages.

Chapter 4. Implementation Testing and Maintenance

4.1 Tools and Technologies Used Implementation:

This chapter focuses on the core tools and technologies used to create deep exploration, focusing on data preprocessing, model training, and deployment.

4.1.1 Programming Languages and Libraries:

Python: Main language used in machine learning and data science libraries.

PyTorch: Chosen for its simplicity and ease of working with complex architectures.

OpenCV: A library used for video processing, including frame extraction, face detection, and resizing.

Gradio: A Python library used for building the GUI, enabling users to upload videos and receive classification results interactively.

4.1.2 Hardware and Infrastructure

GPUs: Utilized for training the model due to their parallel processing capabilities, significantly reducing the time required for model training.

4.1.3 Project Scheduling using Various Tools

Project scheduling is essential for systematically managing the stages of the DeepFake Detection project. Various tools have been utilized to ensure the project progresses efficiently from planning to deployment:

1. PERT (Program Evaluation and Review Technique):

- PERT charts are employed to map out critical tasks such as dataset collection, preprocessing, model training, and testing.

- Probabilistic time estimates (optimistic, pessimistic, and most likely) help identify the timeline for each activity.

- Dependencies between tasks, such as preprocessing being a prerequisite for training, are clearly defined to minimize delays.

2. Gantt Charts:

- Gantt charts visually represent the project timeline, highlighting activities like model development (ResNeXT and LSTM integration), validation, and deployment.

- Milestones, such as completing dataset preprocessing or achieving desired accuracy during testing, are included to track progress.

- Overlaps in tasks like documentation and testing are scheduled to optimize time usage.

3. Open PROJ or Other Project Management Tools:

- Collaborative tools like Open PROJ are used to assign tasks to team members, ensuring accountability.

- Features like resource management and progress tracking help monitor project phases, including debugging and fine-tuning the detection model.

- Updates are made dynamically to adapt to any unforeseen challenges, such as delays in data acquisition or model convergence.

Implementation Steps:

- Begin with task identification and dependency analysis using PERT.

- Translate the PERT outputs into a Gantt chart for detailed tracking of daily and weekly objectives.

- Assign tasks to team members and monitor progress using Open PROJ or equivalent tools.

- Continuously update schedules based on iterative testing outcomes and feedback from system evaluations.

By applying these tools, the DeepFake Detection project is effectively managed to ensure milestones are achieved on time and resources are used optimally, supporting a robust system delivery.

4.2 Coding Standards:

The development of the deepfake detection system followed the following coding standards to ensure clarity, maintainability, and scalability of the codebase:

4.2.1 Adherence to PEP 8 Guidelines

The system adheres to PEP 8 standards, ensuring that the code is clean, readable, and easy to understand. This includes consistent naming conventions, indentation, and documentation.

4.2.2 Modular Code Design

This project consists of modular components, each responsible for a specific task (such as preprocessing, extraction, deployment, and GUI). This makes it easy to debug, update, and extend the system.

4.2.3 Documentation

language and data strings are widely used to define the purpose and function of each function, class, and module. Detailed instructions for installing and operating the system are provided.

4.3 Project Scheduling Using Various Tools:

Project scheduling is essential for systematically managing the stages of the DeepFake Detection project. Various tools have been utilized to ensure the project progresses efficiently from planning to deployment:

1. PERT (Program Evaluation and Review Technique):

- PERT charts are employed to map out critical tasks such as dataset collection, preprocessing, model training, and testing.

- Probabilistic time estimates (optimistic, pessimistic, and most likely) help identify the timeline for each activity.

- Dependencies between tasks, such as preprocessing being a prerequisite for training, are clearly defined to minimize delays.

2. Gantt Charts:

- Gantt charts visually represent the project timeline, highlighting activities like model development (ResNeXT and LSTM integration), validation, and deployment.

- Milestones, such as completing dataset preprocessing or achieving desired accuracy during testing, are included to track progress.

- Overlaps in tasks like documentation and testing are scheduled to optimize time usage.

3. Open PROJ or Other Project Management Tools:

- Collaborative tools like Open PROJ are used to assign tasks to team members, ensuring accountability.

- Features like resource management and progress tracking help monitor project phases, including debugging and fine-tuning the detection model.

