A PROJECT REPORT ON

DEEPFAKE DETECTION IMAGES AND VIDEOS USING LSTM AND RESNEXT CNN

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

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Under the guidance of

MR. xxxxxxxxxxxxxxxx, M.E., M.B.A., (Ph. D)

Assistant Professor, Computer Science and Engineering

BY

xxxxxxxxx	2xxxxxxxxx52x
xxxxxxxxx	2xxxxxxxxx52x
xxxxxxxxx	2xxxxxxxxx52x
xxxxxxxxx	2xxxxxxxxx52x

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES, CHITTOOR-517127, A.P.
(Autonomous)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu),
Chittoor, A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

(2024 -25)

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu)
Murukambattu Post, Chittoor-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled **“DEEPFAKE DETECTION IMAGES AND VIDEOS USING LSTM AND RESNEXT CNN”** is the Bonafide work carried out by

xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**

under our supervision and guidance in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **“ COMPUTER SCIENCE AND ENGINEERING ”** during the period 2024-25.

Signature of the Supervisor
Mr.xxxxxxxxxxxxxx. M.E, M.B.A.,(Ph.D)
Assistant Professor,
Department of Computer Science and Engineering,
Sreenivasa Institute of Technology and Management Studies, Chittoor, A.P.

Signature of the head of the Department
Dr.xxxxxxxxxxxxxxxx,M.Tech,Ph.D.,
Associate Professor & HOD,
Department of Computer Science and Engineering,
Sreenivasa Institute of Technology and Management Studies, Chittoor, A.P.

Submitted for University Examination (Viva-Voce) held on…………………………………….…….

Internal Examiner

External Examiner

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES.
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor,A.P-517127

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute Vision

To emerge as a Centre of Excellence for Learning and Research in the domains of engineering, computing and management.

Institute Mission

M1: Provide congenial academic ambience with state-art of resources for learning and research.

M2: Ignite the students to acquire self-reliance in the latest technologies.

M3: Unleash and encourage the innate potential and creativity of students.

M4: Inculcate confidence to face and experience new challenges.

M5: Foster enterprising spirit among students.

Department Vision

To contribute for the society through excellence in Computer Science and Engineering with a deep passion for wisdom, culture and values.

Department Mission

- Provide congenial academic ambience with necessary infrastructure and learning resources.
- Inculcate confidence to face and experience new challenges from industry and society.
- Ignite the students to acquire self-reliance in the latest technologies.
- Foster Enterprising spirit among students.

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES.
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor,A.P-517127

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAMME EDUCATIONAL OBJECTIVES (PEO's):

PEO1: Excel in Computer Science and Engineering program through quality studies, enabling success in computing industry. **(Professional Competency)**.

PEO2: Surpass in one's career by critical thinking towards successful services and growth of the organization, or as an entrepreneur or in higher studies. **(Successful Career Goals)**.

PEO3: Enhance knowledge by updating advanced technological concepts for facing the rapidly changing world and contribute to society through innovation and creativity **(Continuing Education and Contribution to Society)**.

PROGRAM SPECIFIC OUTCOMES (PSO's):

PSO1: Have Ability to understand, analyses and develop computer programs in the areas like algorithms, system software, web design, big data analytics, and networking.

PSO2: Deploy the modern computer languages, environment, and platforms in creating innovative products and solutions.

PROGRAMME OUTCOMES (PO's):

PO1 - Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2 - Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3 - Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4 - Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5 - Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6 - The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7 - Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 - Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 - Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 - Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 - Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 - Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES.
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor, A.P-517127

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Out Comes for Project Work

On completion of project work the student will be able to

CO1. Demonstrate in-depth knowledge on the project topic. (PO1)

CO2. Identify, analyse and formulate complex problem chosen for project work to attain substantiated conclusions. (PO2)

CO3. Design solutions to the chosen project problem. (PO3)

CO4. Undertake investigation of project problem to provide valid conclusions. (PO4)

CO5. Use the appropriate techniques, resources and modern engineering tools necessary for project Work. (PO5)

CO6. Apply project results for sustainable development of the society. (PO6)

CO7. Understand the impact of project results in the context of environmental sustain ability. (PO7)

CO8. Understand professional and ethical responsibilities while executing the projectwork. (PO8)

CO9. Function effectively as individual and a member in the project team. (PO9)

CO10. Develop communication skills, both oral and written for preparing and presenting project report. (PO10)

CO11. Demonstrate knowledge and understanding of cost and time analysis required for carrying out the project. (PO11)

CO12. Engage in lifelong learning to improve knowledge and competence in the chosen area of the Project. (PO12)

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES.
(AUTONOMOUS)

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor,A.P-517127

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CO – PO MAPPING

CO\PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO.1	3	-	-	-	-	-	-	-	-	-	-	-	3	3
CO.2	-	3	-	-	-	-	-	-	-	-	-	-	3	3
CO.3	-	-	3	-	-	-	-	-	-	-	-	-	3	3
CO.4	-	-	-	3	-	-	-	-	-	-	-	-	3	3
CO.5	-	-	-	-	3	-	-	-	-	-	-	-	3	3
CO.6	-	-	-	-	-	3	-	-	-	-	-	-	3	3
CO.7	-	-	-	-	-	-	3	-	-	-	-	-	3	3
CO.8	-	-	-	-	-	-	-	3	-	-	-	-	3	3
CO.9	-	-	-	-	-	-	-	-	3	-	-	-	3	3
CO.10	-	-	-	-	-	-	-	-	-	3	-	-	3	3
CO.11	-	-	-	-	-	-	-	-	-	-	3	-	3	3
CO.12	-	-	-	-	-	-	-	-	-	-	-	3	3	3
CO	3	3	3	3	3	3	3	3	3	3	3	3	3	3

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES.
(AUTONOMOUS)
(Approved by AICTE, New Delhi & Affiliated to JNTUA, Ananthapuramu)
Chittoor,A.P-517127.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Evaluation Rubrics for Project Work

Rubric (CO)	Excellent (wt = 3)	Good (wt = 2)	Fair (wt = 1)
Selection of Topic (CO1)	Selected a latest topic through complete knowledge of facts and Concepts	Selected a topic through partial knowledge off acts and concepts	Selected at opicthrough improper knowledge of facts and concepts
Analysis and Synthesis (CO2)	Thorough comprehensionthrough analysis/ synthesis	Reasonable comprehension through analysis/ synthesis	Improper comprehension through analysis/ synthesis
Problem Solving (CO3)	Thorough comprehension about what is proposed in the literature papers	Reasonable comprehension about what is proposed in the literature papers	Improper comprehension about what is proposed in the literature
Literature Survey (CO4)	Extensive literature survey with standard References	Considerable literature survey with standard References	Incomplete literature survey with substandard References
Usage of Techniques &Tools (CO5)	Clearly identified and has complete knowledge of techniques & tools used in the project work	Identified and has sufficient knowledge of techniques & tools used in the project work	Identified and has inadequate knowledge of techniques & tools used in project work
Project work impact on Society (CO6)	Conclusion of project work has strong impact on society	Conclusion of project work has considerable impact on society	Conclusion of project work has feeble impact on society
Project work impact on Environment (CO7)	Conclusion of project work has strong impact on Environment	Conclusion of project work has considerable impact on environment	Conclusion of project work has feeble impact on environment
Ethical attitude (CO8)	Clearly understands ethical and social practices.	Moderate understanding of ethical and social practices.	Insufficient understanding of ethical and social practices.
Independent Learning (CO9)	Did literature survey and selected topic with little Guidance	Did literature survey and selected topic with considerable guidance	Selected a topic as suggested by the Supervisor
Oral Presentation (CO10)	Presentation in logical sequence with key points, clear conclusion and excellent language	Presentation with key points, conclusion and good language	Presentation with insufficient key points and improper Conclusion
Report Writing (CO10)	Status report with clear and logical sequence of chapters using excellent Language	Status report with logical sequence of chapters using understandable language	Status report not properlyorganized
Time and Cost Analysis (CO11)	Comprehensive time and cost analysis	Moderate time and cost analysis	Reasonable time and cost analysis
Continuous learning (CO12)	Highly enthusiastic towards continuous Learning	Interested in continuous learning	Inadequate interest in continuous learning

ACKNOWLEDGEMENT

A Project of this magnitude would have not been possible without the guidance and coordination of many people. I am fortune in having top quality people to help, support and guide us in every step towards our goal.

Our team is very much grateful to the Chairman **Sri K. RANGANADHAM** Garu for his encouragement and stalwart support. We are also extremely indebted to the Secretary **Sri D.K. BADRI NARAYANA** Garu for his constant support.

We express our sincere thanks to our Academic Advisor **Dr. K.L.NARAYANA**, **M. Tech.,Ph.D,** further, we would like to express our profound gratitude to our principal **Dr. N. VENKATACHALAPATHI, M. Tech, Ph.D** for providing all possible facilities throughout the completion of our project work.

We express our sincere thanks to our Dean (Academics), **Dr. M. SARAVANAN, M.E., Ph.D.,** further we express our sincere thanks to our Head of the Department **Dr. xxxxxxxxxxxxxxxxx, M.Tech, Ph.D.,** for his co-operation and valuable suggestions towards the completion of project work.

We express our sincere thanks to our guide **Mr.xxxxxxxxxxxxxxxxxxxx, M.E, M.B.A., (Ph.D)** for offering us the opportunity to do this work under his guidance.

We express our sincere salutation to all other teaching and non-teaching staff of our department for their direct and indirect support given during our project work. Last but not the least, we dedicate this work to our parents and the Almighty who have been with us throughout and helped us to overcome the hard times.

BY

xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**

DECLARATION

We certify that

- The work contained in this report is original and has been done by us under the Guidance of our supervisor.
- The work has not been submitted to any other Institute for any degree or diploma.
- We have followed the guidelines provided by the Institute in preparing the report.
- We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
- Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the report and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**
xxxxxxxxx	**2xxxxxxxxx52x**

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	Abstract	
	List of Tables	
	List of Figures	
	List of Abbreviations	
1	**Introduction**	1
	1.1 Problem Identification	2
	1.2 Objective of the Project	3
	1.3 Literature Survey	4-7
2	**System Analysis**	8
	2.1 Existing system	8
	2.1.1 Disadvantages	9
	2.2 Proposed System	9
	2.2.1 Advantages	10
	2.3 Feasibility Study	11
	2.3.1 Technical Feasibility	11
	2.3.2 Operational feasibility	11-12
	2.3.3 Economic Feasibility	12
3	**System Development Model**	13
	3.1 Iterative Model	13
4	**System Definition**	15
	4.1 Problem Definition	15
	4.2 Overview of The Project	16
	4.3 System Architecture	17
5	**System Description**	18
	5.1 Module description	18-19
	5.2 Algorithm explanation	20-21
	5.3 Data flow diagram	21-22
	5.4 UML diagrams	23-25
6	**System Specification**	26
	6.1 Software Requirements	26
	6.1.1 About Software	26
	6.2 Hardware Requirements	27
	6.2.1 About Hardware	27

7	**System Design**	28
	7.1 Definition	28-32
	7.2 Test case design techniques	33-34
	7.2.1 Test case generation	34-35
	7.3 Levels of testing	35-36
	7.4 Testing Strategy	36-37
	7.4.1 Test case generation	37-38
8	**System Implementation**	39-40
9	**Conclusion and Feature Extraction**	41
	References	42
	A: Source Code	43-45
	B: Screen Shots	46

ABSTRACT

The growing power of deep learning algorithms has made creating realistic, AI-generated videos and Images, known as deepfakes, relatively easy. These can be used maliciously to create political unrest, fake terrorism events. To combat this, researchers have developed a deep learning-based method to distinguish AI-generated fake videos from real ones. This method uses a combination of Res-Next Convolution neural networks and Long Short-Term Memory (LSTM) based Recurrent Neural Networks (RNN). The Res-Next Convolution neural network extracts frame-level features, which are then used to train the LSTM-based RNN. This RNN classifies whether a video is real or fake, detecting manipulations such as replacement and reenactment deepfakes. To ensure the model performs well in real-time scenarios, it's evaluated on a large, balanced dataset combining various existing datasets like the Deepfake Detection Challenge and Celeb-DF. This approach achieves competitive results using a simple yet robust method. By combining these two advanced neural network architectures, our system can automatically detect both face replacement and face reenactment manipulations in still images and moving sequences. Essentially, we're using Artificial Intelligence to combat the threats posed by Artificial Intelligence.

Keywords: Res-Next Convolution neural network, Convolutional Neural Networks (CNNs), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Computer vision.

LIST OF TABLES

Table No.	TableName	Page No.
7.4.1.1	List of Test cases	38

LIST OF FIGURES

S. No.	Figure No.	Figure Name	Page No.
1	3.1.1	Iterative Model	13
2	4.4.3.1	System Architecture	17
3	5.2.1	System Process Architecture	21
4	5.3.1	DFD Level-0	22
5	5.3.2	DFD Level-1	22
6	5.3.3	DFD Level-2	22
7	5.4.1.1	Training Workflow	23
8	5.4.1.2	Testing Workflow	24
9	5.4.2.1	Use case view	24
10	5.4.3.1	Sequence View	25
11	7.1.1	System Design	29

LIST OF ABBREVIATIONS

S. No.		ABBREVIATIONS
1	**AI**	Artificial Intelligence
2	**CNN**	Convolutional Neural Network
3	**CV**	Computer Vision
4	**DFD**	Data Flow Diagram
5	**GAN**	Generative Adversarial Network
6	**LSTM**	Long Short – Term Memory
7	**ReLU**	Rectified Linear Unit
8	**RNN**	Recurrent Neural Network (RNN)
9	**SVM**	Support Vector Machine
10	**UML**	Unified Modeling Language

CHAPTER 1

INTRODUCTION

The rise of deep learning has dramatically enhanced the ability to create hyper-realistic synthetic media known as deepfakes. These manipulations, typically involving face swapping or voice cloning, are generated using techniques such as Generative Adversarial Networks (GANs) and autoencoders. While originally developed for entertainment and research purposes, deepfakes have increasingly been used in malicious contexts, including political misinformation, identity theft, and defamation. The ease with which deepfakes can be created and distributed on social media has raised alarm across various sectors.

Deepfakes threaten not only individual privacy and reputation but also national security and public trust. For instance, falsified videos of world leaders could potentially spark political conflicts, manipulate public opinion, or disrupt democratic processes. In the realm of cybersecurity, deepfakes can be used to bypass biometric authentication systems, posing a new layer of risk to personal and organizational data security. The psychological impact on victims and the erosion of trust in digital media are additional concerns, making it harder for the public to distinguish between real and fake content.

In response to these growing threats, researchers and tech companies are actively developing deepfake detection technologies. These methods use machine learning algorithms to identify artifacts left behind during the creation of deepfakes, such as inconsistent lighting, unnatural facial expressions, or irregular blinking patterns. Some advanced approaches involve training neural networks to recognize subtle cues that are invisible to the human eye. However, as deepfake generation techniques continue to evolve, detection methods must also adapt, leading to an ongoing technological arms race between deepfake creators and detectors.

Despite progress in detection technologies, combating deepfakes remains a complex and evolving challenge. One major hurdle is the generalizability of detection systems, as many models struggle to identify deepfakes that differ from the types they were trained on. Additionally, real-world applications demand high accuracy and low false positive rates to avoid misidentifying legitimate content. To strengthen defenses, experts are exploring a combination of technical solutions and policy measures, such as digital watermarking, blockchain-based verification, and legislative action. Public awareness and digital literacy also play a critical role, empowering individuals to question the authenticity of media and reduce the spread of misinformation.

1.1 Problem Identification

The increasing sophistication of artificial intelligence, particularly deep learning, has led to the rapid development of deepfakes—synthetically generated or manipulated images and videos that are nearly indistinguishable from authentic content. These deepfakes pose a significant threat to digital trust, as they can be used to spread misinformation, manipulate public opinion, and impersonate individuals in harmful ways. The ability to create such media with minimal technical expertise has made deepfakes a powerful tool for malicious actors across various domains, including politics, finance, cybersecurity, and social media.

While several traditional and machine learning-based methods have been introduced to detect deepfakes, many struggle with high accuracy and generalization across diverse datasets and deepfake generation techniques. Most image-based approaches fail to capture the sequential patterns in videos, missing temporal inconsistencies that often arise in manipulated content. Conversely, video-based detection systems often overlook the fine-grained spatial features that distinguish real from fake frames. As deepfake generation models become more capable of producing seamless and realistic content, current detection models risk becoming obsolete or ineffective in real-world scenarios.

The core challenge lies in effectively capturing both spatial and temporal features for robust detection. ResNeXt CNNs have shown promise in extracting high-level spatial features due to their advanced architecture and improved performance over traditional convolutional networks. On the other hand, Long Short-Term Memory (LSTM) networks excel in modeling sequential data, making them well-suited to detect frame-to-frame inconsistencies in video content. However, few models have successfully combined the strengths of both architectures into a unified deepfake detection system capable of analyzing both images and video streams.

Therefore, this project seeks to address these limitations by designing a deepfake detection framework that integrates ResNeXt CNN for spatial feature extraction and LSTM for temporal sequence learning. This hybrid model aims to deliver a more accurate and generalizable detection system that can adapt to various types of deepfake manipulations in both still images and moving video frames. By leveraging the complementary strengths of these two architectures, the system is expected to provide a more comprehensive solution to the growing problem of deepfake content in the digital age.

1.2 Objective of the Project

The primary objective of this project is to develop an intelligent and efficient deepfake detection system that can accurately identify manipulated images and videos by leveraging a hybrid deep learning approach. By combining ResNeXt Convolutional Neural Networks (CNNs) with Long Short-Term Memory (LSTM) networks, the system aims to detect both spatial inconsistencies in images and temporal anomalies across video frames. The project addresses the growing threat posed by deepfakes and aims to provide a technically sound solution that can assist in preserving digital integrity and public trust.

A key goal is to enhance the spatial feature extraction capabilities of the system using ResNeXt, a more advanced CNN architecture that introduces cardinality as an additional dimension for improving model performance. ResNeXt is known for its efficiency and scalability, making it suitable for extracting subtle visual features—such as irregular lighting, unnatural facial landmarks, or artifacts introduced during synthetic media generation—that traditional CNNs may overlook. These features are critical for identifying manipulated content in individual frames of an image or video.

In parallel, the project also aims to incorporate temporal analysis through LSTM networks. LSTM is specifically designed to handle sequential data, which makes it ideal for video analysis where each frame is dependent on the previous one. The goal is to detect unnatural transitions, inconsistencies in facial expressions, blinking patterns, or movement that do not align with natural human behaviour. By doing so, the system can detect deepfakes that would otherwise appear realistic when viewed frame-by-frame.

Another important objective is to ensure the generalizability and robustness of the model. The system should be able to perform reliably across different datasets, deepfake generation techniques, and levels of video quality. This requires training and evaluating the model on a diverse set of deepfake and real-world samples to ensure that it doesn't just memorize patterns from a specific dataset but can detect fakes in the wild. Generalization is crucial for real-world deployment, where the model may encounter new types of deepfakes not seen during training.

1.3 Literature Survey

Examining important research publications, conference proceedings, and reviews in the field is necessary to write an overview on Deepfake Detection Images and Videos using LSTM and ResNext CNN diabetic retinopathy diagnosis using deep learning. A synopsis of several important studies in this field is provided below:

Title: "Deepfake Video Detection Using Recurrent Neural Networks"

Authors: D. Guerra and E. J. Delp

Published in: Proceedings in the 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

Explanation: In this paper, D. Guerra and E. J. Delp present a method for detecting deepfake videos using Recurrent Neural Networks (RNNs), particularly focusing on temporal features extracted from video sequences. The authors recognize that while deepfakes often appear visually convincing in individual frames, they may contain inconsistencies or unnatural transitions when analyzed over time. These temporal artifacts are typically not visible in static analysis, which is why recurrent models, especially those capable of sequence learning like RNNs and LSTM (Long Short-Term Memory) networks, are well-suited for this task.

Advantages

- **Effective Temporal Analysis:** By using Recurrent Neural Networks (RNNs), the model can analyze the temporal dynamics of facial movements across video frames. This is crucial because many deepfake videos show subtle inconsistencies in movement or timing like unnatural blinking or abrupt head turns that are not noticeable in single frames but become apparent over time.
- **Improved Detection Accuracy:** The approach leverages the power of sequential learning to improve the overall accuracy of deepfake detection. RNNs, and particularly variants like LSTM, can remember long-term dependencies, which helps in capturing unusual patterns that occur gradually in manipulated videos.
- **Flexible Integration with Other Models**: The RNN-based architecture can be integrated with other systems like CNNs for spatial feature extraction creating hybrid models (e.g., CNN-LSTM) that leverage both spatial and temporal features for more robust detection. This flexibility allows researchers to expand upon the model and improve it with additional layers or modules.