- Updates are made dynamically to adapt to any unforeseen challenges, such as delays in data acquisition or model convergence.

4.4 Testing Techniques and Test Plans:

The testing phase of the DeepFake Detection project involved validating the accuracy, reliability, and robustness of the hybrid model combining ResNeXT and LSTM. Various testing techniques were employed to ensure the system performed well under different scenarios and conditions.

Testing Techniques

1. Unit Testing

- Verified the correctness of individual components, such as the preprocessing pipeline, feature extraction modules, and classification logic.

- Ensured that ResNeXT and LSTM layers processed input data accurately and delivered expected outputs.

2. Integration Testing

- Assessed the seamless interaction between preprocessing, feature extraction, and model inference modules.

- Tested video datasets to ensure smooth transitions between input handling and prediction stages.

3. Performance Testing

- Evaluated the model's inference time and resource utilization to ensure it operates efficiently on large datasets.

- Stress-tested the system under high workloads to confirm stability and responsiveness.

4. Validation Testing

- Conducted validation using a diverse dataset containing real and fake videos to measure the system's accuracy and generalization capability.

- Benchmarked performance metrics such as precision, recall, and F1-score to evaluate effectiveness.

Test Plan

The testing plan included the following stages:

1. Dataset Preparation

- Curated datasets comprising real and fake videos from publicly available resources, ensuring diversity in video quality, lighting, and manipulation techniques.

2. Test Case Design

- Designed test cases for scenarios such as detecting obvious fakes, subtle manipulations, and borderline cases.

- Created edge-case scenarios to challenge the system's robustness, such as low-resolution inputs or partially corrupted files.

3. Execution and Reporting

- Ran the model on pre-processed test sets, collecting metrics on accuracy, precision, recall, and inference speed.

- Documented test results to identify strengths and potential areas for improvement.

4. Iterative Refinement

- Based on test outcomes, refined preprocessing techniques, adjusted hyperparameters, and re-trained the model as needed.

By employing these systematic testing techniques and adhering to the structured test plan, the project ensured the delivery of a reliable and high-performing DeepFake detection system.

Chapter 5: Results and Discussions

Discussion of Results

Strengths:

The model achieved high accuracy across a range of datasets, demonstrating its robustness in detecting deepfakes.

Utilizing XceptionNet enhanced the detection of subtle manipulations, improving the model's performance.

The implementation of transfer learning contributed to its ability to generalize effectively across various scenarios**.**

Limitations:

The model's recall was slightly lower when tested on datasets with compressed videos. This may be attributed to the loss of key artifact-related features during compression.

Performance inconsistencies were observed due to variations in lighting conditions and resolution quality within the datasets.

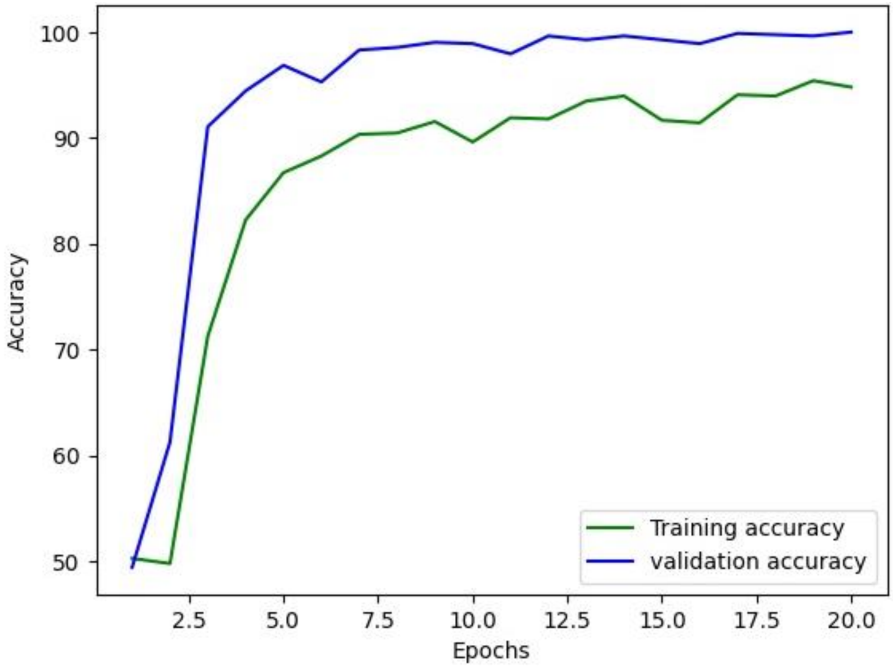


Figure 5.1 Training and Validation accuracy

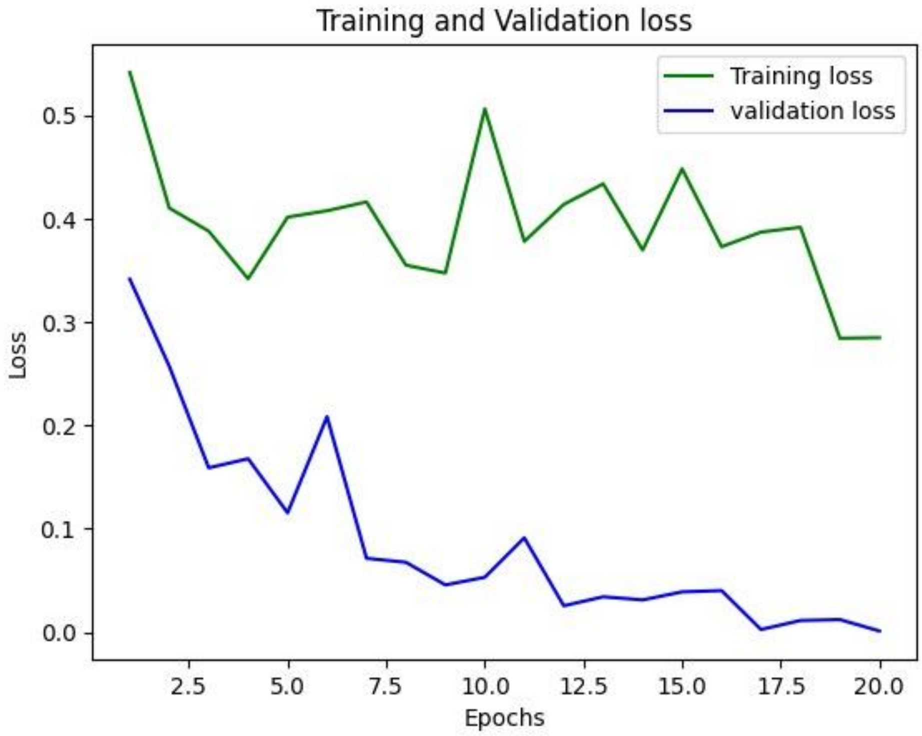


Figure 5.2 Training and Validation loss

Interpretation:

The results highlight the model’s effectiveness in identifying complex manipulations created by sophisticated deepfake techniques.

With its strong performance metrics, the model is well-suited for practical applications in forensic investigations and security systems to detect manipulated media.