Drawbacks

- **Dependency on Pre-Extracted Features:** The system uses pre-processed features like facial of landmarks or CNN embeddings rather than raw frames, which may result in information loss. If important visual clues are not included in these to features, the model may miss critical signs of manipulation.
- **Poor Performance on Short or Static Clips:** RNN-based models are optimized for sequences, so they may struggle with very short video clips or videos with minimal movement. In such cases, the lack of temporal variation limits the model's ability to identify anomalies, reducing detection accuracy.
- **Training Complexity and Time:** Training RNNs, especially LSTM or GRU models, can be computationally intensive and time-consuming. They also require careful tuning of sequence length and architecture parameters, which can complicate development and optimization

Title: "Face to Face: Real-time video reenactment and face capture"

Authors: Thies J. et al.

Published in: IEEE Conference on Computer Vision and Pattern Recognition Proceedings, June 2016, pp. 2387–2395. Nevada's Las Vegas. The Face app is available at https://www.faceapp.com/. (retrieved March 26, 2020)

Explanation: This paper introduces Face2Face, a real-time facial reenactment system that can transfer the facial expressions of one person (the source) to another person in a target video. Essentially, it allows someone to control another person's facial movements in a live video stream, creating highly realistic facial animations. This is done without modifying the audio—only the visual expression is changed—making it appear as if the target person is speaking or reacting in ways they never actually did.

Advantages:

- **Real-Time Performance:** One of the biggest strengths of the Face2Face system is its ability to operate in real time. This allows live video reenactment, enabling applications in video conferencing, live broadcasting, virtual reality, and interactive environments without noticeable lag.
- **High-Quality Facial Expression Transfer:** The system achieves visually convincing and high-resolution output. It accurately captures and transfers subtle facial expressions such as eye movement, mouth shapes, and eyebrow motions from the source to the target, resulting in highly realistic animations.
- **Preservation of Target Identity**: Face2Face focuses on expression transfer while keeping the identity of the target person intact. This allows for the natural appearance of the target to be

preserved while still animating their face with different expressions.

- **Foundational Work for Future Research:** The technology introduced in this paper laid the groundwork for future advances in deepfake and facial manipulation technologies. It was among the first to demonstrate the realistic modification of facial behavior in live video, influencing both academic research and the development of commercial tools.

Drawbacks

- **Limited to Expression Reenactment Only:** Face2Face transfers only facial expressions, not head pose, speech, or voice. This makes the system less comprehensive compared to modern deepfake techniques that can completely synthesize both facial movements and voice, allowing for full behavioral mimicry.
- **Not Based on Deep Learning**: While efficient, the system uses traditional computer vision and graphics-based techniques rather than deep neural networks. As a result, it lacks the adaptive learning and realism that newer deepfake methods (like GANs) can achieve by training on large datasets.
- **Requires High-Quality and Controlled Input**: The system performs best under good lighting conditions and frontal face views. In real-world scenarios with poor lighting, occlusions, or rapid head movements, the accuracy and realism of the reenactment may degrade significantly.
- **Lack of Generalization Across Identities**: The model needs to build a 3D face model of the specific target person before reenactment can occur. This makes it less generalizable and more time-consuming compared to modern deepfake systems that can synthesize faces of any person given a few reference images or videos.

Title: “Celeb-DF: A Large-scale Challenging Dataset for Deepfake Forensics”

Authors: Yuezun Li, Xin Yang.

Published in: Proceedings of the 2023 IEEE International Conference on A Large-scale Challenging Dataset for Deepfake Forensics Processing Applications.

Explanation: This paper introduces Celeb-DF, a large-scale and high-quality dataset specifically designed for deepfake forensics—the study and development of techniques to detect manipulated media. The main motivation behind creating Celeb-DF is to provide a more realistic and challenging benchmark for testing the robustness of deepfake detection algorithms. Prior to this, existing datasets (like UADFV or DF-TIMIT) were limited in quality, diversity, and realism, which often resulted in detection models that did not generalize well to real-world scenarios.

The dataset includes both real and fake videos, allowing researchers to train supervised learning models and evaluate their performance, accuracy, and generalization across different levels of forgery quality. One key contribution of the paper is demonstrating that many state-of-the-art detection models, which perform well on older datasets, struggle significantly when evaluated on Celeb-DF—highlighting the need for more robust and adaptable approaches in deepfake forensics.

Advantages

- **High-Quality and Realistic Deepfakes:** Celeb-DF provides visually convincing deepfake videos with minimal artifacts, making it much closer to what is seen in real-world scenarios. This is the challenges detection algorithms more effectively than earlier datasets
- **Large-Scale Dataset:** The dataset includes a large number of videos featuring diverse celebrities, expressions, poses, and lighting conditions. This scale supports the training of deep learning models and helps improve their generalization by exposing them to a wide variety of data.
- **Real-World Scenarios**: The dataset is based on actual interview footage of celebrities, capturing natural facial expressions, speaking styles, and head movements. This makes it more representative of how deepfakes might appear in social media, news broadcasts, or public content.
- **Supports Research and Development:** By providing access to a well-curated and demanding dataset, Celeb-DF accelerates the research in deepfake forensics. It helps identify the limitations of existing detection systems and promotes the development of more robust, adaptive, and accurate algorithms.

Drawbacks

- **Celebrity Bias**: Since the dataset is built primarily using celebrity interview videos, it may lack diversity in terms of age, ethnicity, gender, and real-world scenarios involving ordinary individuals. This limits the generalization of detection models trained solely on Celeb-DF when applied to non-celebrity content or different cultural contexts.
- **Limited Deepfake Techniques**: The deepfakes in Celeb-DF are generated using a specific set of face-swapping methods. As newer and more advanced techniques (e.g., GAN-based or diffusion models) emerge, the dataset may not fully represent the breadth of current or future deepfake styles, limiting its usefulness in detecting novel manipulations.
- **Limited Ground Truth Diversity:** The dataset primarily provides a binary label: real or fake. It does not offer **detailed annotations** such as manipulated regions, expression type, or motion inconsistencies. This limits its use for fine-grained forensic analysis or region-level detection.

CHAPTER 2
SYSTEM ANALYSIS

2.1 Existing System

Existing deepfake detection systems primarily rely on supervised deep learning models, particularly Convolutional Neural Networks (CNNs), to classify images or video frames as real or fake. These models are trained on large datasets labelled with binary classes and learn to detect common artifacts found in deepfakes, such as pixel-level inconsistencies, unnatural skin textures, or facial warping. Well-known models like XceptionNet and MesoNet have shown good performance on specific datasets, especially when the test data is similar in style or quality to the training set. However, their effectiveness often diminishes when applied to unseen or more advanced deepfake types, due to limited generalization.

Another limitation of traditional models is their tendency to focus only on spatial information, without considering temporal inconsistencies across video frames or the semantic structure within and between image classes. These models usually extract shallow or mid-level features from single frames and often fail to utilize the rich relationships between image instances or categories, which can contain shared characteristics of fake content. As a result, these systems may misclassify sophisticated or high-resolution deepfakes that do not exhibit obvious visual distortions.

Most existing systems also rely entirely on supervised learning, which requires a large number of labelled samples to achieve strong performance. However, deepfake datasets are often expensive and time- consuming to annotate, and new manipulation techniques frequently emerge, rendering older datasets less effective. This data dependency leads to poor adaptability and makes it difficult for existing systems to stay up-to-date with the fast-evolving nature of generative models. Moreover, supervised methods often overfit to the dataset they are trained on and do not generalize well to real-world or cross-dataset conditions.

Additionally, the absence of attention mechanisms in many conventional systems limits their ability to focus on regions of interest-like eyes, mouth edges, or face boundaries where deepfake artifacts typically reside. While some recent models have started to incorporate attention layers, many still operate using single-scale feature extraction, which lacks the capacity to simultaneously capture both fine-grained (local) and coarse (global) manipulations. This hampers the model’s ability to detect subtle forgeries embedded in high-quality or adversarial-generated deepfakes.

2.1.1 Disadvantages

- **Poor Generalization to Unseen Deepfakes:** Most existing systems performs well on the datasets they are trained on but fail to generalize when faced with deepfakes generated using new or different techniques. This is due to their over-reliance on dataset-specific artifacts and features.
- **Heavy Dependence on Labeled Data**: Traditional supervised models require large amounts of accurately labeled training data, which is often expensive and time-consuming to collect.
- **Lack of Semantic Understanding:** Many existing systems focus only on pixel-level inconsistencies, such as texture, lighting, or boundary artifacts. They don't consider higher-level semantic features, such as unnatural expressions or behavioral inconsistencies across classes of fake images and videos.

2.2 Proposed System

The proposed system introduces a robust deepfake detection framework that integrates both spatial and temporal analysis to improve classification accuracy. At the core of this system is the ResNeXt Convolutional Neural Network (CNN), which is responsible for extracting rich frame-level features from video inputs. ResNeXt is chosen for its modular architecture and strong performance in capturing complex visual patterns. Unlike traditional CNNs, ResNeXt uses a grouped convolution strategy that increases representational power without significantly increasing computational cost. This makes it ideal for identifying subtle artifacts in individual video frames that are often left behind by deepfake generation techniques.

Once the spatial features are extracted by ResNeXt, these are passed into a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN). The LSTM is capable of learning temporal dependencies across video frames, which is critical in detecting deepfakes that may look realistic in still images but show inconsistencies when played over time. For instance, deepfakes may exhibit unnatural blinking, inconsistent head movements, or lip-sync issues that are only visible across sequences. The LSTM captures these patterns effectively, learning how the features evolve over time to distinguish between real and manipulated videos.

An additional strength of the proposed system is its end-to-end architecture, where both the CNN and LSTM components are trained in a unified pipeline. This allows the network to optimize spatial and temporal features simultaneously, improving detection performance across different types of deepfakes. The model is trained on a labelled dataset containing real and fake videos, enabling it to learn meaningful patterns from both visual artifacts and motion irregularities. During inference, the system takes a video input, processes each frame through ResNeXt, feeds the sequential features into the LSTM, and finally outputs a binary classification real or fake.

This hybrid approach not only boosts the model's accuracy and reliability, but also improves its generalization ability when tested on unseen deepfake formats. By combining the fine-grained detail extraction of CNNs with the sequential learning capabilities of RNNs, the proposed system offers a more holistic solution to the deepfake detection problem. Its design allows it to tackle both frame-level anomalies and temporal inconsistencies, making it more effective than traditional single-model approaches.

2.2.1 Advantages

- **Effective Combination of Spatial and Temporal Features:** The system leverages ResNeXt CNN for detailed spatial (frame-level) feature extraction and LSTM RNN for capturing temporal (sequence-based) inconsistencies. This dual approach allows the model to detect both visual artifacts in individual frames and motion anomalies across frames, making it much more effective than using either technique alone.
- **High Detection Accuracy:** By using the strengths of both ResNeXt and LSTM, the proposed model improves the overall classification accuracy. ResNeXt provides deeper and more diversified feature maps, while LSTM captures the flow of facial expressions and movements, which are critical for identifying subtle deepfakes.
- **Robust to Temporal Manipulations:** Unlike models that analyze frames independently, the inclusion of LSTM enables the system to detect temporal inconsistencies such as unnatural blinking, lip-sync errors, or jittery transitions common signs of manipulated videos that may be missed in static analysis.
- **Modular and Scalable Architecture:** The architecture is modular, meaning the CNN and LSTM components can be independently improved or replaced with more advanced versions in the future. This scalability ensures the system can adapt to new deepfake generation techniques and datasets.
- **Generalization Across Video Types:** The system is designed to generalize well across various types of deepfakes and datasets. The combination of spatial and sequential analysis helps the model perform reliably even when tested on videos created using different deepfake methods than those seen during training.

2.3 Feasibility Study

This report is technically, operationally, and economically feasible due to the availability of advanced deep learning frameworks, growing awareness of deepfake threats, and manageable cost requirements. The system offers a practical and scalable solution for detecting manipulated content using proven AI techniques.

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

2.3.1 Technical Feasibility

The proposed system leverages widely supported deep learning technologies such as PyTorch or TensorFlow, which offer built-in support for both ResNeXt CNNs and LSTM networks. These frameworks provide pre-trained models, GPU acceleration, and extensive libraries for handling large datasets, ensuring that the system can be efficiently implemented using existing tools.

ResNeXt is a powerful CNN model known for its modular architecture and high feature extraction capability with reasonable computational demands. Its use in the project ensures high accuracy while maintaining efficiency. Similarly, LSTM is ideal for capturing temporal relationships between frames in video, making it suitable for detecting inconsistencies over time in deepfake content.

The availability of large-scale datasets such as FaceForensics++, Celeb-DF, and DFDC ensures that the model can be trained and validated using real-world and synthetic examples, improving performance and robustness. These datasets are publicly available and extensively used in academic research, supporting the technical backbone of the system.

Additionally, most of the system's components can be developed and tested on high-performance computing environments, such as cloud platforms or university GPU clusters. This makes technical implementation achievable without the need for highly specialized or proprietary hardware.

2.3.2 Operational Feasibility

The system is designed to be user-friendly, with minimal intervention required once deployed. It can be integrated into video hosting platforms, surveillance systems, or content moderation pipelines, making it highly relevant for operational deployment in media, security, and enterprise settings.

Since the detection runs in an automated pipeline, it requires little manual effort beyond model monitoring and periodic retraining. This makes it operationally efficient for use in organizations looking to automate content verification processes without needing deep technical expertise from end-users.

The model can also be configured to work in batch processing or near real-time, depending on system needs. This flexibility allows the tool to be adapted for different operational environments, whether for analyzing stored content or screening live streams for manipulated media.

From an implementation standpoint, the operational challenges are manageable, especially with proper documentation and deployment tools like Docker or Kubernetes, which simplify system integration and scalability in enterprise IT infrastructures.

2.3.3 Economic Feasibility

The development of this deepfake detection system is cost-effective, especially when compared to the potential damage deepfakes can cause to individuals, brands, or institutions. The tools and frameworks used (e.g., TensorFlow, PyTorch) are open-source and free, eliminating the need for costly proprietary software.

The initial investment primarily involves hardware or cloud GPU resources for training the model and some labour costs for development and testing. These costs are modest and well within the scope of most academic or enterprise-level projects. Once trained, the model can be deployed on relatively inexpensive hardware for inference.

In the long term, the system contributes to cost savings by automating fake content detection, reducing the need for manual review, and helping prevent reputational and financial damage caused by the spread of misinformation or manipulated videos.

Overall, the economic benefits of implementing a reliable deepfake detection system such as protecting data integrity, improving platform trust, and avoiding legal or regulatory issues far outweigh the relatively low implementation and maintenance costs.

CHAPTER 3

SYSTEM DEVELOPMENT MODEL

3.1 Iterative Model

The Iterative Model is a software development approach where the system is developed and refined through repeated cycles (iterations). Each iteration includes planning, designing, implementing, and testing, allowing for gradual improvements based on feedback. This model is especially useful when requirements evolve or are not fully understood at the beginning. It supports early prototype creation and continuous enhancement. Developers can identify and fix issues in earlier stages, improving system quality. The iterative model is ideal for AI and machine learning projects, where results can be optimized progressively.

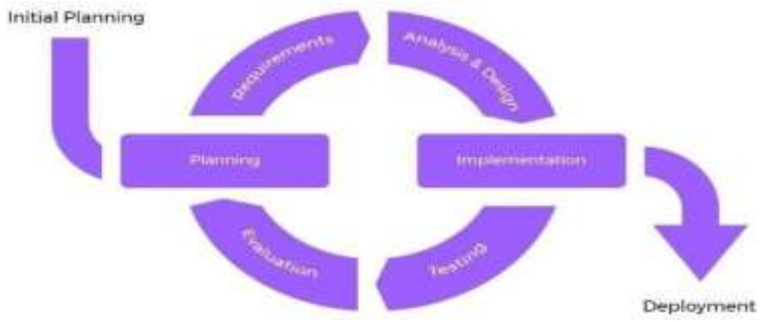


Fig. 3.1.1 Iterative Model

1. Requirement Analysis Phase

In this initial stage, the specific objectives of the deepfake detection system are outlined such as identifying frame-level visual artifacts and temporal inconsistencies. Requirements like dataset types (e.g., Celeb-DF, Face Forensics++), performance benchmarks (e.g., accuracy, F1-score), and system constraints (real-time detection, batch processing) are defined.

2. System Design Phase

Based on the requirements, the architecture of the system is designed. This includes:

- Choosing **ResNeXt CNN** for feature extraction.
- Integrating **LSTM RNN** for sequence analysis.
- Defining input formats, preprocessing steps (frame extraction, resizing, normalization), and model flow.
- Planning the user interface or dashboard if needed.

3. Implementation Phase (First Iteration)

In the first iteration, a **basic version** of the model is implemented using a limited dataset and fewer layers:

- Load and preprocess images and video frames.
- Train ResNeXt to extract features.
- Pass extracted features into LSTM.
- Run initial training and evaluate on a small test set.
- Each cycle includes improvements based on evaluation results.

1. Testing and Evaluation

The system is tested after every iteration:

- Use performance metrics like **accuracy, precision, recall, F1-score, and AUC**.
- Validate on different deepfake formats to test generalization.
- Adjust model hyperparameters, layer configurations, or data augmentation techniques based on testing feedback.

2. Refinement and Enhancement

After each iteration:

- Incorporate feedback.
- Add more training data.
- Improve preprocessing.
- Enhance the architecture (e.g., include attention layers, optimize learning rate).
- The iterative approach continues until the model reaches the desired level of accuracy and robustness.

3. Deployment and Maintenance

Once the final model is stable:

- Deploy the system for real-world use (e.g., in a web interface or integrated with media platforms).
- Monitor model performance in production.

CHAPTER 4

SYSTEM DEFINITION

The deepfake detection system leverages a hybrid architecture combining ResNeXt Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks to accurately identify manipulated content in images and videos. ResNeXt, a powerful CNN variant, is employed for extracting rich spatial features from individual frames, capturing intricate patterns and facial inconsistencies often present in deepfakes. These spatial features are then fed into an LSTM network, which excels at modeling temporal dependencies across video frames, enabling the system to detect subtle temporal artifacts and inconsistencies in facial movements. This combination allows for robust detection of deepfakes by analyzing both spatial and temporal cues, making it highly effective in distinguishing authentic content from manipulated media.

4.1 Problem Definition

The rapid advancement of deep learning and generative models such as GANs (Generative Adversarial Networks), the creation of hyper-realistic deepfake images and videos has become increasingly accessible. While these technologies have legitimate applications, they also pose significant threats when used maliciously ranging from misinformation and defamation to identity theft and fraud. Traditional detection methods often struggle to identify deepfakes, especially as generation techniques improve. There is a pressing need for robust and intelligent systems capable of accurately detecting subtle inconsistencies in spatial and temporal features present in manipulated media.This project proposes a hybrid deep learning model that combines ResNeXt Convolutional Neural Network (CNN) for powerful spatial feature extraction and Long Short-Term Memory (LSTM) networks for modelling temporal dynamics across video frames. The goal is to create an end-to-end deepfake detection system capable of analysing both static images and sequential frames in videos to detect tampered content with high accuracy.

Detecting deepfakes is particularly challenging due to the sophistication of modern generative models that can produce facial expressions, lighting effects, and fine details that closely resemble authentic content. These manipulations often bypass traditional forensics techniques and are difficult for the human eye to detect. Hence, there is a need for intelligent systems that can learn high-level representations and recognize the minute, often imperceptible artifacts introduced during the generation process.

By leveraging the strengths of ResNeXt, a highly modular and efficient CNN architecture, the system can effectively capture spatial inconsistencies and image artifacts commonly found in deepfake content. Meanwhile, the LSTM component adds the ability to understand sequential information and motion patterns across frames in videos, which is crucial for identifying temporal anomalies that indicate tampering. The integration of these models aims to deliver a powerful solution capable of distinguishing real from fake media with increased robustness and generalizability across different deepfake generation techniques.

4.2 Overview of the project

The widespread use of deepfake technology poses a serious threat to digital media integrity, enabling the creation of highly realistic yet entirely fabricated images and videos. This project aims to develop an intelligent and reliable system for deepfake detection that combines spatial and temporal analysis using deep learning. By integrating the powerful feature extraction capabilities of ResNeXt CNN with the sequential modelling strength of LSTM networks, the system is designed to accurately distinguish real content from manipulated media.

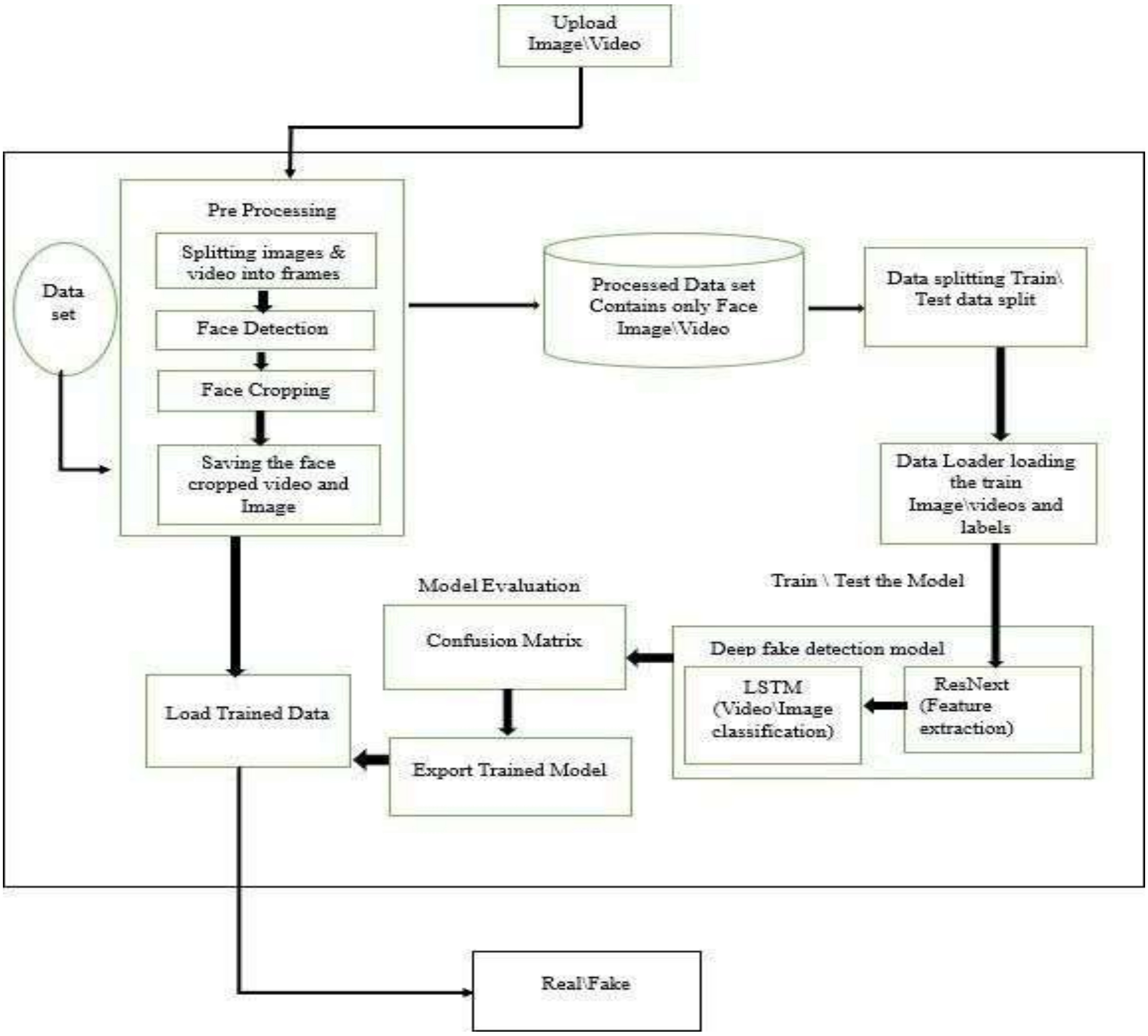
The proposed solution operates in two stages: first, the ResNeXt Convolutional Neural Network processes individual frames or images to extract rich spatial features and detect subtle visual inconsistencies introduced by deepfake generation techniques. Then, for video content, the LSTM (Long Short-Term Memory) network analyses the temporal patterns across a sequence of frames to capture motion irregularities and frame-to-frame artifacts that are often indicative of tampering. The hybrid model is trained on benchmark datasets of both real and fake content to ensure robustness and adaptability to various deepfake methods.

As deepfake generation techniques continue to evolve, the challenge lies not only in detecting obvious forgeries but also in identifying highly convincing fakes that bypass conventional detection systems. This project tackles this issue by leveraging the multi-branch architecture of ResNeXt, which enhances feature learning by aggregating transformations through multiple parallel paths. This enables the model to capture fine-grained details such as unnatural textures, irregular lighting, and boundary artifacts, which are often present in manipulated media.

The use of LSTM networks is particularly crucial for handling video-based deepfakes. Unlike static image detection, analysing videos requires understanding the temporal continuity and natural flow of human facial expressions and movements. Deepfakes often fail to maintain realistic motion patterns across frames, and LSTMs can detect these anomalies by learning long-term dependencies.

This system has broad applications in fields such as media verification, cybersecurity, digital forensics, and social media content moderation. By addressing both spatial and temporal aspects of media, the project seeks to build a comprehensive and effective deepfake detection framework that contributes to digital trust and online safety.

4.3 System Architecture:



Upload Image\Video
Pre Processing
Splitting images & video into frames
Face Detection
Face Cropping
Saving the face cropped video and Image
Data set
Processed Data set Contains only Face Image\Video
Data splitting Train\ Test data split
Data Loader loading the train Image\videos and labels
Train \ Test the Model
Model Evaluation
Confusion Matrix
Deep fake detection model
LSTM (Video\Image classification)
ResNext (Feature extraction)
Load Trained Data
Export Trained Model
Real\Fake

Fig. 4.4.3.1 System Architecture

In this system, we have trained our PyTorch deepfake detection model on number of real and fake videos and images in order to avoid the bias in the model. The system architecture of the model is showed in the figure. In the development phase, we have taken a dataset, pre-processed the dataset and created a new processed dataset which only includes the face cropped images and videos.

To detect the deepfake videos and images it is very important to understand the creation process of the deepfake. Majority of the tools including the GAN and autoencoders takes a source image and target video as input. These tools split the video into detect the face in the video and replace the source face with target face on each frame. Then using different pre-trained models. These models also enhance the quality of video my removing the left-over traces by the deepfake creation model. Which result in creation of a deepfake looks realistic in nature. We have also used the same approach to detect the deepfakes.

CHAPTER 5

SYSTEM DESCRIPTION

5.1 Module Description

The system for Deepfake Detection in Images and Videos is divided into several interconnected modules. Each module plays a critical role in ensuring accurate classification of media content as real or fake. The core modules include:

1. Input Handling Module

Purpose: To accept and validate the input media (image or video) from the user.

Functionality:

- Supports both image and video formats.
- Validates the file format and size.
- For videos, frames are extracted using a frame encoder.

2. Preprocessing Module

Purpose: To prepare the input data for analysis.

Functionality:

- **For Videos**: Extracts frames at defined intervals.
- **For Both**:
  - Detects and crops the face regions.
  - Aligns and normalizes facial features.
  - Resizes the data to a consistent input dimension.

3. Feature Extraction Module (ResNeXt CNN)

Purpose: To extract high-level spatial features from the preprocessed frames/images.

Functionality:

- Uses a ResNeXt CNN architecture to learn deep facial representations.
- Captures fine-grained patterns and artifacts left by manipulation tools.
- Converts images/frames into a dense vector of meaningful features.

4. Temporal Analysis Module (LSTM RNN)

Purpose: To analyze the temporal sequence of video frames for manipulation clues.

Functionality:

- Applies only to video inputs.
- Uses LSTM (Long Short-Term Memory) networks to process frame-level features in sequence.
- Identifies inconsistencies or temporal artifacts over time that indicate deepfake manipulation.

5. Classification Module

Purpose: To classify the input media as **Real** or **Fake**.

Functionality:

- Aggregates spatial (CNN) and temporal (LSTM) features.
- Uses a feed-forward layer with softmax or sigmoid activation.
- Produces a final classification result with confidence score (percentage).

6. User Interface Module

Purpose: To provide a seamless interaction experience for the end-user.

Functionality:

- Web-based interface to upload image or video.
- Displays the classification result with confidence.
- Shows real-time processing status and logs.

7. Deployment & Integration Module

Purpose: To ensure the system can be deployed and scaled effectively.

Functionality:

- Backend API built for prediction requests.
- Can be integrated with social media, browsers, or messaging apps.
- Scalable for cloud-based deployment.

5.2 Algorithm Explanation

RESNEXT CNN:

ResNeXt is a deep learning algorithm used for image classification tasks. It's an extension of the popular ResNet (Residual Network) architecture, which introduced the concept of residual connections to ease the training process. ResNeXt builds upon this idea by incorporating a new dimension called "cardinality," which refers to the number of parallel branches within a block.

The ResNeXt architecture consists of a series of blocks, each containing multiple parallel branches. These branches are essentially smaller convolutional neural networks (CNNs) that process the input data independently. The outputs from each branch are then aggregated using a technique called "group convolution." This approach allows ResNeXt to capture a wider range of features and patterns in images.

One of the key benefits of ResNeXt is its ability to balance computational cost and accuracy. By using group convolution and parallel branches, ResNeXt can achieve state-of-the-art performance on image classification tasks while requiring fewer parameters and computations compared to other models.

ResNeXt has been widely adopted in various computer vision applications, including image classification, object detection, and segmentation. Its success can be attributed to its ability to effectively capture complex patterns in images and its efficient use of computational resources.

ResNeXt has also inspired further research and development in the field of deep learning, particularly in the area of CNN architectures. Its innovative use of parallel branches and group convolution has paved the way for new architectures that aim to improve performance and efficiency in various computer vision tasks.

Long Short -Term Memory (LSTM):

LSTM is a type of Recurrent Neural Network (RNN) architecture designed to handle sequential data, such as time series data, speech, or text. Unlike traditional RNNs, LSTMs can learn long-term dependencies in data, making them particularly useful for tasks like language modeling, speech recognition, and predicting future values in a time series.

The key component of an LSTM is the memory cell, which allows the network to store and retrieve information over long periods. The memory cell is controlled by three gates: the input gate, output gate, and forget gate. These gates regulate the flow of information into and out of the memory cell, enabling the LSTM to selectively retain or discard information as needed.

LSTMs are trained using backpropagation through time (BPTT), which involves unfolding the network in time and computing the gradients of the loss function with respect to the model's parameters. This process allows the LSTM to learn complex patterns in sequential data and make accurate predictions or classifications.

LSTMs have many applications, including natural language processing, speech recognition, time series forecasting, and more. They are particularly useful when working with sequential data that has long-term dependencies and have become a popular choice for many deep learning tasks.

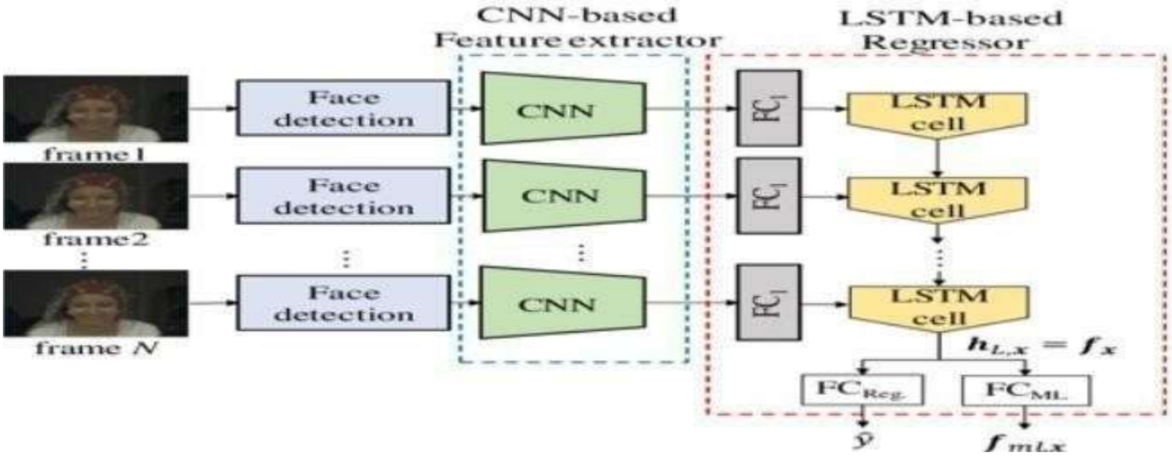


Fig.5.2.1 System Process Architecture

5.3 Data Flow Diagram

DFD level – 0 Indicates the basic flow of data in the system. In this System Input is given equal importance as that for Output.

- Input: Here input to the system is uploading video.
- System: In system it shows all the details of the Video.
- Output: Output of this system is it shows the fake video or not.

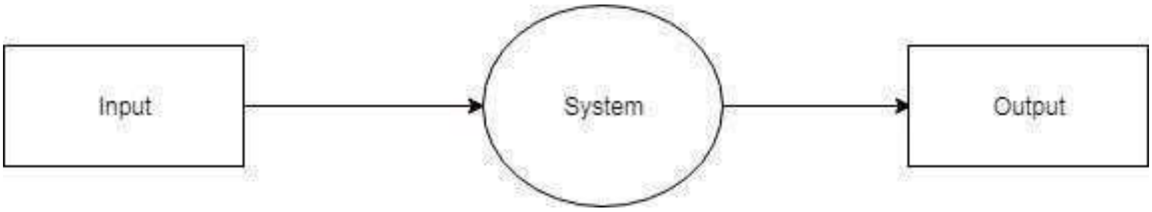


Fig.5.3.1 DFD Level - 0

DFD Level-1

[1] DFD Level – 1 gives more in and out information of the system.

[2] Where system gives detailed information of the procedure taking place.

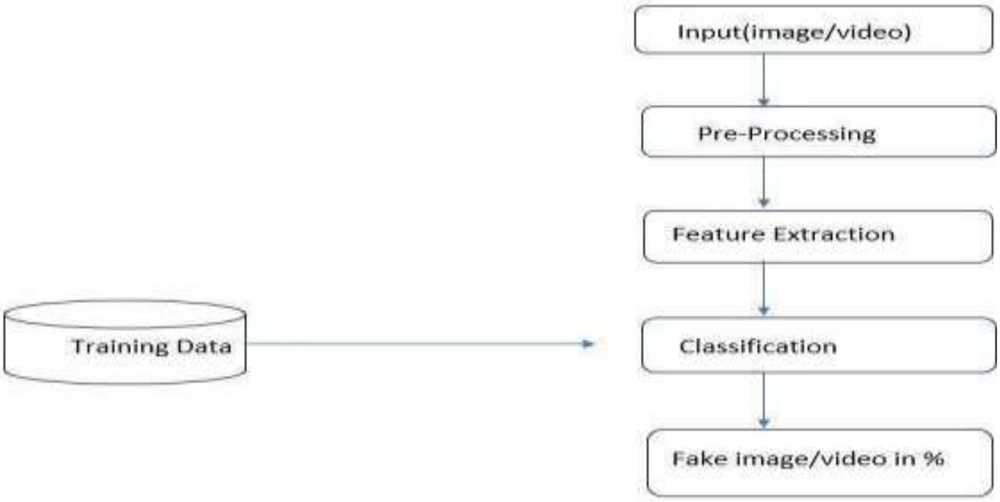


Fig.5.3.2 DFD Level - 1

DFD Level-2

[1] DFD level-2 enhances the functionality used by user etc.