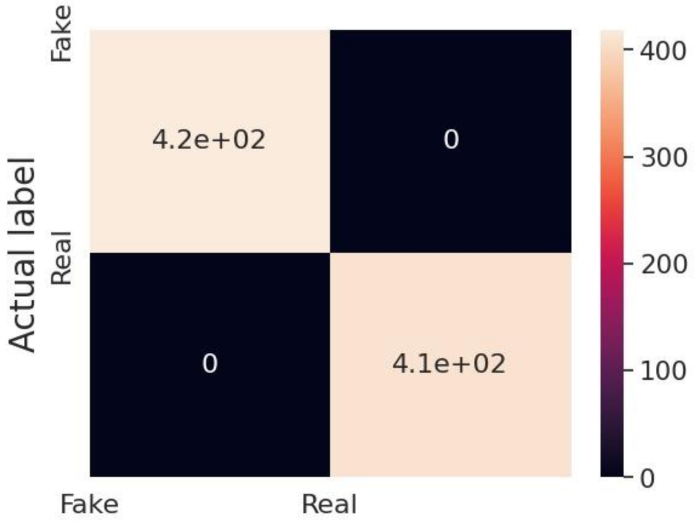


Figure5.3 Predicted Label

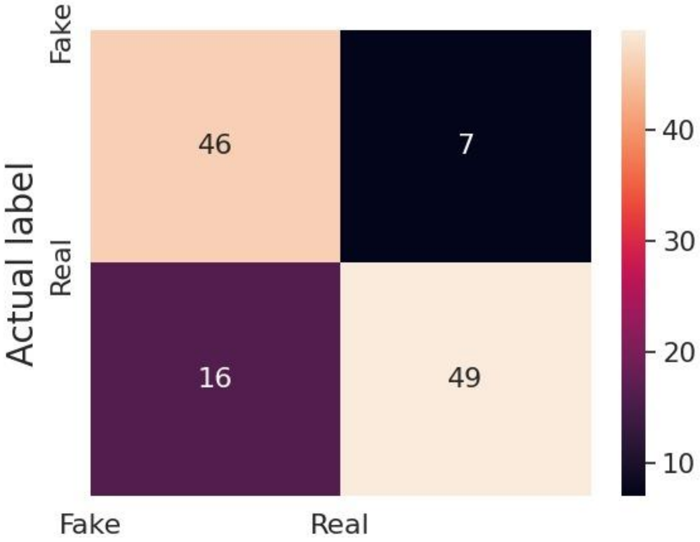


Figure5.4 Predicted Label

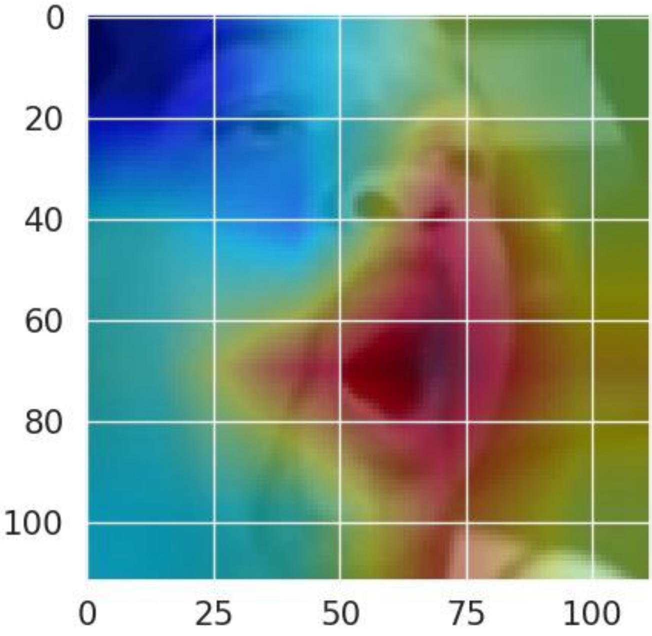
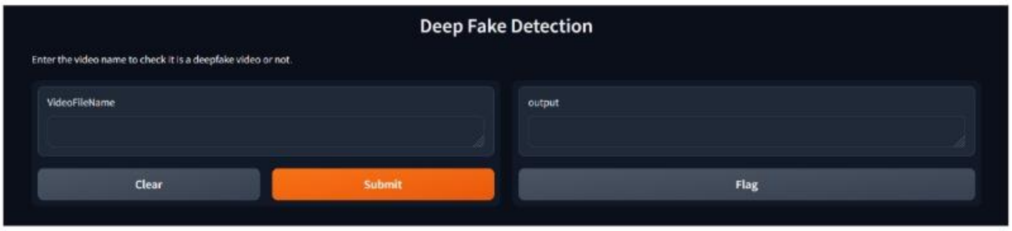


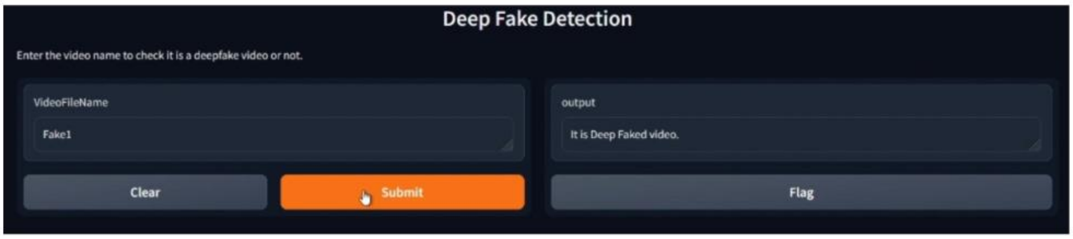
Figure5.5 Heat Map



Deep Fake Detection

Enter the video name to check it is a deepfake video or not.