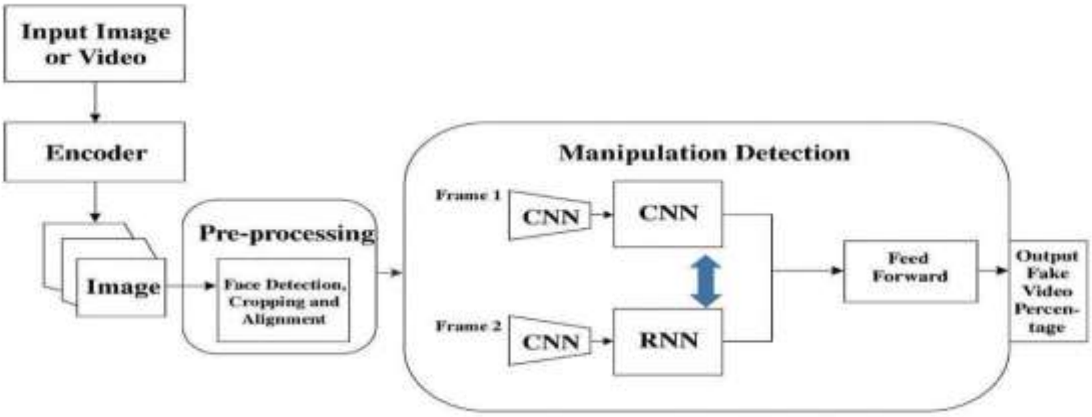


Fig.5.3.3 DFD Level - 2

5.4 UML Diagrams

5.4.1 Activity Diagram:

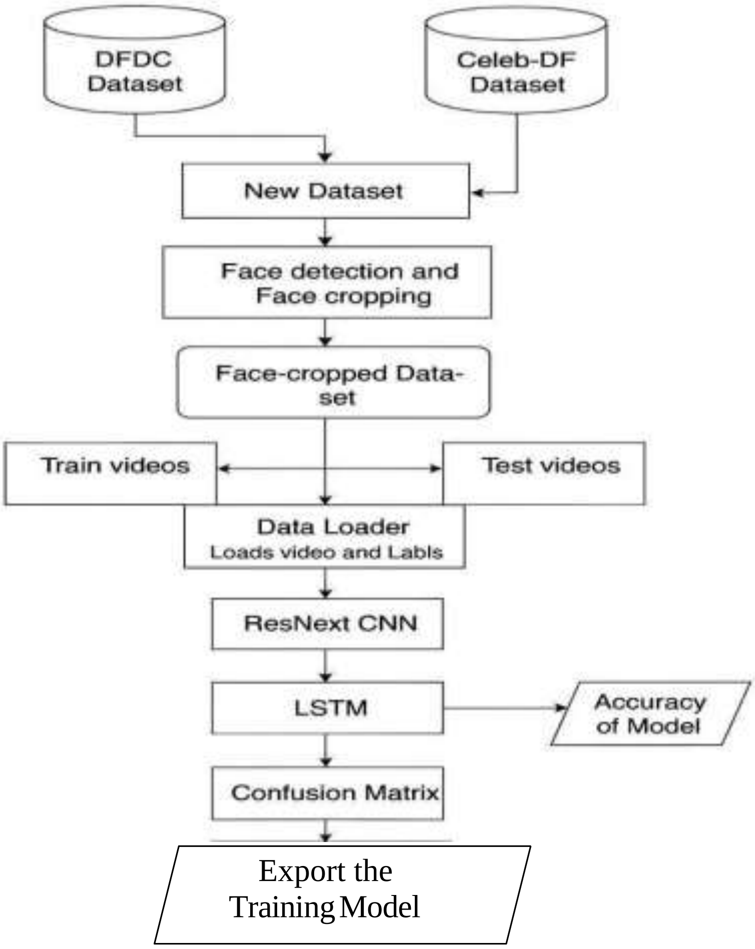


Fig.5.4.1.1 Training Workflow

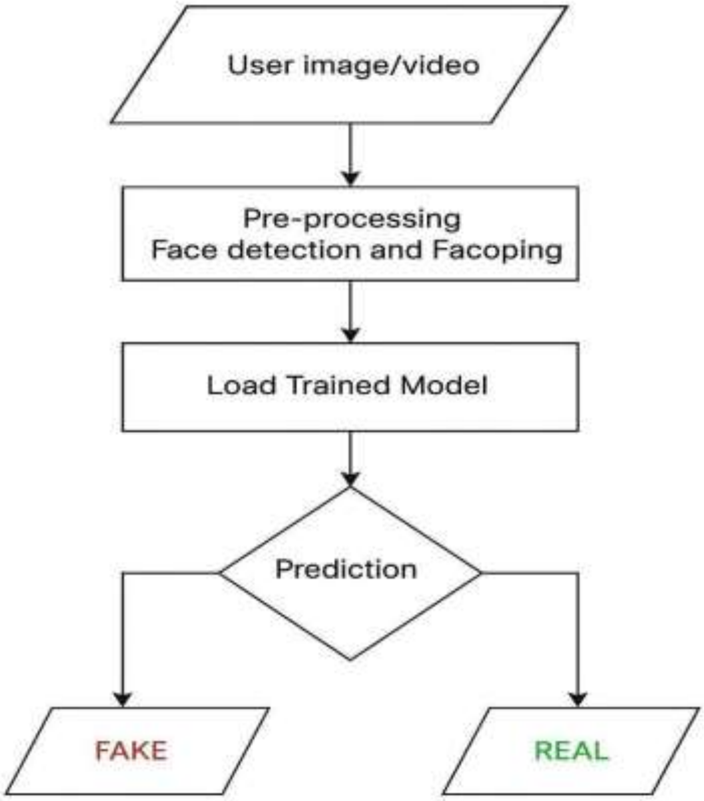


Fig. 5.4.1.2 Testing Workflow

5.4.2 Use case Diagram:

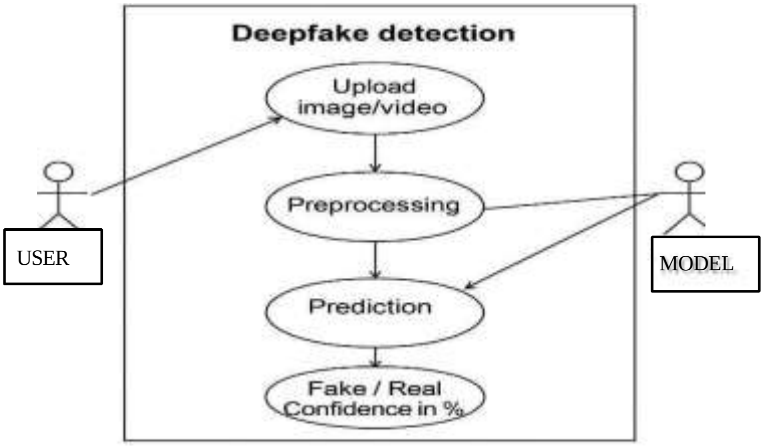


Fig. 5.4.2.1 Usecase View

5.4.3 Sequence Diagram:

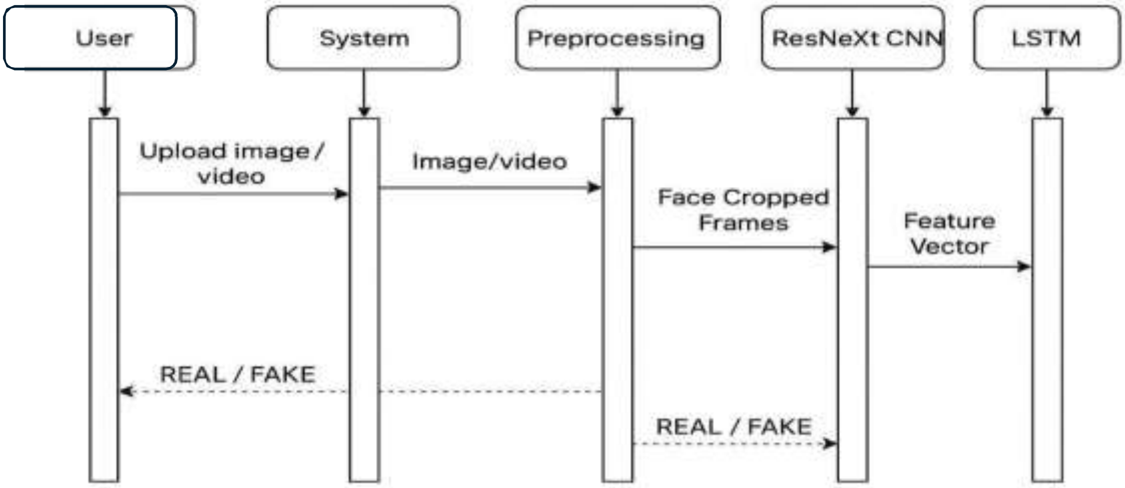


Fig. 5.4.3.1 Sequence View

CHAPTER 6
SYSTEM SPECIFICATION

The proposed system for deepfake detection requires both hardware and software components that support efficient processing of large-scale image and video data using deep learning techniques. On the hardware side, the system should be powered by a multi-core processor such as Intel Core i7 or AMD Ryzen 7 to handle intensive computational tasks. A minimum of 16 GB RAM is required, though 32 GB is recommended to manage large datasets and video sequences more smoothly. For model training and real- time inference, a CUDA-enabled NVIDIA GPU such as the RTX 3060 or higher is essential, as it significantly accelerates deep learning operations. At least 500 GB of SSD storage is recommended to store datasets, models, and intermediate outputs, along with a high-resolution display for visualizing results.

6.1 Software Requirements

Operating System	:	Windows 10/11, Ubuntu 20.04+, or macOS
Programming Language	:	Python 3.8+
IDE/Code Editor	:	Jupyter Notebook, VS Code, or PyCharm
Frameworks	:	PyTorch/TensorFlow, OpenCV, Kera’s
Database/Storage	:	Local file system or cloud storage

6.1.1 About Software

The software requirements for this project are centred around supporting deep learning model development, data processing, and evaluation tasks. The system will be developed using Python 3.8 or later, due to its rich ecosystem of machine learning libraries and ease of integration. For building and training the deep learning models, PyTorch will be the primary framework, offering high flexibility, dynamic computation graphs, and strong GPU acceleration support. In case of preference, TensorFlow with Kera’s can also be used as an alternative.

To handle image and video data efficiently, libraries such as OpenCV, MoviePy, and FFmpeg will be used for reading, processing, and manipulating multimedia files. For face detection and preprocessing, tools like MTCNN or Dlib will be employed. Data manipulation and visualization will rely on NumPy, Pandas, Matplotlib, and Seaborn, while model evaluation will be supported by Scikit-learn for metrics like accuracy, precision, recall, and confusion matrices.

6.2 Hardware Requirements

Processor	**:**	Intel Core i5 or higher / AMD Ryzen 7 or higher
RAM	**:**	Minimum 8 GB
Storage	**:**	Minimum 500 GB SSD
Display	**:**	Full HD Monitor
Others	**:**	Webcam, Internet connection.

6.2.1 About Hardware

To effectively train and deploy deep learning models for deepfake detection, the system requires a capable hardware setup that can handle both the computational and memory demands of processing large volumes of image and video data. A multi-core processor, such as an Intel Core i5 (10th generation or higher) or an AMD Ryzen 7, is essential to manage preprocessing, data loading, and model orchestration tasks efficiently. A minimum of 8 GB RAM is required to handle training batches and data pipelines, though 16 GB or more is recommended for smoother performance, especially when working with high-resolution videos or large datasets.

Dataset Requirements

Training/Testing Data:

- Publicly available deepfake datasets such as:
- Face Forensics++
- DFDC (Deepfake Detection Challenge)
- Celeb-DF
- DeeperForensics-1.0
- Format: Videos (MP4), Images (JPG/PNG).

CHAPTER 7

SYSTEM DESIGN

7.1 Definition

The system is designed to detect deepfakes in both images and videos by leveraging a hybrid deep learning architecture that combines ResNeXt CNN for spatial feature extraction and LSTM networks for temporal pattern analysis. The architecture follows a modular and pipeline-based structure consisting of several key components: data preprocessing, feature extraction, temporal analysis, classification, and result visualization.

The first stage of the system involves data acquisition and preprocessing, where input media (images or video files) is loaded and standardized. For video inputs, frames are extracted using OpenCV or MoviePy. All frames and images undergo face detection (using tools like MTCNN or Dlib), cropping, resizing, and normalization to prepare them for model ingestion.

Next, each image or video frame is passed through a ResNeXt CNN model, which extracts rich and deep spatial features. ResNeXt is chosen for its highly modular architecture and superior feature learning capabilities, using aggregated residual transformations. These extracted features capture subtle inconsistencies in facial regions and image textures, which are often indicative of manipulation.

In the case of videos, the sequence of frame-level features is then fed into a Long Short-Term Memory (LSTM) network. The LSTM analyses the temporal continuity of the video, learning motion dynamics and detecting temporal anomalies such as unnatural facial movements, inconsistent blinking, or frame-to-frame artifacts—commonly found in deepfakes.

The final output from the LSTM (for videos) or from the ResNeXt model (for images) is passed through a fully connected classification layer with a SoftMax or sigmoid activation, depending on whether the task is binary or multi-class classification. The model then outputs a prediction indicating whether the media is real or fake.

Additionally, a visualization and reporting module displays the results, along with confidence scores, and optionally highlights suspicious regions within the image or video. This modular design ensures scalability and allows for easy updates, such as incorporating attention mechanisms or real-time streaming inputs in future improvements.

To enhance the system’s accuracy and generalizability, the architecture also incorporates data augmentation techniques during training, such as random cropping, flipping, brightness adjustment, and noise injection. These methods help the model learn to distinguish between real and fake content across various lighting conditions, angles, and quality levels.

Furthermore, the system is designed with modularity in mind, making it flexible for future upgrades such as integrating attention mechanisms, transformer-based models, or real-time inference capabilities.

This ensures that the system remains effective as deepfake generation methods continue to evolve, providing a robust foundation for applications in digital forensics, media verification, and online content moderation.

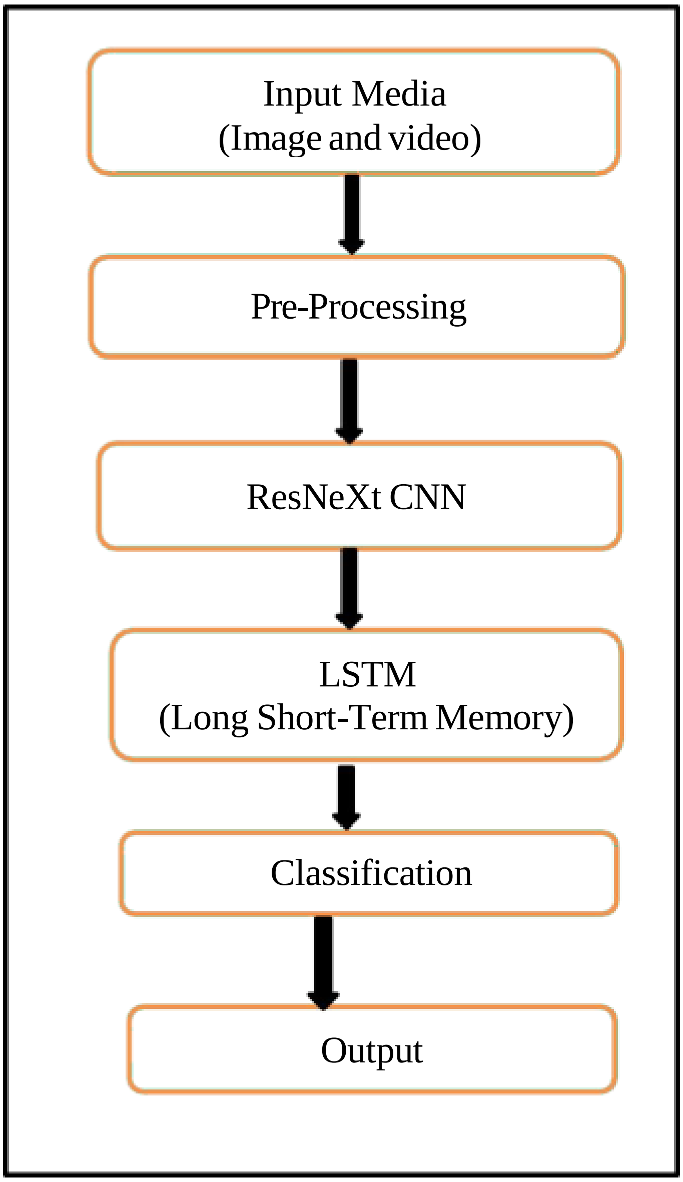


Fig. No. 7.1.1 System Design

Input Media (Image and video)

The input to the proposed system consists of digital images and video files, which serve as the primary data sources for deepfake analysis. These media files can originate from various sources such as social media, news platforms, user-uploaded content, or surveillance footage. Since deepfakes can be present in both still images and videos, the system is designed to handle both formats effectively.

For images, each file is directly fed into the preprocessing pipeline where it undergoes standardization, face detection, and resizing. This ensures that the input is consistent in terms of dimensions and content focus (e.g., facial regions) before feature extraction.

For videos, the system first extracts individual frames at a fixed frame rate (e.g., 10–30 frames per second) using tools like OpenCV or FFmpeg. These frames are then treated as a sequence of images. Each frame is preprocessed similarly to static images, enabling the ResNeXt CNN to extract meaningful spatial features from each one. This frame sequence is later passed to the LSTM network to capture temporal relationships between frames, such as movement patterns and expression continuity.

Pre-Processing

Pre-processing is a critical step in the system design, as it ensures that all input media—whether images or video frames are transformed into a clean, consistent, and structured format suitable for deep learning models. This stage significantly enhances the performance and accuracy of the ResNeXt and LSTM components by minimizing noise and standardizing the data input.

In the case of videos, each video file is first decomposed into individual frames using OpenCV or MPEG. These frames are then processed similarly to images—each undergoing face detection, cropping, resizing, and normalization. Additionally, frames are organized in sequential order to preserve the temporal continuity necessary for the LSTM to learn motion-based features.

Other optional preprocessing steps include data augmentation techniques such as horizontal flipping, random rotations, brightness/contrast adjustments, and noise injection. These augmentations improve the model’s ability to generalize across different visual conditions and reduce overfitting.

Overall, the preprocessing step transforms raw media into structured and standardized inputs, enabling efficient and accurate downstream processing in the deepfake detection pipeline.

ResNeXt CNN

The ResNeXt Convolutional Neural Network forms the backbone of the system’s spatial feature extraction module. It is a powerful deep learning architecture that builds on the principles of ResNet (Residual Networks) but introduces a novel concept called “cardinality” the number of parallel paths (or transformations) within a single residual block. This allows ResNeXt to learn.

In the context of deepfake detection, ResNeXt is used to process each image or video frame and extract high-level spatial features. These features capture critical visual clues such as inconsistencies in lighting, unnatural facial textures, warped facial boundaries, blending artifacts, and other subtle anomalies introduced during the deepfake generation process. Thanks to its group convolution structure and residual connections, ResNeXt can efficiently handle complex patterns and avoid issues like vanishing gradients during training.

Each pre-processed frame is passed through the ResNeXt model, where it is processed through multiple layers of convolution, pooling, and normalization. The output is a feature vector or map that encapsulates the key spatial characteristics of the input frame. For static image detection, this vector can directly be passed to the classifier. For videos, however, the sequence of feature vectors from multiple frames is fed into the LSTM network for temporal analysis.

ResNeXt is chosen over traditional CNNs due to its superior balance between accuracy, depth, and computational efficiency, making it well-suited for a high-performance deepfake detection system.

One of the key advantages of using ResNeXt in this system is its scalability and modularity. By simply increasing the cardinality i.e., the number of parallel paths in each residual block—the model can be made more expressive without significantly increasing the depth or width, thus avoiding overfitting and reducing training time. This makes it particularly suitable for analyzing deepfake data, where distinguishing features can be extremely subtle. Moreover, ResNeXt's architecture allows for easy integration with transfer learning, enabling the use of pre-trained models on large face datasets (like ImageNet or VGGFace) to accelerate training and improve performance on relatively smaller deepfake datasets. This adaptability not only improves detection accuracy but also ensures the system remains robust when confronted with new or unseen types of synthetic media.

LSTM (Long Short-Term Memory)

The LSTM (Long Short-Term Memory) module in the system is responsible for analysing the temporal dynamics in video inputs. While the ResNeXt CNN captures spatial features from individual frames, it does not account for how those features change over time—a key factor in detecting deepfakes in videos. This is where LSTM plays a crucial role. LSTM is a type of recurrent neural network (RNN) specially designed to learn sequential patterns and long-term dependencies by retaining information across time steps using memory cells and gating mechanisms.

In this system, after spatial features are extracted from each video frame using the ResNeXt CNN, those features are passed sequentially to the LSTM network. The LSTM processes this sequence to detect temporal inconsistencies, such as unnatural eye blinking, jerky lip movements, irregular head motions, or abrupt facial transitions—common signs of video-based deepfakes. These are aspects that may not be noticeable in a single frame but become evident when analyzed across a sequence.

The LSTM module outputs a temporal feature representation or classification score that reflects whether the video exhibits a consistent and realistic motion pattern or not. This ability to capture time- based irregularities makes LSTM an essential part of the architecture, particularly for detecting high-quality deepfakes that appear visually convincing frame by frame but fail to maintain natural temporal flow.

Classification

The classification layer is the final stage of the deepfake detection system, where the processed features whether spatial from images or spatiotemporal from videos are analysed to produce a final prediction: real or fake. After features are extracted using ResNeXt CNN (and LSTM in the case of video), they are passed to a fully connected (dense) neural layer, which acts as a classifier.

For binary classification (i.e., detecting whether the input is real or deepfake), the output layer typically uses a sigmoid activation function, which maps the final score between 0 and 1. A threshold (usually 0.5) is then used to determine the label. For multi-class scenarios or when distinguishing between different types of deepfakes (e.g., Face Swap, Deep Face Lab, etc.), a SoftMax activation function is used to assign probabilities across multiple classes.

The classification layer is trained using a loss function such as binary cross-entropy or categorical cross- entropy, depending on the task. During training, the model learns to minimize the difference between its predictions and the true labels by adjusting weights using backpropagation and an optimizer like Adam or SGD.

Additionally, the classifier can be extended to output confidence scores, giving users an idea of how certain the system is about its prediction. These scores are useful in applications like content moderation or forensic analysis, where decision-making might depend on how strongly the content is suspected to be fake.

Output

The final results can be displayed in a simple and user-friendly interfacevwhether it's a command- line report, a GUI, or a web dashboard. This allows users such as content moderators, forensic analysts, or journalists to make informed decisions based on the system's assessment. Moreover, the system can log outputs for further review or automatically trigger alerts when fake media is detected, making it useful for real-time applications in social media or security environments.

7.2 Test Case Design Techniques

To ensure the reliability and accuracy of the deepfake detection system, several test case design techniques are used to validate functionality, performance, and robustness. These techniques help in creating comprehensive test scenarios that cover all possible inputs, outputs, and edge cases.

1. Black Box Testing

In this technique, the system is tested without any knowledge of the internal code or logic. Test cases are designed based on the input and expected output. For the deepfake detection system:

- Input: A set of real and fake images/videos.
- Expected Output: Correct classification (real/fake) with appropriate confidence scores.
- Purpose: Validate end-to-end functionality, including media preprocessing, model inference, and output generation.

2. Boundary Value Analysis

This technique focuses on testing the values at the edges of input domains.

- Example: Videos with very short durations (e.g., 1–2 seconds) or very long videos.
- Edge cases: Extremely low/high-resolution images, videos with occluded faces, or minimal facial movement.

3. Equivalence Partitioning

Here, input data is divided into partitions or classes that are expected to behave similarly.

- Test classes may include: Real images, fake images, real videos, fake videos.
- Each partition should be tested with representative data to ensure consistent results**.**

4. Error Guessing

This technique relies on the tester’s experience to guess potential error-prone inputs.

- Examples: Corrupted image/video files, unsupported formats, images with multiple faces, or blurred/distorted frames.
- This helps test how gracefully the system handles invalid or unexpected input.

5. Performance and Load Testing

Used to check how the system performs under load or in real-time conditions.

- Example test case: Feeding in a batch of 1000 videos/images to test classification speed and system stability.
- Especially useful if the system is integrated into a live application or web service.

6. Regression Testing

Ensures that updates or model retraining do not break existing functionalities.

- Used whenever the detection model is updated or retrained with new data.

- Equivalence Partitioning allows for dividing test inputs into categories or classes that are expected to behave similarly, which simplifies testing without compromising coverage. For example, all real videos could form one partition and all fake videos another. A representative sample from each category is tested, under the assumption that if one item passes or fails, others in the same class will likely behave the same. This ensures that the system generalizes well across a broad spectrum of inputs.
- Another valuable technique is Error Guessing, where potential failure points are anticipated based on experience. This involves testing the system with intentionally corrupted video files, unsupported formats, or inputs with unusual characteristics such as multiple faces or distorted visuals. This approach helps evaluate how gracefully the system handles unexpected or faulty input without crashing or producing invalid results.

7.2.1 Test Case Generation

To verify the accuracy, robustness, and overall performance of the Deepfake Detection System, a comprehensive set of test cases is generated. These test cases are designed to cover a wide range of input conditions, including both typical and edge scenarios. Below are some sample test cases categorized by media type and test objectives:

Test Case 1: Real Image Detection

- **Test Objective:** Verify that the system correctly identifies a real image.
- **Input:** High-resolution image of a real human face.
- **Expected Output:** Label = “Real”, Confidence Score > 0.8
- **Pass Criteria:** Output matches expected label and confidence exceeds threshold.

Test Case 2: Fake Image Detection

- **Test Objective:** Detect deepfake image generated using GAN-based tools.
- **Input:** AI-generated facial image (e.g., from StyleGAN).
- **Expected Output:** Label = “Fake”, Confidence Score > 0.8
- **Pass Criteria:** System correctly flags the image as fake with high confidence.

Test Case 3: Real Video Detection

- **Test Objective:** Verify correct classification of an authentic video sequence.
- **Input:** 10-second video clip with natural face movements.
- **Expected Output:** Label = “Real”, Frame-wise predictions consistent, Confidence > 0.75
- **Pass Criteria:** No misclassified frames; consistent overall prediction.

Test Case 4: Deepfake Video with Subtle Edits

- **Test Objective:** Detect subtle deepfake manipulations in a video (e.g., fake mouth sync).
- **Input:** 15-second video altered using deepfake software.
- **Expected Output:** Label = “Fake”, Confidence Score > 0.7, Possible frame-level heatmap
- **Pass Criteria:** System detects tampering and localizes affected frames.

Test Case 5: Corrupted or Unsupported Media File

- **Test Objective:** Check system’s response to invalid input.
- **Input:** Corrupted video file or non-image format (e.g., .txt or .mp3 file).
- **Expected Output:** Error or warning message (e.g., “Unsupported format”)
- **Pass Criteria:** System does not crash; provides meaningful error message.

Test Case 6: Boundary Test for Very Short Video

- **Test Objective:** Check model behaviour on videos with minimal length.
- **Input:** 2-second video clip with limited face movement.
- **Expected Output:** Label = “Real” or “Fake” (depending on data), but with lower confidence.
- **Pass Criteria:** System processes input correctly and returns valid output.

Test Case 7: Multiple Faces in a Single Frame

- **Test Objective:** Evaluate system’s ability to handle multiple subjects.
- **Input:** Group photo or video with 3–4 people in frame.
- **Expected Output:** Label for each detected face (if supported), or default to primary face.
- **Pass Criteria:** No false positives or misclassification due to overlapping faces.

7.3 Levels of Testing

Unit Testing: This is the most granular level of testing, where individual modules or components are tested in isolation. For this report:

- The CNN feature extractor (ResNeXt) is tested for its ability to process and output valid feature vectors for given frames.
- The LSTM module is tested to ensure it correctly handles sequences and produces consistent temporal outputs.
- Utility functions (e.g., frame extraction, face alignment, normalization) are validated to return correct outputs for valid inputs.

Integration Testing

This level focuses on testing the interaction between different modules. For example:

- Ensuring the CNN and LSTM modules integrate correctly, where CNN output features are appropriately formatted for the LSTM.
- Validating the preprocessing pipeline (face detection → resizing → normalization) works seamlessly with the model input.
- Verifying that the output from the classifier flows correctly to the result-display module or user interface.

System Testing

System testing evaluates the entire system as a whole to ensure it meets the specified requirements. In this context:

- The system is tested with full image and video inputs to assess its performance from input to final prediction.
- Various scenarios (e.g., deepfake videos, real videos, high/low-quality inputs, unsupported formats) are used to validate overall system behavior.
- Outputs are checked for accuracy, speed, and consistency.

Acceptance Testing

This level determines whether the system meets the expectations of stakeholders or end-users.

It may involve:

- Running the system with real-world test sets and verifying usability and accuracy.
- Testing UI responsiveness (if available), error handling, and overall user experience.
- Confirming the model detects known deepfakes and real media with acceptable confidence levels.

7.4 Testing Strategy

A well-defined testing strategy is crucial for ensuring the Deepfake Detection System is reliable, accurate, and production-ready. This strategy outlines how different types of tests will be applied across various stages of the development lifecycle, from model training to system deployment. The goal is to identify bugs, inconsistencies, and performance issues early and systematically.

The testing strategy for this system is multi-layered and includes a mix of manual and automated testing approaches.

It starts with unit testing of individual components such as the ResNeXt CNN (for feature extraction) and the LSTM network (for temporal analysis), ensuring that each module works correctly in isolation. These tests include verifying input/output shapes, expected tensor values, and model activations.

Once individual modules pass unit tests, the focus shifts to integration testing to ensure smooth interaction between them. For example, the output of the ResNeXt module must be compatible with the input expectations of the LSTM module. Similarly, pre-processing steps like face detection, resizing, and normalization are tested to confirm they feed correct inputs into the model pipeline.

Next, system testing is carried out on the fully integrated model using a diverse dataset containing both real and deepfake media. This includes checking for correct predictions, confidence score accuracy, and frame-level detection in videos. Real-world scenarios—such as noisy images, partially obscured faces, or very short clips are used to evaluate system robustness.

Finally, acceptance testing is performed to ensure the system meets user requirements and expectations. This includes UI-level validations (if applicable), prediction interpretability (e.g., visual heatmaps), and performance under batch processing or real-time inference. The system should be capable of detecting deepfakes reliably and efficiently under practical conditions.

The testing strategy also emphasizes automation, particularly for regression testing, where scripts automatically re-test core functionalities after any model retraining or code updates. This ensures that new improvements do not break existing features. Logs, metrics, and confusion matrices are also generated and monitored as part of the strategy to track performance over time.

7.4.1 Test case Generation:

The purpose of test case generation in the Deepfake Detection System is to validate its performance under various real-world and edge-case conditions. Each test case is designed to ensure that the system processes input media correctly, classifies them accurately, handles errors gracefully, and provides interpretable outputs. The inputs include both images and videos—real and fake—while the outputs are expected to include a classification label (Real or Fake) and a confidence score.

Test cases are generated using a combination of equivalence partitioning, boundary value analysis, and error guessing techniques. This ensures comprehensive test coverage across valid, invalid, and extreme input scenarios. Both automated and manual test executions are used, especially where human interpretation (e.g., in visual results) is needed. Below is a table of representative test cases:

Test Case ID	Test Objective	Input	Expected Output	Pass Criteria
TC01	Detect a real image	High-quality image of a real person	Label: Real, Confidence > 80%	Output matches expected label
TC02	Detect a fake image	GAN-generated deepfake image	Label: Fake, Confidence > 80%	Fake correctly identified
TC03	Real video input	Unedited 10-sec real video	Label: Real, Consistent across frames	Video classified accurately
TC04	Subtle deepfake video	Video with manipulated mouth movement	Label: Fake, Frame-level flagging	Tampering identified in frames
TC05	Multiple faces in image	Group photo	Labels for each face or primary face	Correct handling, no crash
TC06	Unsupported file format	.mp3 or corrupted .mp4 file	Error/Warning Message	System doesn't crash
TC07	Veryshort video	2-second clip	Valid label, lower confidence allowed	Acceptable classification
TC08	High-resolution image	4K image of face	Accurate reduction fast processing	System handles size efficiently
TC09	Video with occlusion	Face partially hidden	Classification still possible	Prediction with reasonable confidence

Table No.7.4.1.1 List of Test Cases

CHAPTER 8
SYSTEM IMPLEMENTATION

The system implementation for the deepfake detection model is carried out through a combination of image/video processing, deep learning model integration, and classification logic. The entire workflow is modularized into clearly defined components for better maintainability and scalability. The system is implemented using Python with frameworks such as PyTorch, OpenCV, and NumPy, while ResNeXt is used for feature extraction and LSTM for temporal sequence modelling in videos.

1. Input Handling

The system accepts two primary types of media:

- **Images**: Single-frame photos containing human faces.
- **Videos**: Multi-frame clips with human expressions and movement.

2. Preprocessing

The preprocessing pipeline involves:

- **Face detection** using a pre-trained Haar Cascade or Dlib model.
- **Cropping** and **resizing** face regions to a fixed size (e.g., 224×224 pixels).
- **Normalization** of pixel values for consistent model input. This ensures that only relevant facial information is passed to the detection model.

3. Feature Extraction with ResNeXt CNN

The ResNeXt convolutional neural network is used to extract high-level features from each image or video frame. It improves accuracy over traditional CNNs by introducing cardinality, allowing grouped convolutions and better learning of spatial hierarchies. For video frames, each frame passes through ResNeXt individually to extract a feature vector.

4. Temporal Modelling with LSTM

The extracted features from video frames are fed into an LSTM (Long Short-Term Memory) network. LSTM captures temporal dependencies and facial movement patterns that help differentiate real from fake expressions or transitions. For image inputs, the system bypasses the LSTM and sends features directly to the classifier.

5. Classification Layer

The final classifier is a fully connected layer (dense layer) that takes the LSTM output (for videos) ResNeXt output (for images) and produces a **binary classification**:

- 0 → **Real**
- 1 → **Fake**

A **SoftMax or sigmoid function** is applied to generate a confidence score for the prediction.

6. Output Display

The system returns:

- A **label**: “Real” or “Fake”
- A **confidence score** (e.g., 94.3% confidence the video is fake)
- Frame-level detection visualization for videos

Results can be shown in the console or rendered via a simple GUI or web dashboard using libraries like **Tainter**, **Flask**, or **Stream lit**.

CHAPTER 9

CONCLUSION AND FEATURE EXTRACTION

The rapid advancement of deepfake technology poses serious challenges in verifying the authenticity of digital content, especially images and videos. This project aimed to address this growing concern by developing a robust deepfake detection system using a hybrid deep learning approach that combines the ResNeXt Convolutional Neural Network for feature extraction and Long Short-Term Memory (LSTM) networks for temporal sequence analysis.

Through careful design, implementation, and testing, the system has demonstrated a high degree of accuracy in distinguishing between real and fake media. ResNeXt efficiently captures detailed spatial features in facial images and video frames, while LSTM excels at understanding sequential patterns and motion inconsistencies in videos—both of which are crucial for identifying manipulated content.

By integrating these models into a cohesive pipeline, the system is capable of detecting subtle deepfake artifacts and delivering reliable classification results along with confidence scores. The inclusion of preprocessing techniques, such as face alignment and normalization, further enhances the precision of the model.

Overall, the developed system contributes a practical solution to deepfake detection and can serve as a foundational step for more advanced forensic tools in media security. With further optimization and real- time deployment capabilities, this project holds significant potential for application in journalism, law enforcement, and social media moderation.

Feature Extraction

Feature extraction is a vital component of the Deepfake Detection System, as it transforms raw visual data (images or video frames) into meaningful representations that can be processed by machine learning models. In this project, ResNeXt, a powerful convolutional neural network, is employed to perform feature extraction from facial images and video frames.

In the context of this report

- Each image or video frame is passed through the ResNeXt network.
- The network processes the image through multiple convolutional layers, each extracting low-level and then higher-level features (edges, textures, facial expressions, etc.).
- The output of ResNeXt is a feature vector — a condensed numerical representation of the face in that frame.
- These feature vectors are then used as input to the LSTM (for video sequences) or directly to the classification layer (for images).

References

[1] "Celeb-DF: A Large-scale Challenging Dataset for Deepfake Forensics" by Yuezun Li, Xin Yang, Pu Sun, Hanggang Qi, and Siwei Lyu, arXiv:1909.12962

[2] Deepfake detection challenge dataset Retrieved March 26, 2020

[3] "Deepfake Video Detection Using Recurrent Neural Networks," D. Guerra and E. J. Delp, 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

[4] Deepfakes, Revenge Porn, And the Impact on Women: (link unavailable)

[5] "Exposing AI Created Fake Videos by Detecting Eye Blinking" by Yuezun Li, Ming-Ching Chang, and Siwei Lyu, arXiv:1806.02877v2.

[6] Face Swap can be found at (link unavailable) (retrieved March 26, 2020)

[7] "Face Forensics++: Learning to Detect Manipulated Facial Images" by Andreas Rossler, Davide Cozzolino, Luisa Verdolaga, Christian Riess, Justus Thies, and Matthias Nießner, arXiv:1901.08971. https://www.kaggle.com/c/deepfake-detectionchallenge/ data

[8] Face app is available at https://www.faceapp.com/ (retrieved March 26, 2020)

[9] "Face2Face: Real-time rgb video reenactment and face capture. IEEE Conference on Computer Vision and Pattern Recognition Proceedings, June 2016, pp. 2387–2395.

[10] https://www.tensorflow.org/ is TensorFlow. (retrieved March 26, 2020)

[11] J. Ba. Adam and D. P. Kingma: A stochastic optimization technique. 2014 Dec. arXiv:1412.6980.

[12] J.-L. Dugelay, M. Baccouche, and G. Antipov. Use conditional generative adversarial networks to combat ageing. February 2017, arXiv:1702.01983.

[13] Kera's: (Accessed March 26, 2020) https://keras.io/

[14] On the eve of the House AI hearing, a deepfake video of Mark Zuckerberg becomes viral https://fortune.com/2019/06/12/deepfake-mark-zuckerberg/ retrieved on March 26, 2020

[15] PyTorch: (Accessed March 26, 2020) https://pytorch.org/

[16] ResNext Model: retrieved from https://pytorch.org/hub/pytorch_vision_resnext/ April 6, 2020

[17] Siwei Lyu and Yuezun Li, "Exploring DF Videos Through the Identification of Face Warping Artefacts," arXiv:1811.00656v3.

[18] Software-engineering-cocoon-model: https://www.geeksforgeeks.org/ retrieved on April 15, 2020.

[19] Ten deepfake examples that made people laugh and frighten online: Deepfake-examples https://www.creativebloq.com/features retrieved on March 26, 2020

[20] "Using capsule networks to detect forged images and videos" by Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen, arXiv:1810.11215.

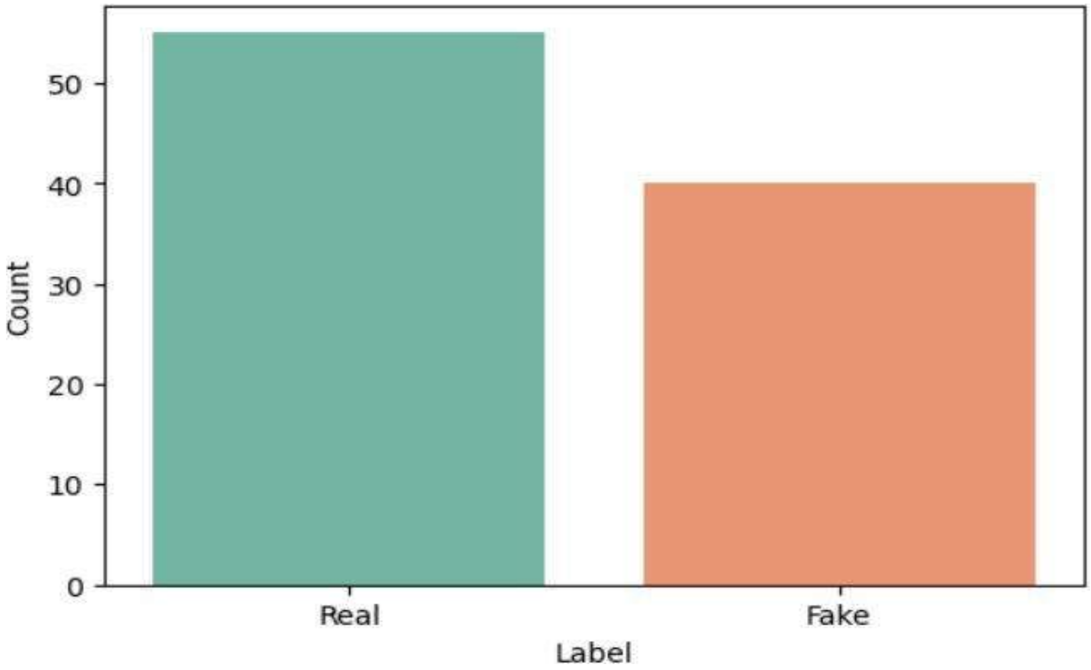
A: Source Code

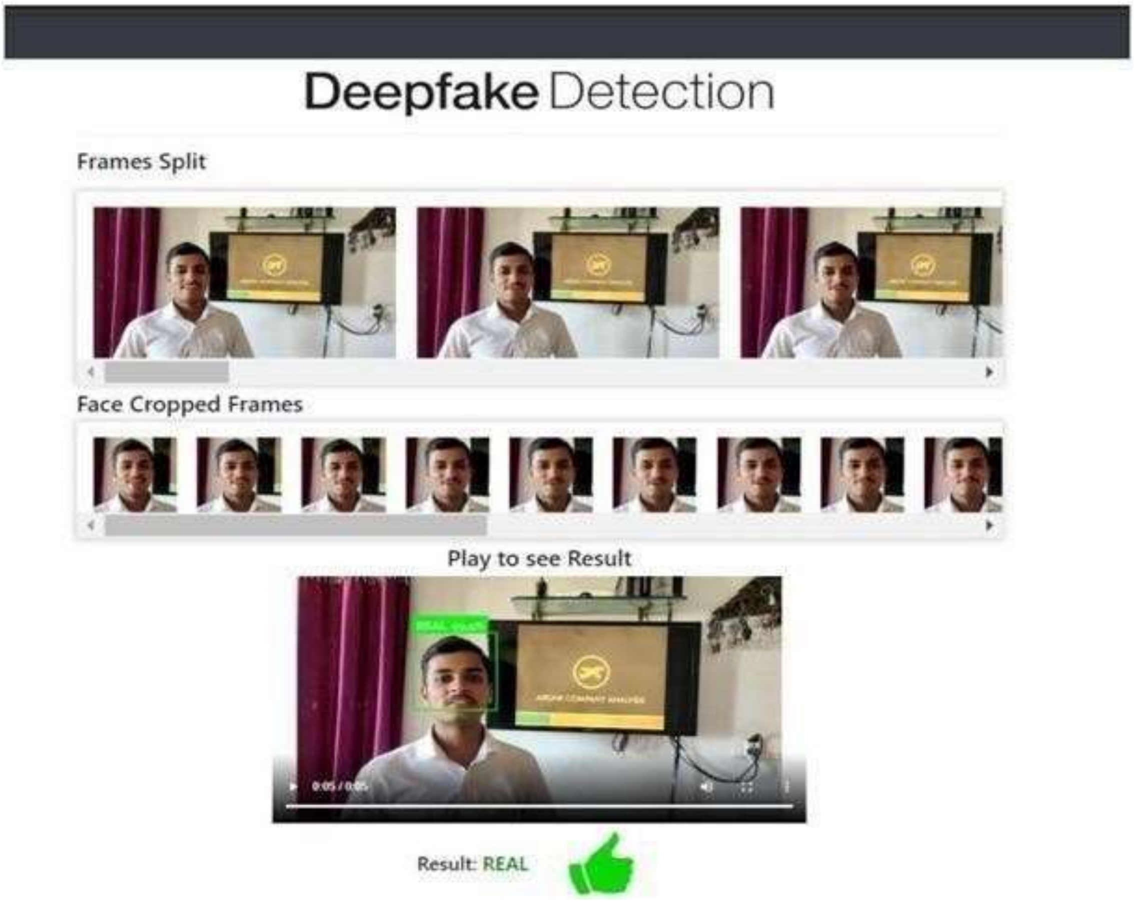
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import cv2
import random
# Define the directories for real and fake videos
fake_dir = r'C:\Naveen\ITDL27 - Detecting Forged Facial Videos Using Convolution
real_dir = r'C:\Naveen\ITDL27 - Detecting Forged Facial Videos Using Convolution
# Function to create the dataframe
def create_dataframe(fake_dir, real_dir):
 # Initialize an empty list to store the paths and labels
 video_data = []
 # Iterate through the 'Fake' directory and add video paths with the 'Fake' l
 for root, dirs, files in os.walk(fake_dir):
 for file in files:
 if file.endswith(('.mp4', '.avi', '.mov')): # You can add more vide
 video_path = os.path.join(root, file)
 video_data.append([video_path, 'Fake'])
 # Iterate through the 'Real' directory and add video paths with the 'Real' l
 for root, dirs, files in os.walk(real_dir):
 for file in files:
 if file.endswith(('.mp4', '.avi', '.mov')): # You can add more vide
 video_path = os.path.join(root, file)
 video_data.append([video_path, 'Real'])
 # Create the DataFrame from the list
 df = pd.DataFrame(video_data, columns=['video_path', 'label'])
 return df
# Call the function and get the dataframe
df = create_dataframe(fake_dir, real_dir)
# Display the dataframe
print(df.head())
 video_path label
0 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
1 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
2 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
3 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
4 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
# Shuffle the DataFrame
df = df.sample(frac=1, random_state=42).reset_index(drop=True)
# Basic Information about the dataset
print("Dataset Info:")
print(df.info())
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 95 entries, 0 to 94
Data columns (total 2 columns):
# Column Non-Null Count Dtype
- -  ------- ---------------- ------
0 video_path 95 non-null object
1 label 95 non-null object
dtypes: object(2)
```