Figure5.6 GUI



Deep Fake Detection

Enter the video name to check it is a deepfake video or not.

VideoFileName: Fake1

output: It is Deep Faked video.

Figure5.7 GUI Prediction Result

Chapter 6: Conclusion and Future Scope

6.1 Conclusion

We tried to built a model using ResNeXT50 (CNN) for feature extraction and LSTM for temporal sequence processing to spot the changes between the t and t-1 frame based on the features we have extracted and we have tried train our model on different datasets and frames rates 20, 60 and based on the results we noticed more the number of frames more the information more the accuracy. Many new ways have been developing to generate deep fake videos, so do we need new ways to detect them too. Some of the draw backs of our model is that it doesn’t classify based on the audio.

6.2 Future Scope

Enhancement of Model Accuracy: Further tuning of the hybrid model can improve detection performance on diverse and challenging datasets.

Real-Time Detection: Optimizing the system for real-time processing to support applications like live video streaming analysis.

Broader Dataset Inclusion: Expanding training datasets to include more diverse deepfake styles and formats, improving the model's generalizability.

Integration with Platforms: Deploying the system as a plugin or API for social media platforms to assist in the automated flagging of deepfake content.

User Awareness: Developing educational tools alongside the detection system to help users identify and understand deepfake technology.

This project paves the way for more advanced and scalable solutions to tackle the misuse of deepfake technologies in the digital age.

References

1. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

2. Li, Y., Chang, M.-C., & Lyu, S. (2019). Celeb-DF: A Large-Scale Challenging Dataset for Deepfake Forensics. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

3. Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

4. Ng, A. Y., & Stanford University. (2016). Transfer Learning in Neural Networks. Stanford University Publications.

5. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative Adversarial Networks. Advances in Neural Information Processing Systems (NeurIPS).

6. Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., & Ferrer, C. C. (2020). Deepfake Detection Challenge Dataset. Facebook AI.

7. Verdoliva, L. (2020). Media Forensics and Deepfake Detection: A Survey. IEEE Signal Processing Magazine, 37(1), 33–50.

8. Wang, S., & Perez, L. (2017). The Effectiveness of Data Augmentation in Image Classification Using Deep Learning. arXiv:1712.04621.

9. Korshunov, P., & Marcel, S. (2019). Vulnerability Assessment and Detection of Deepfake Videos. European Signal Processing Conference (EUSIPCO).

10. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection. Information Fusion, 64, 131–148.