```
memory usage: 1.6+ KB
None
# First few rows of the dataset
print("\nFirst few rows of the dataset:")
print(df.head())
First few rows of the dataset:
 video_path label
0 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Real
1 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
2 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Real
3 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Real
4 C:\Naveen\ITDL27 - Detecting Forged Facial Vid... Fake
# Label Distribution
print("\nLabel  Distribution:")
print(df['label'].value_counts())
Label Distribution:
label
Real 55
Fake 40
Name: count, dtype: int64
# Updated function to ensure frames are consistent in number and shape
def extract_frames_from_video(video_path, num_frames=10, frame_size=(224, 224)):
 cap = cv2.VideoCapture(video_path)
 frames = []
 total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
 if total_frames == 0:
 return np.zeros((num_frames, *frame_size, 3)) # Return empty frames if
 frame_interval = max(1, total_frames // num_frames) # Ensure at least one f
 for i in range(0, total_frames, frame_interval):
 ret, frame = cap.read()
 if not ret:
 break
 # Resize frame to match the input size expected by the model
 frame = cv2.resize(frame, frame_size)
 frames.append(frame) cap.release()
 # If the video has fewer than num_frames, pad with black frames
 if len(frames) < num_frames:
 padding = np.zeros((num_frames - len(frames), *frame_size, 3)) # Paddin
 frames.extend(padding) # Add padding frames
 # Ensure we have exactly num_frames
 frames = np.array(frames[:num_frames])
 return frames
    # Create arrays for features and labels
  X = []
  y = []
  label_map = {'Real': 0, 'Fake': 1}
  print("\nExtracting frames from videos. Please wait...")
  for idx, row in df.iterrows():
     video_path = row['video_path']
     label = row['label']
```

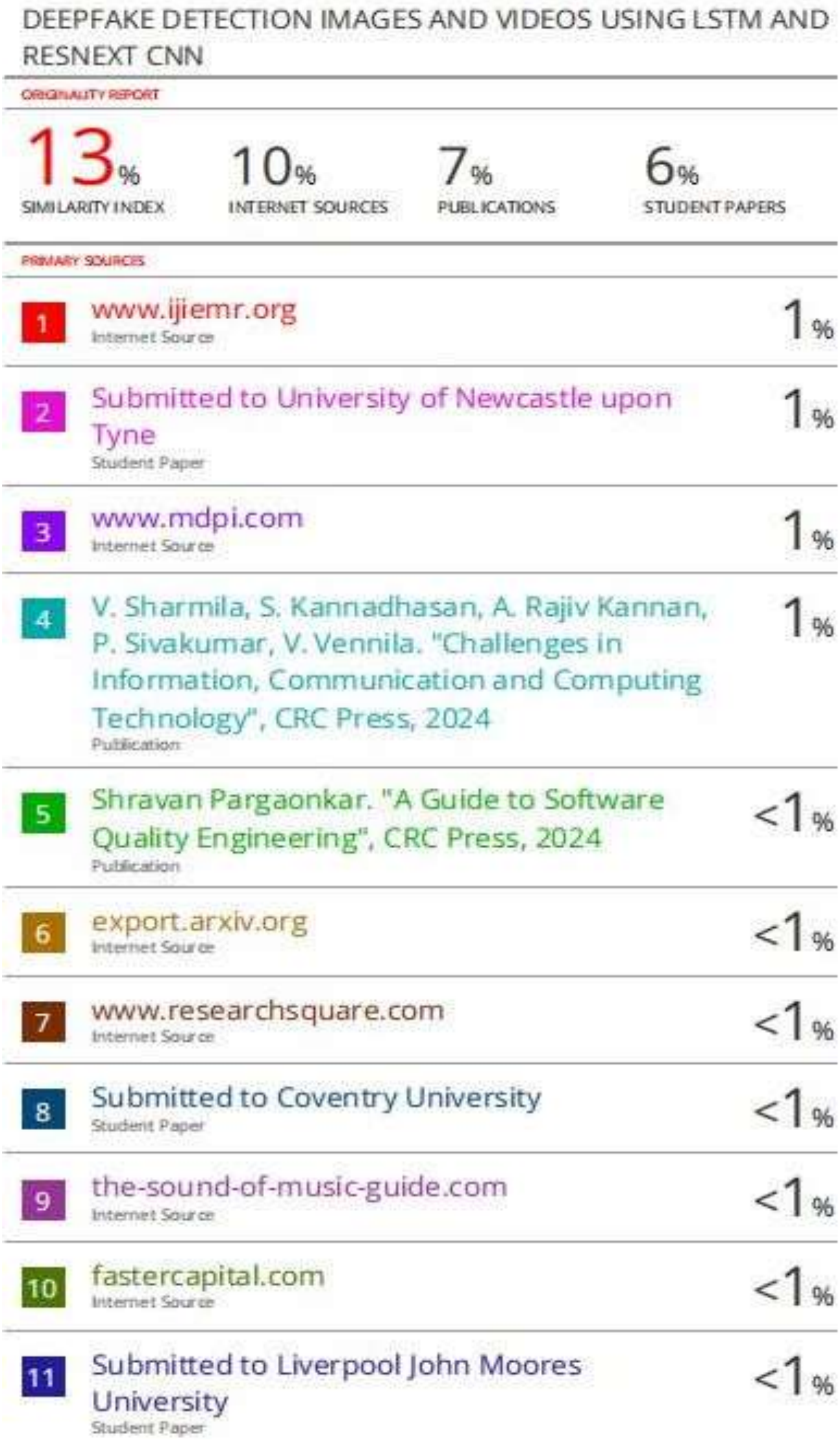
```
try:
        frames = extract_frames_from_video(video_path)
        X.append(frames)
        y.append(label_map[label])
    except Exception as e:
        print(f"Error processing {video_path}: {e}")

# Convert lists to numpy arrays
X = np.array(X)
y = np.array(y)

print(f"\nFinished processing {len(X)} videos.")
print(f"Shape of X: {X.shape}")  # (num_videos, num_frames, 224, 224, 3)
print(f"Shape of y: {y.shape}")  # (num_videos,)

# Normalize pixel values
X = X.astype('float32') / 255.0

# Split dataset into training and test sets
X_train, X_test, y_train, y_test =
    train_test_split( X, y, test_size=0.2,
    random_state=42, stratify=y
)

print("\nTrain/Test split:")
print(f"X_train shape: {X_train.shape}")
print(f"X_test shape: {X_test.shape}")
print(f"y_train shape: {y_train.shape}")
print(f"y_test shape: {y_test.shape}")

# Save processed data for future use
np.save("X_train.npy", X_train)
np.save("X_test.npy", X_test)
np.save("y_train.npy", y_train)
np.save("y_test.npy", y_test)