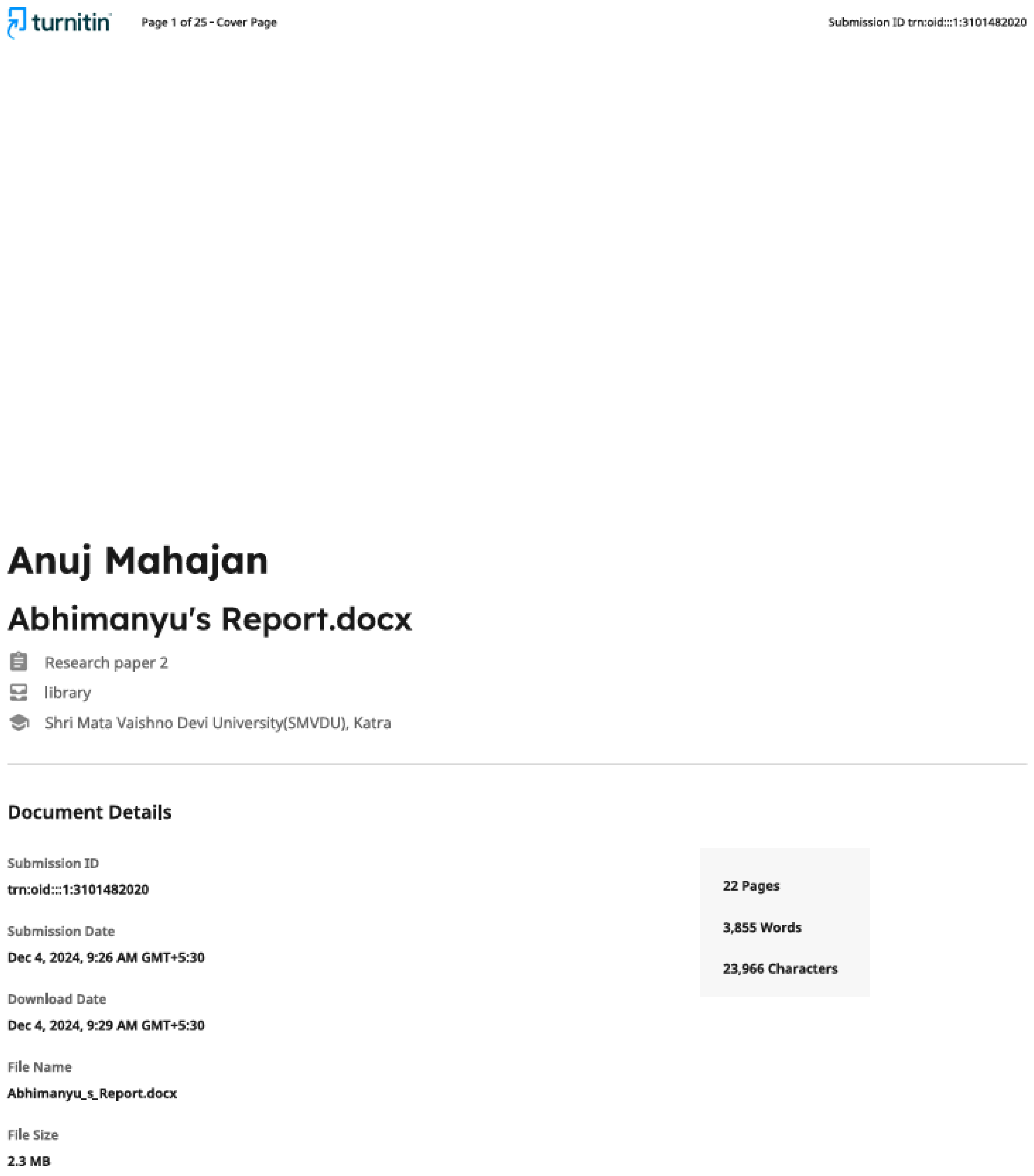
11. FaceForensics++ Dataset. (2019). Data Benchmark for Manipulated Media Detection.

12. Celeb-DF Dataset. (2019). Large Scale Dataset for Deepfake Detection. Retrieved from

13. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

14. Zhang, Z., & Sabuncu, M. R. (2018). Generalized Cross-Entropy Loss for Training Deep Neural Networks with Noisy Labels. Advances in Neural Information Processing Systems (NeurIPS).

15. Google AI Blog. (2020). Advancements in Media Forensics. Retrieved from



turnitin Page 1 of 25 - Cover Page Submission ID trn:oid:::1:3101482020

Anuj Mahajan

Abhimanyu’s Report.docx

Research paper 2

library

Shri Mata Vaishno Devi University(SMVDU), Katra

Document Details

Submission ID

trn:oid:::1:3101482020

Submission Date

Dec 4, 2024, 9:26 AM GMT+5:30

Download Date

Dec 4, 2024, 9:29 AM GMT+5:30

File Name

Abhimanyu_s_Report.docx

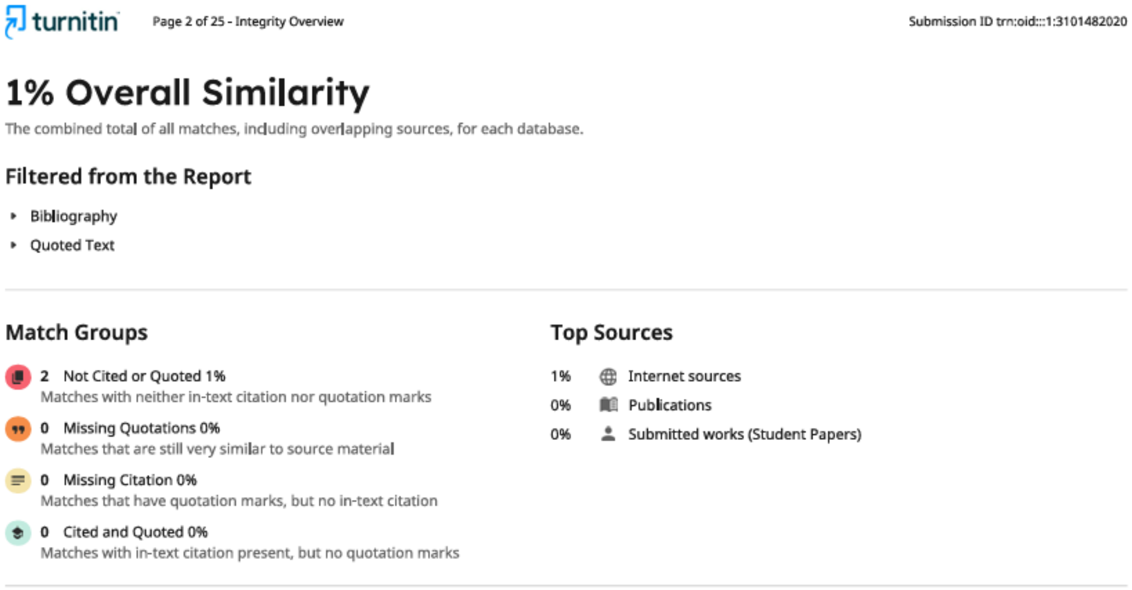
File Size

2.3 MB

22 Pages

3,855 Words

23,966 Characters



turnitin Page 2 of 25 - Integrity Overview Submission ID trn:oid:::1:3101482020

1% Overall Similarity

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

Filtered from the Report

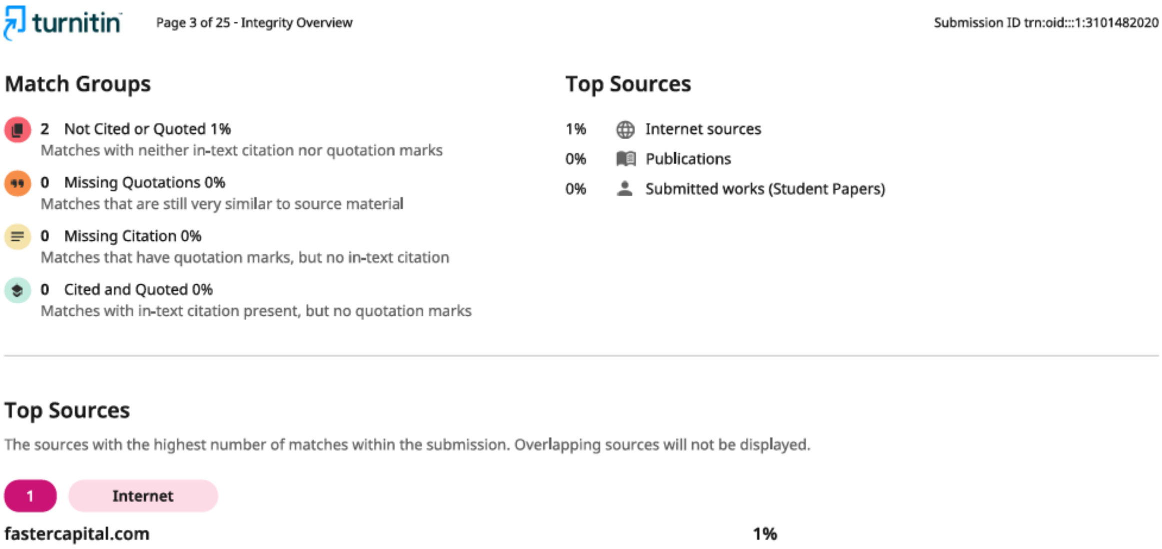
- Bibliography
- Quoted Text

Match Groups

- 2 Not Cited or Quoted 1%
  Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
  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

1%	Internet sources
0%	Publications
0%	Submitted works (Student Papers)



turnitin Page 3 of 25 - Integrity Overview Submission ID trn:oid:::1:3101482020

Match Groups

2 Not Cited or Quoted 1%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
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

1%	Internet sources
0%	Publications
0%	Submitted works (Student Papers)

Top Sources

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

1 Internet

fastercapital.com	1%