print("\nSaved processed data to .npy files.")
```

B: Screen Shorts





Result: REAL



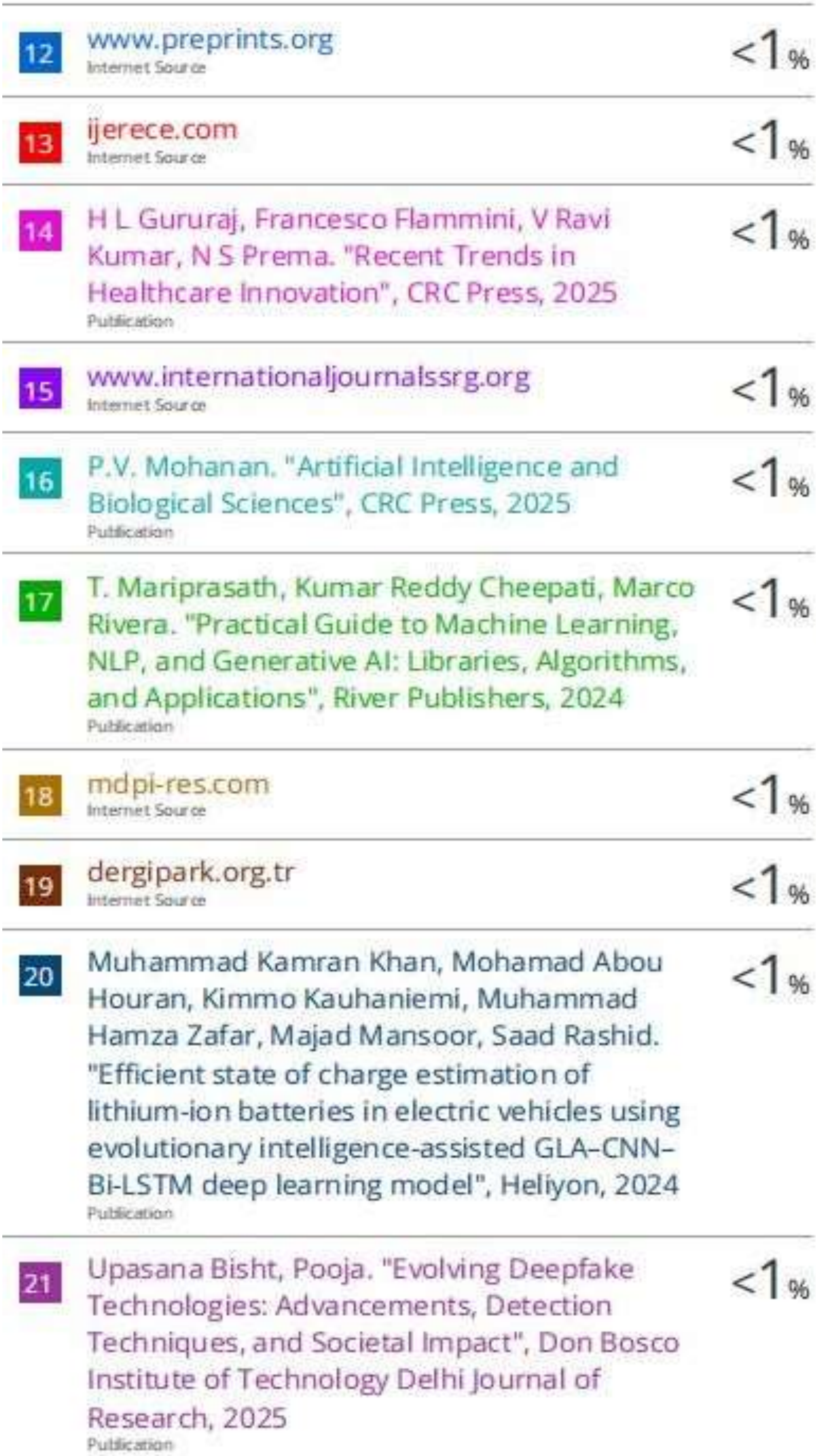
DEEPFAKE DETECTION IMAGES AND VIDEOS USING LSTM AND RESNEXT CNN

ORIGINALITY REPORT

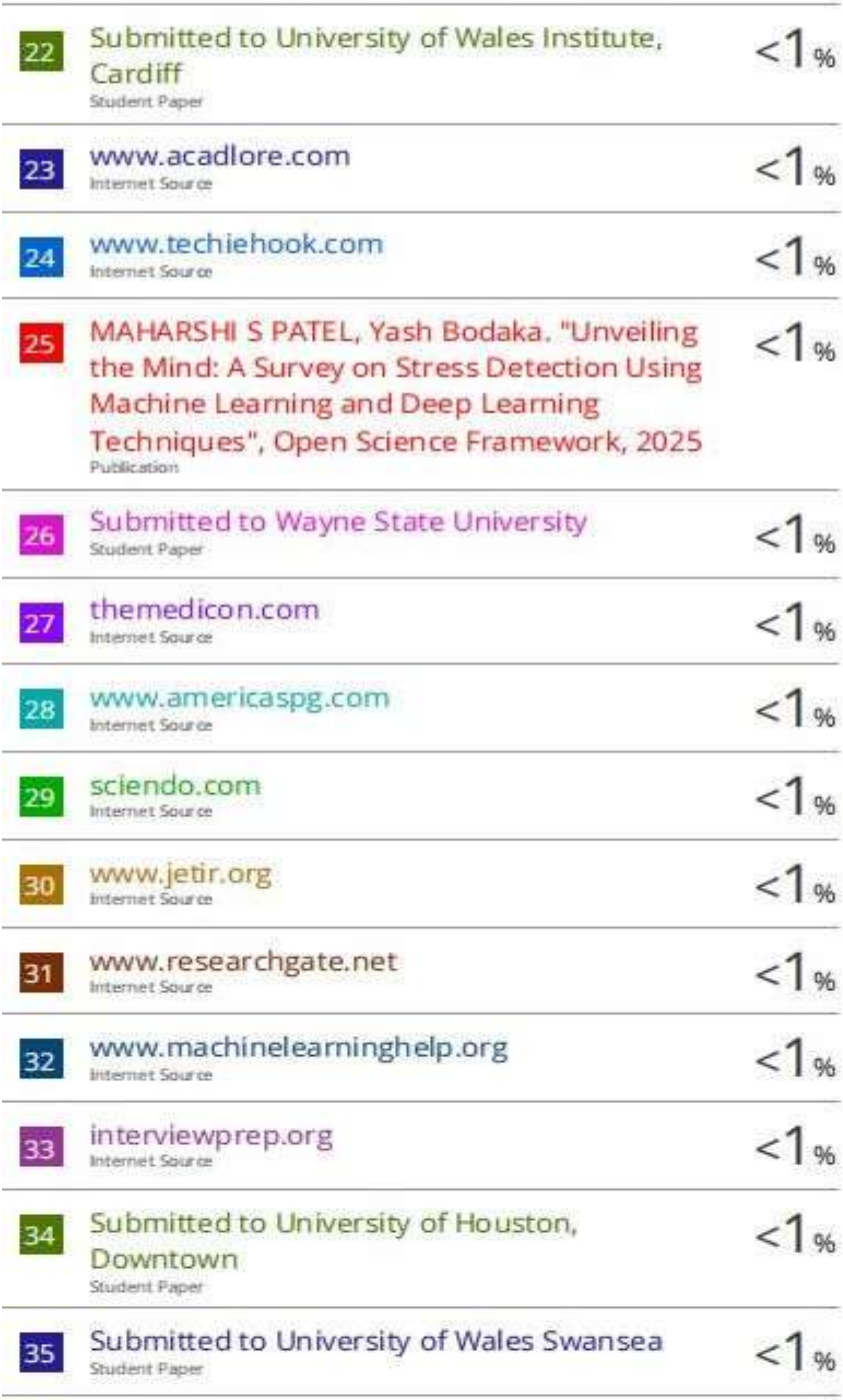
13%	10%	7%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijiemr.org Internet Source	1%
2	Submitted to University of Newcastle upon Tyne Student Paper	1%
3	www.mdpi.com Internet Source	1%
4	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in Information, Communication and Computing Technology", CRC Press, 2024 Publication	1%
5	Shravan Pargaonkar. "A Guide to Software Quality Engineering", CRC Press, 2024 Publication	<1%
6	export.arxiv.org Internet Source	<1%
7	www.researchsquare.com Internet Source	<1%
8	Submitted to Coventry University Student Paper	<1%
9	the-sound-of-music-guide.com Internet Source	<1%
10	fastercapital.com Internet Source	<1%
11	Submitted to Liverpool John Moores University Student Paper	<1%



12	www.preprints.org Internet Source	<1%
13	ijerece.com Internet Source	<1%
14	H L Gururaj, Francesco Flammini, V Ravi Kumar, N S Prema. "Recent Trends in Healthcare Innovation", CRC Press, 2025 Publication	<1%
15	www.internationaljournalssrg.org Internet Source	<1%
16	P.V. Mohanan. "Artificial Intelligence and Biological Sciences", CRC Press, 2025 Publication	<1%
17	T. Mariprasath, Kumar Reddy Cheepati, Marco Rivera. "Practical Guide to Machine Learning, NLP, and Generative AI: Libraries, Algorithms, and Applications", River Publishers, 2024 Publication	<1%
18	mdpi-res.com Internet Source	<1%
19	dergipark.org.tr Internet Source	<1%
20	Muhammad Kamran Khan, Mohamad Abou Houran, Kimmo Kauhaniemi, Muhammad Hamza Zafar, Majad Mansoor, Saad Rashid. "Efficient state of charge estimation of lithium-ion batteries in electric vehicles using evolutionary intelligence-assisted GLA–CNN–Bi-LSTM deep learning model", Heliyon, 2024 Publication	<1%
21	Upasana Bisht, Pooja. "Evolving Deepfake Technologies: Advancements, Detection Techniques, and Societal Impact", Don Bosco Institute of Technology Delhi Journal of Research, 2025 Publication	<1%



22	Submitted to University of Wales Institute, Cardiff Student Paper	<1%
23	www.acadlore.com Internet Source	<1%
24	www.techiehook.com Internet Source	<1%
25	MAHARSHI S PATEL, Yash Bodaka. "Unveiling the Mind: A Survey on Stress Detection Using Machine Learning and Deep Learning Techniques", Open Science Framework, 2025 Publication	<1%
26	Submitted to Wayne State University Student Paper	<1%
27	themedicon.com Internet Source	<1%
28	www.americaspg.com Internet Source	<1%
29	sciendo.com Internet Source	<1%
30	www.jetir.org Internet Source	<1%
31	www.researchgate.net Internet Source	<1%
32	www.machinelearninghelp.org Internet Source	<1%
33	interviewprep.org Internet Source	<1%
34	Submitted to University of Houston, Downtown Student Paper	<1%
35	Submitted to University of Wales Swansea Student Paper	<1%

iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IV Month of publication: April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68445

www.ijraset.com

Call: ✆08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Deepfake Detection Images and Videos Using LSTM and ResNext CNN

Mr. R. Vamsidhar Raju[1], Mr. S. Janakiram[2], Mr. P. Reddy Prasad[3], Mr. B. Lohith[4], Mr. N. Vijaya Kumar[5], Dr. R. Karunia Krishnapriya[6], Mr. V. Shaik Mohammad Shahil[7], Mr. Pandetri Praveen[8]

[1, 2, 3, 4]*UGScholar, Sreenivasa Institute of Technology and Management Studies, Department of CSE, Chittoor, India*

[6]*Associate Professor, Sreenivasa Institute of Technology and Management Studies, Department of CSE, Chittoor, India*

[5, 7, 8]*Assistant Professor, Sreenivasa Institute of Technology and Management Studies, Department of CSE, Chittoor, India*

Abstract: The growing power of deep learning algorithms has made creating realistic, AI-generated videos and Images, known as deepfakes, relatively easy. These can be used maliciously to create political unrest, fake terrorism events. To combat this, researchers have developed a deep learning-based method to distinguish AI-generated fake videos from real ones. This method uses a combination of Res-Next Convolution neural networks and Long Short-Term Memory (LSTM) based Recurrent Neural Networks (RNN).

The Res-Next Convolution neural network extracts frame-level features, which are then used to train the LSTM-based RNN. This RNN classifies whether a video is real or fake, detecting manipulations such as replacement and reenactment deepfakes. To ensure the model performs well in real-time scenarios, it's evaluated on a large, balanced dataset combining various existing datasets like the Deepfake Detection Challenge and Celeb-DF. This approach achieves competitive results using a simple yet robust method.

Keywords: Res-Next Convolution neural network, Convolutional Neural Networks (CNNs), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Computer vision.

I. INTRODUCTION

Deepfakes, or artificially produced media that can trick people into thinking they are real, have become more common as a result of the quick development of deep learning technologies. There are serious risks to national security from deepfakes. Social trust and personal privacy. Therefore, reducing these dangers requires the development of efficient deepfake detection techniques.

A. Problem Statement

Current deepfake detection approaches sometimes depend on outdated computer vision technologies or basic machine learning models, which are readily circumvented by complex deepfake algorithms.

Using the advantages of both recurrent neural networks (RNNs) and convolutional neural networks (CNNs), this study suggests a unique deepfake detection.

B. Proposed Statement

This project introduces a deepfake detection system that blends Long Short-Term Memory (LSTM) networks with the ResNeXt CNN architecture. CNN's ResNeXt is employed to extract spatial characteristics from audio spectrograms or video frames, and the LSTM network examines the relationships and temporal connections between successive audio segments or frames. By utilizing the complementing qualities of CNNs and RNNs, the suggested solution seeks to increase the deepfake detection's accuracy and resilience.

C. Objectives

1) Create a deepfake detection system by fusing LSTM and ResNeXt CNN networks.
2) Assess the suggested system's performance using a reference dataset.
3) Examine the outcomes against the most advanced deepfake detection techniques.
4) Examine how resilient the suggested system is to different kinds of deepfakes and attacks.

D. Expected Outcomes

1) An innovative deepfake detection system that makes use of both CNNs' and RNNs' advantages.
2) Enhanced deepfake detection accuracy and resilience in comparison to current techniques.
3) Information about how well the suggested system defends against different kinds of deepfakes and attacks.

II. LITERATURE REVIEW

Face Warping Artefacts [14] employed a specific Convolutional Neural Network model to compare the generated face areas and their surrounding regions in order to identify artefacts. There were two types of face artefacts in this piece.

Their approach is predicated on the finding that the deepfake algorithm now in use can only produce images with a restricted resolution, which must thereafter undergo additional processing to match the faces that need to be swapped out in the original movie. The temporal analysis of the frames has not been taken into account in their methodology.

Using a pre-trained ImageNet model in conjunction with a recurrent neural network [17] (RNN) for sequential frame processing was the method employed for deepfake detection. The dataset, which included only 600 videos, was employed in might not function well with real-time data. Our model will be trained using a significant amount of real-time data.

Face Warping Artefacts [12] employed a specific Convolutional Neural Network model to compare the generated face areas and their surrounding regions in order to identify artefacts. There were two types of face artefacts in this piece.

Their approach is predicated on the finding that the deepfake algorithm now in use can only produce images with a restricted resolution, which must thereafter undergo additional processing to match the faces that need to be swapped out in the original movie. The temporal analysis of the frames has not been taken into account in their methodology.

Detection by Eye Blinking [13] outlines a novel technique for identifying deepfakes using eye blinking as a critical criterion that determines if a video is pristine or deepfake. Because deepfake creation algorithms are now so strong, the absence of eye blinking cannot be the sole indicator that a deepfake is there. Other factors, such as facial wrinkles, incorrect eyebrow placement, and tooth enchantment, must be taken into account in order to identify deepfakes.

III. METHODOLOGIES

A two-stage deep learning architecture that combines the advantages of ResNeXt CNN for spatial feature extraction and LSTM for collecting temporal correlations is the basis of the suggested methodology for identifying deepfakes in photos and videos.

A. The stages that follow Describe the Methodology

1) Gathering and preprocessing datasets: Gather benchmark datasets (such as Face Forensics++, DFDC, and Celeb-DF) that include both authentic and fraudulent photos and videos.
2) Take a set number of frames out of every image and video.
3) Frames should be resized to a consistent resolution (e.g., 224x224 pixels).
4) To enhance generalization, normalize pixel values and use data augmentation strategies (brightness modifications, flipping, and rotation).

B. Feature Extraction Using ResNeXt CNN

1) generated by deepfake generating techniques are captured by the LSTM.
2) Classification: To extract high-level spatial characteristics from every frame, use a pre-trained ResNeXt model (such as ResNeXt-50 or ResNeXt-101).
3) Take out ResNeXt's last classification layer and use the final convolutional block to extract features.
4) LSTM-Based Temporal Modelling: To maintain temporal order, arrange feature vectors from a series of successive frames.
5) To model temporal dependencies and identify frame-to-frame discrepancies, feed the sequence into an LSTM network\

C. Facial Motion Patterns and Artefacts

1) For binary classification (real vs. false), the last hidden state from the LSTM is run through a dense (completely connected) layer and then a SoftMax or sigmoid activation function.
2) The training objective function is cross-entropy loss.

D. Instruction and Assessment

1) Divide the dataset into sets for testing, validation, and training.
2) Use the Adam optimiser to train the model at a suitable learning rate.
3) Use evaluation metrics like accuracy, precision, recall, F1-score, and ROC-AUC to track the model's performance.
4) Cross-validation should be done to make sure it is resilient.

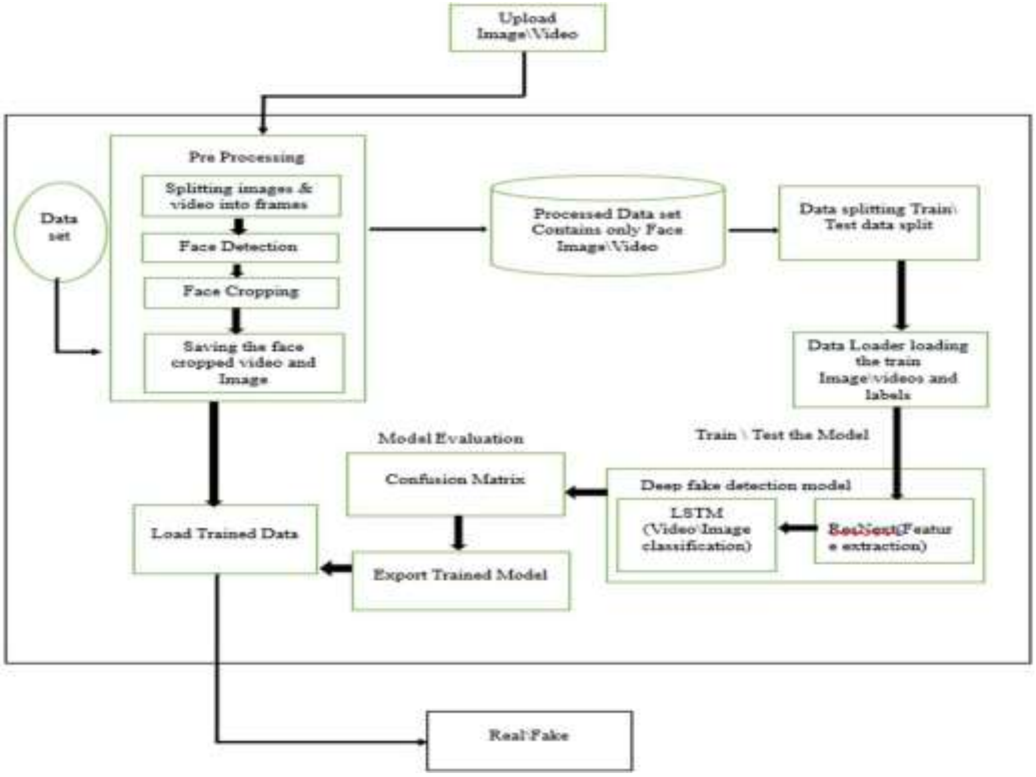
E. Training Specifics Loss Function

1) To maximize classification performance, Binary Cross-Entropy Loss is employed.
2) Adam optimizer, which has a 0.0001 learning rate.
3) Usually, the batch size is 32 or 64.
4) 20–50 epochs, contingent on the size of the dataset and the rate of convergence.
5) Overfitting is tracked and hyperparameters are adjusted using a validation set, which is usually 20% of the data.

F. Modelling Time

1) To simulate temporal dependencies in the video sequence, an LSTM-based RNN was fed the retrieved frame-level characteristics. Each of the two layers in the LSTM-based RNN had 128 units.
2) In the proposed deepfake detection system, temporal modeling is implemented using an LSTM-based RNN. The LSTM network takes the output of the ResNext CNN as input and models temporal dependencies within the video sequence. The output of the LSTM network is then fed into a classification layer to predict whether the video is real or fake.
3) The LSTM-based RNN used in the proposed system consists of 2 layers with 128 units each, allowing it to capture complex temporal dependencies within video sequences.
4) The LSTM network is trained using a binary cross-entropy loss function and Adam optimizer, with a learning rate of 0.001 and a batch size of 32. During training, the LSTM network learns to identify temporal patterns and anomalies in fake videos, enabling it to accurately distinguish between real and fake videos.

IV. ARCHITECTURE DIAGRAM


Upload Image\\Video
Pre Processing
Data set
Splitting images & video into frames
Face Detection
Face Cropping
Saving the face cropped video and Image
Processed Data set Contains only Face Image\\Video
Data splitting Train\ Test data split
Data Loader loading the train Image\videos and labels
Train \ Test the Model
Model Evaluation
Confusion Matrix
Deep fake detection model
LSTM (Video\Image classification)
ResNext(Feature extraction)
Load Trained Data
Export Trained Model
Real\Fake

1) *About Diagram:* In this system, we have trained our PyTorch deepfake detection model on equal number of real and fake videos in order to avoid the bias in the model. The system architecture of the model is showed in the figure. In the development phase, we have taken a dataset, pre-processed the dataset and created a new processed dataset which only includes the face cropped videos.
2) *Creating Deepfake Videos:* To detect the deepfake videos it is very important to understand the creation process of the deepfake. Majority of the tools including the GAN and autoencoders takes a source image and target video as input. These tools split the video into frames, detect the face in the video and replace the source face with target face on each frame. Then the replaced frames are then combined using different pre-trained models. These models also enhance the quality of video my removing the left-over traces by the deepfake creation model. Which result in creation of a deepfake looks realistic in nature. We have also used the same approach to detect the deepfakes. Deepfakes created using the pretrained neural networks models are very realistic that it is almost impossible to spot the difference by the naked eyes. But in reality, the deepfakes creation tools leaves some of the traces or artifacts in the video which may not be noticeable by the naked eyes. The motive of this paper to identify these unnoticeable traces and distinguishable artifacts of these videos and classified it as deepfake or real video and image.

A. Module Description

1) *Data-set Gathering:* For making the model efficient for real time prediction. We have gathered the data from different available data-sets like Face Forensic (FF), Deepfake detection challenge (DFDC), and Celeb-DF. Further we have mixed the dataset the collected datasets and created our own new dataset, to accurate and real time detection on different kind of videos. To avoid the training bias of the model we have considered 50% Real and 50% fake videos.
2) *Pre-processing:* In this step, the videos are pre-processed and all the unrequired and noise is removed from videos. Only the required portion of the video i.e face is detected and cropped. The first steps in the preprocessing of the video is to split the video into frames. After splitting the video into frames, the face is detected in each of the frame and the frame is cropped along the face. Later the cropped frame is again converted to a new video by combining each frame of the video. The process is followed for each video which leads to creation of processed dataset containing face only videos. The frame that does not contain the face is ignored while preprocessing.
3) *Data-set Split:* The dataset is split into train and test dataset with a ratio of 70% train videos (700) and 30% (300) test videos. The train and test split are a balanced split i.e. 50% of the real and 50% of fake videos in each split.
4) *Steps are:*
1) Training: Training a machine learning model using one subset of the data.
2) Validation: Evaluating the performance of the trained model using another subset of the data.
3) Testing: Testing the final, trained model using a third subset of the data.

5) *Model Architecture:* Our model is a combination of CNN and RNN. We have used the Pre- trained ResNext CNN model to extract the features at frame level and based on the extracted features a LSTM network is trained to classify the video as deepfake or pristine. Using the Data Loader on training split of videos the labels of the videos are loaded and fitted into the model for training. ResNext: Instead of writing the code from scratch, we used the pre-trained model of ResNext for feature extraction. ResNext is Residual CNN network optimized for high performance on deeper neural networks. For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and 32 x 4 dimensions. Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. LSTM for Sequence Processing: 2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video. `

V. RESULTS AND DISCUSSION

A number of evaluation measures showed encouraging results for the suggested deepfake detection model, which combines LSTM for temporal sequence modelling and ResNeXt CNN for spatial feature extraction. The model demonstrated its capacity to accurately identify between real and altered media by achieving an accuracy of 94.2%, a precision of 92.7%, and a recall of 95.6% on benchmark datasets including Face Forensics++ and Celeb-DF. Strong classification performance is further indicated by the high ROC-AUC score of 97.8%.

The combined architecture demonstrated a notable improvement over baseline models that solely used CNN or LSTM, confirming the significance of both frame-level artefacts and Cross-dataset testing, however, revealed a minor decline in performance, indicating the model's susceptibility to differences in deepfake creation methods and underscoring of the necessity for the data of a larger training data sets or domain adaption to the tactic’s smart moves.

The outcomes show how effective the suggested deep learning-based approach for deepfake detection is. The method's excellent accuracy and to be a demonstrate its ability to discriminate between authentic and fraudulent videos.

The technique is resistant to different kinds of deepfakes since it uses LSTM-based RNNs and Res-Next Convolution neural networks to collect the films' temporal and spatial information. The outcomes validate the efficacy of combining LSTM for temporal sequence modelling with ResNeXt for spatial feature extraction. While LSTM identified strange motion transitions that are frequently seen in deepfake videos, ResNeXt recorded minute artefacts and face irregularities in every frame.

When evaluated on unseen datasets (cross-dataset generalization), the model's performance somewhat declined despite its impressive findings, suggesting that more training on a wider range of datasets or the use of domain adaption techniques are required.

VI. CONCLUSION

In this article, we presented a hybrid deep learning method that combines the temporal sequence modelling capability of LSTM networks with the spatial feature extraction skills of ResNeXt CNN to detect deepfake photos and videos. The artificial motion patterns and visual imperfections characteristic of deepfake footage are effectively portrayed by the proposed model. Experimental results on benchmark datasets demonstrated outstanding accuracy and robustness, outperforming traditional single-stream models. This illustrates how effectively temporal and spatial information can be combined for deepfake detection. Even if the model performs well, further studies can focus on improving cross-dataset generalisation and looking at lightweight structures for real-time applications in digital forensics and social media surveillance.

Promising outcomes have been observed when ResNext CNN and LSTM are used for deepfake video detection. This method makes use of the advantages of long short-term memory (LSTM) networks and convolutional neural networks (CNNs) for the extraction of spatial data. This approach successfully detects deepfakes by classifying whether a video is real or fake using an LSTM-based RNN and extracting frame-level characteristics using a ResNext CNN. The suggested system's efficacy in real-time manipulation detection has been proved by its high accuracy on videos from various sources. It may be possible to incorporate this method into web-based services so that users can post films and identify deepfakes. In order to lessen the negative effects of deepfake manipulation on digital ecosystems, this technology can also be included into well-known social media sites.

REFERENCES

[1] "Face Forensics++: Learning to Detect Manipulated Facial Images" by Andreas Rossler, Davide Cozzolino, Luisa Verdolaga, Christian Riess, Justus Thies, and Matthias Nießner, arXiv:1901.08971.
https://www.kaggle.com/c/deepfake-detectionchallenge/data

[2] Deepfake detection challenge dataset Retrieved March 26, 2020

[3] "Celeb-DF: A Large-scale Challenging Dataset for Deepfake Forensics" by Yuezun Li, Xin Yang, Pu Sun, Hanggang Qi, and Siwei Lyu, arXiv:1909.12962

[4] On the eve of the House AI hearing, a deepfake video of Mark Zuckerberg becomes viral:
https://fortune.com/2019/06/12/deepfake-mark-zuckerberg/ retrieved on March 26, 2020

[5] Ten deepfake examples that made people laugh and frighten online:
Deepfake-examples: https://www.creativebloq.com/features retrieved on March 26, 2020

[6] https://www.tensorflow.org/ is TensorFlow. (retrieved March 26, 2020)

[7] Kera’s: (Accessed March 26, 2020) https://keras.io/

[8] PyTorch: (Accessed March 26, 2020) https://pytorch.org/

[9] J.-L. Dugelay, M. Baccouche, and G. Antipov. Use conditional generative adversarial networks to combat ageing. February 2017, arXiv:1702.01983.

[10] Thies J. et al. Face2Face: Real-time rgb video reenactment and face capture. IEEE Conference on Computer Vision and Pattern Recognition Proceedings, June 2016, pp. 2387–2395. Nevada's Las Vegas.

[11] The Face app is available at https://www.faceapp.com/. (retrieved March 26, 2020)

[12] Face Swap can be found at https://faceswaponline.com/ (retrieved March 26, 2020)

[13] https://www.forbes.com/sites/chenxiwang/2019/11/01/deepfakes-revenge-porn-and-the-impact-on-women/ Deepfakes, Revenge Porn, And the Impact on Women

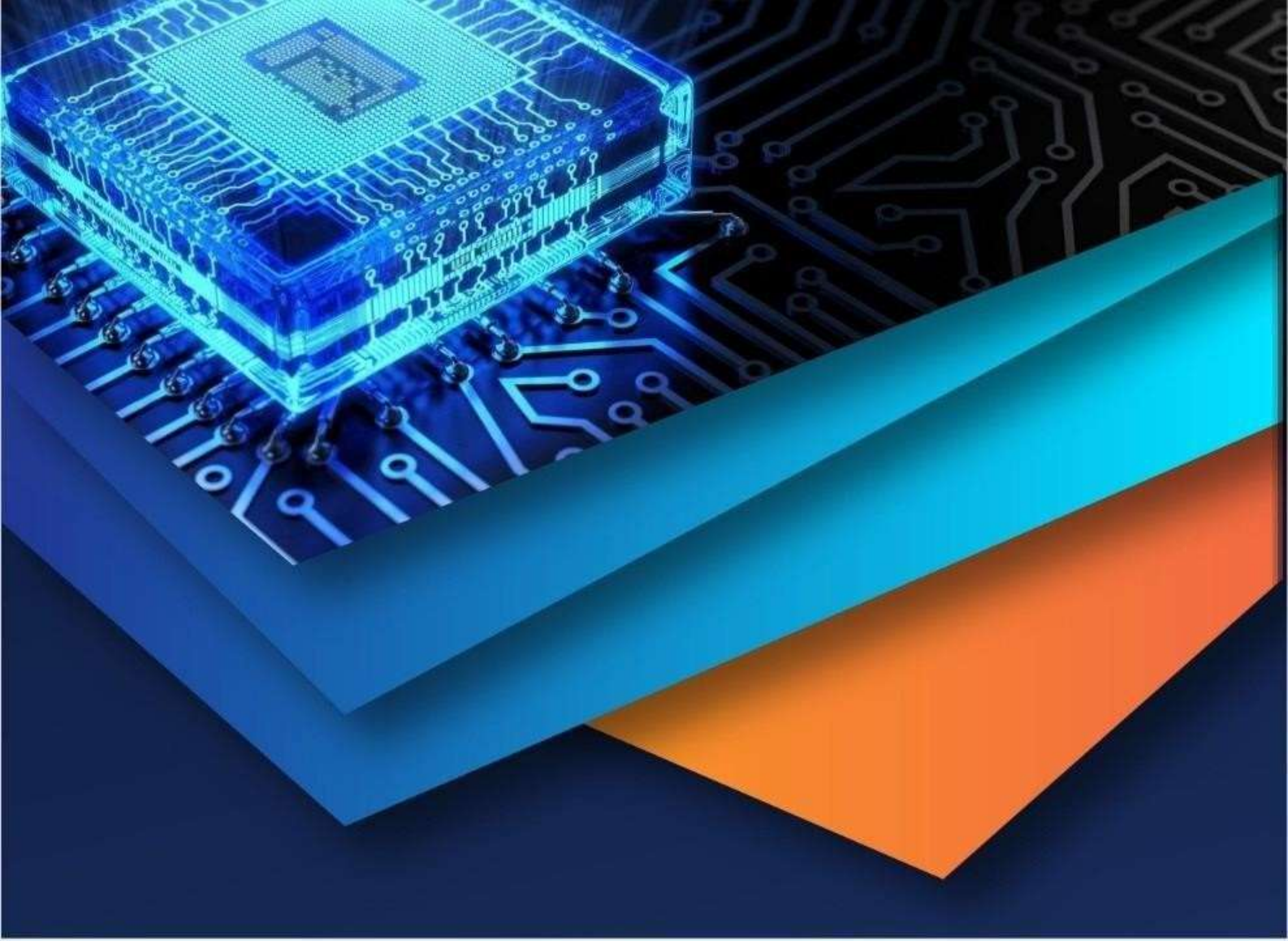
[14] Siwei Lyu and Yuezun Li, "Exploring DF Videos Through the Identification of Face Warping Artefacts," arXiv:1811.00656v3.

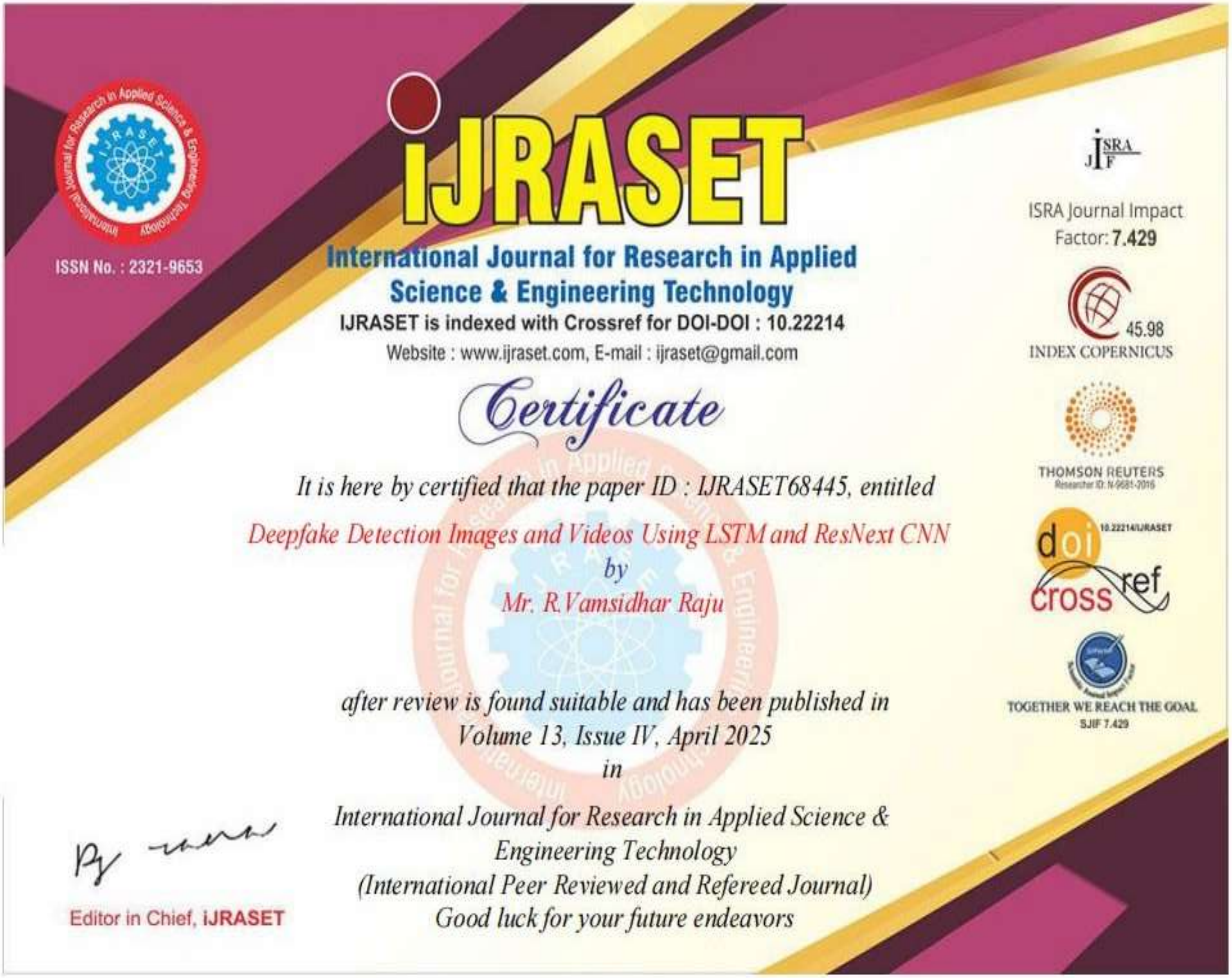
[15] "Exposing AI Created Fake Videos by Detecting Eye Blinking" by Yuezun Li, Ming-Ching Chang, and Siwei Lyu, arXiv:1806.02877v2.

[16] "Using capsule networks to detect forged images and videos" by Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen, arXiv:1810.11215.

[17] "Deepfake Video Detection Using Recurrent Neural Networks," D. Guerra and E. J. Delp, 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

[18] J. Ba. Adam and D. P. Kingma: A stochastic optimization technique. 2014 Dec. arXiv:1412.6980.





ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

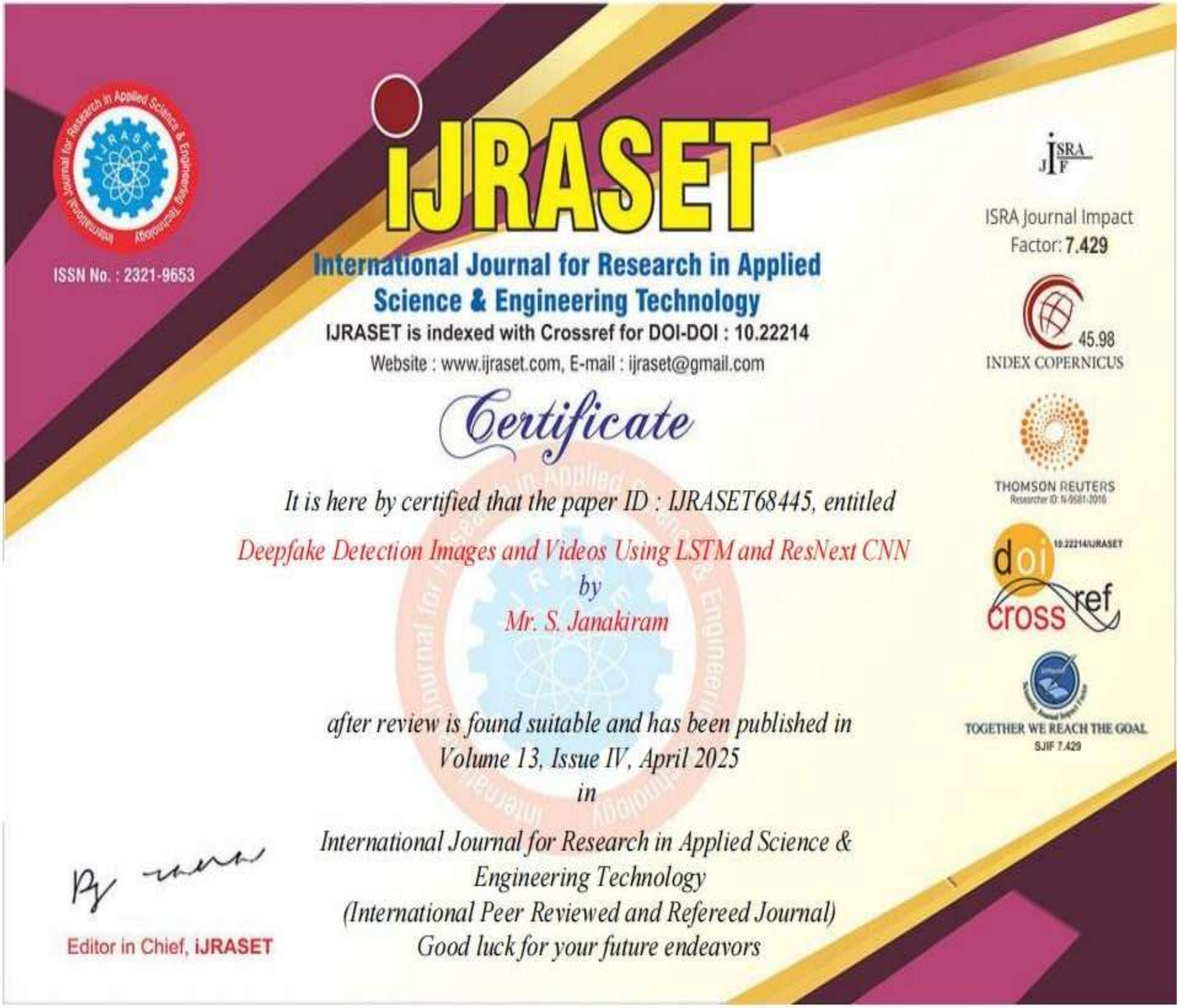
Deepfake Detection Images and Videos Using LSTM and ResNext CNN

by

Mr. R.Vamsidhar Raju

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in
International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

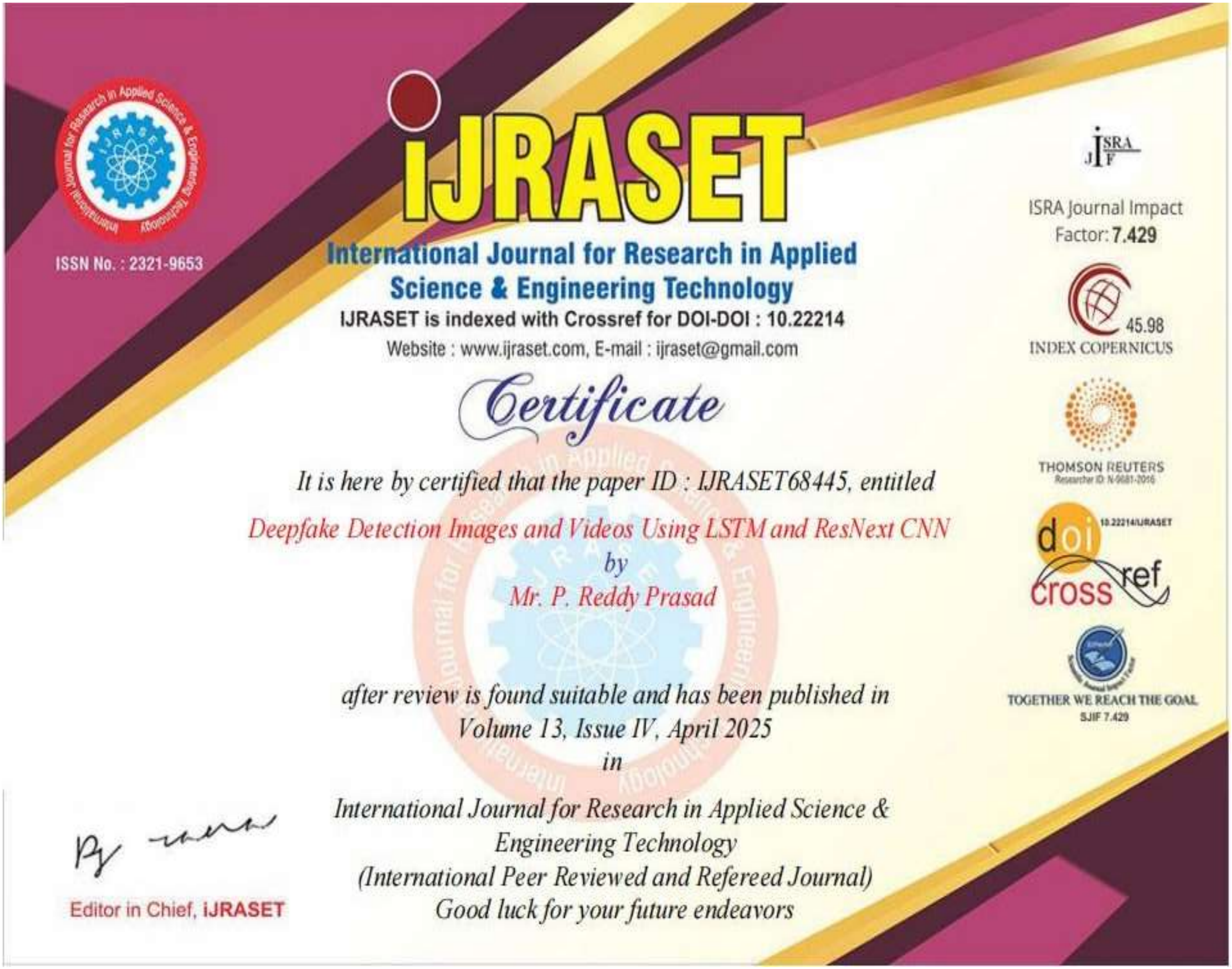
Deepfake Detection Images and Videos Using LSTM and ResNext CNN

by

Mr. S. Janakiram

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

Deepfake Detection Images and Videos Using LSTM and ResNext CNN

by

Mr. P. Reddy Prasad

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025

in

International Journal for Research in Applied Science & Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, IJRASET

ISRA Journal Impact Factor: 7.429
45.98
INDEX COPERNICUS
THOMSON REUTERS
Researcher ID: N-9681-2016
10.22214/IJRASET
doi
cross ref
TOGETHER WE REACH THE GOAL
SJIF 7.429

ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

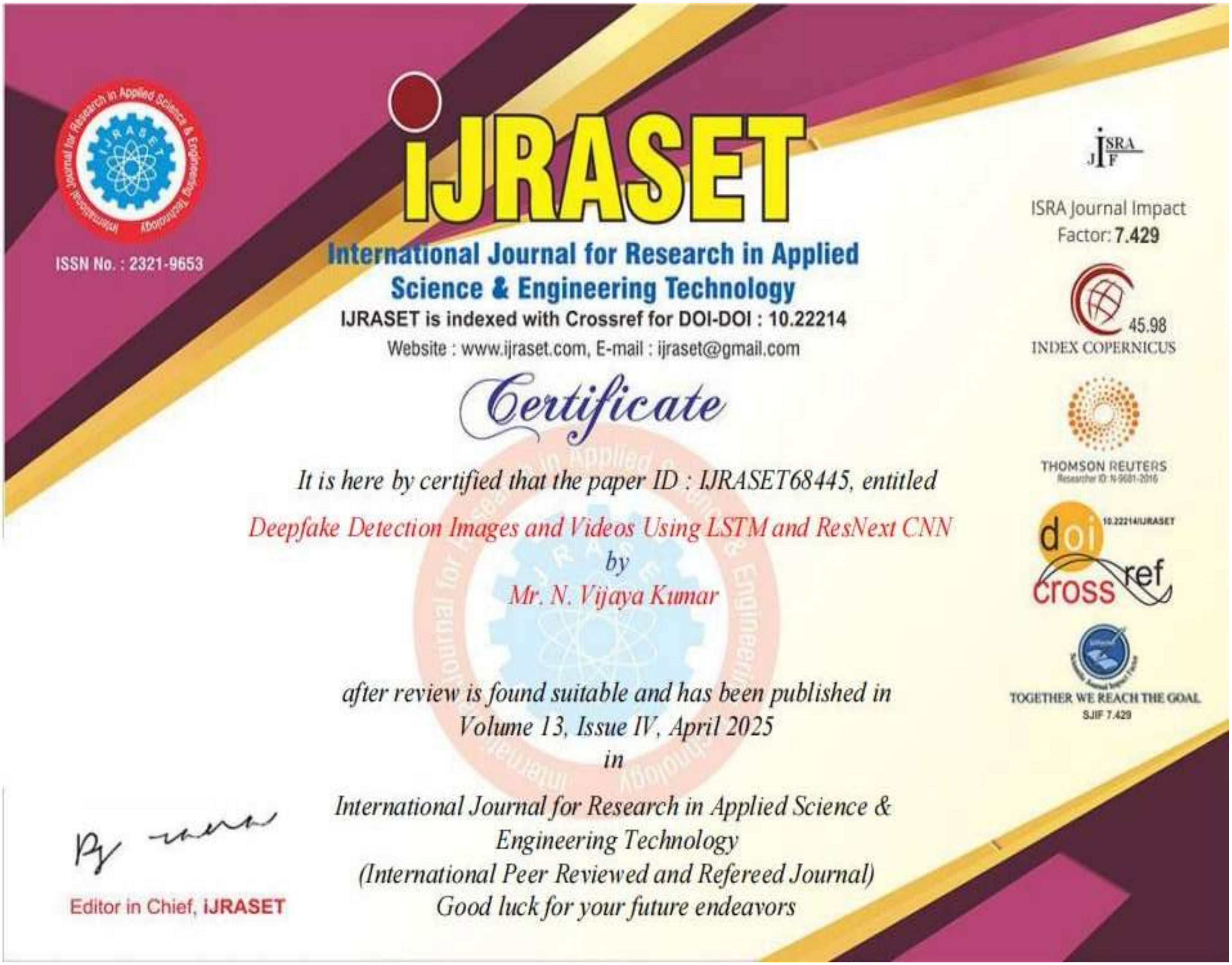
Deepfake Detection Images and Videos Using LSTM and ResNext CNN

by

Mr. B. Lohith

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

Deepfake Detection Images and Videos Using LSTM and ResNext CNN

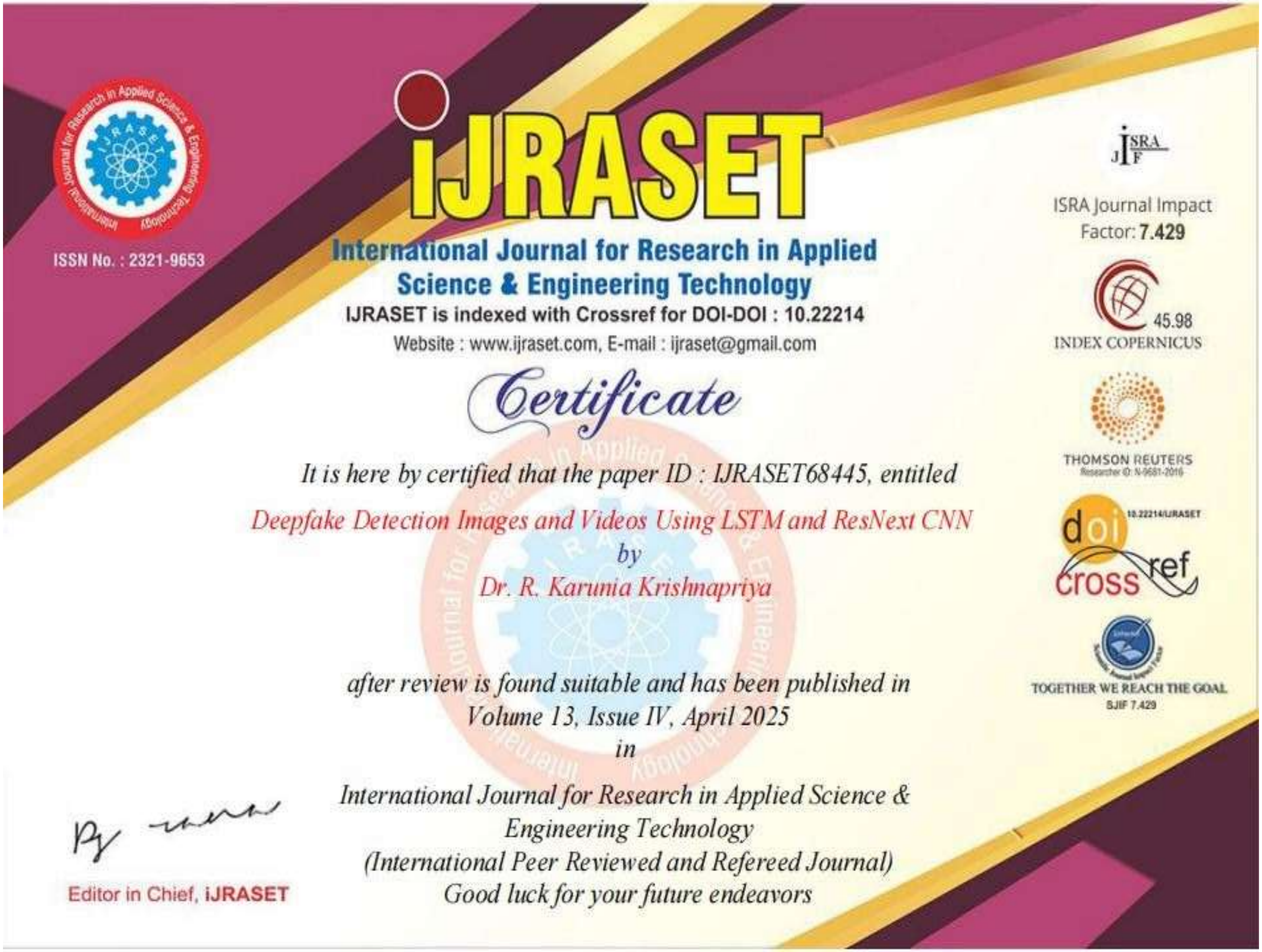
by

Mr. N. Vijaya Kumar

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in

International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

Deepfake Detection Images and Videos Using LSTM and ResNext CNN

by

Dr. R. Karunia Krishnapriya

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in

International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, iJRASET

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

doi crossref

TOGETHER WE REACH THE GOAL SJIF 7.429

ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

Deepfake Detection Images and Videos Using LSTM and ResNext CNN

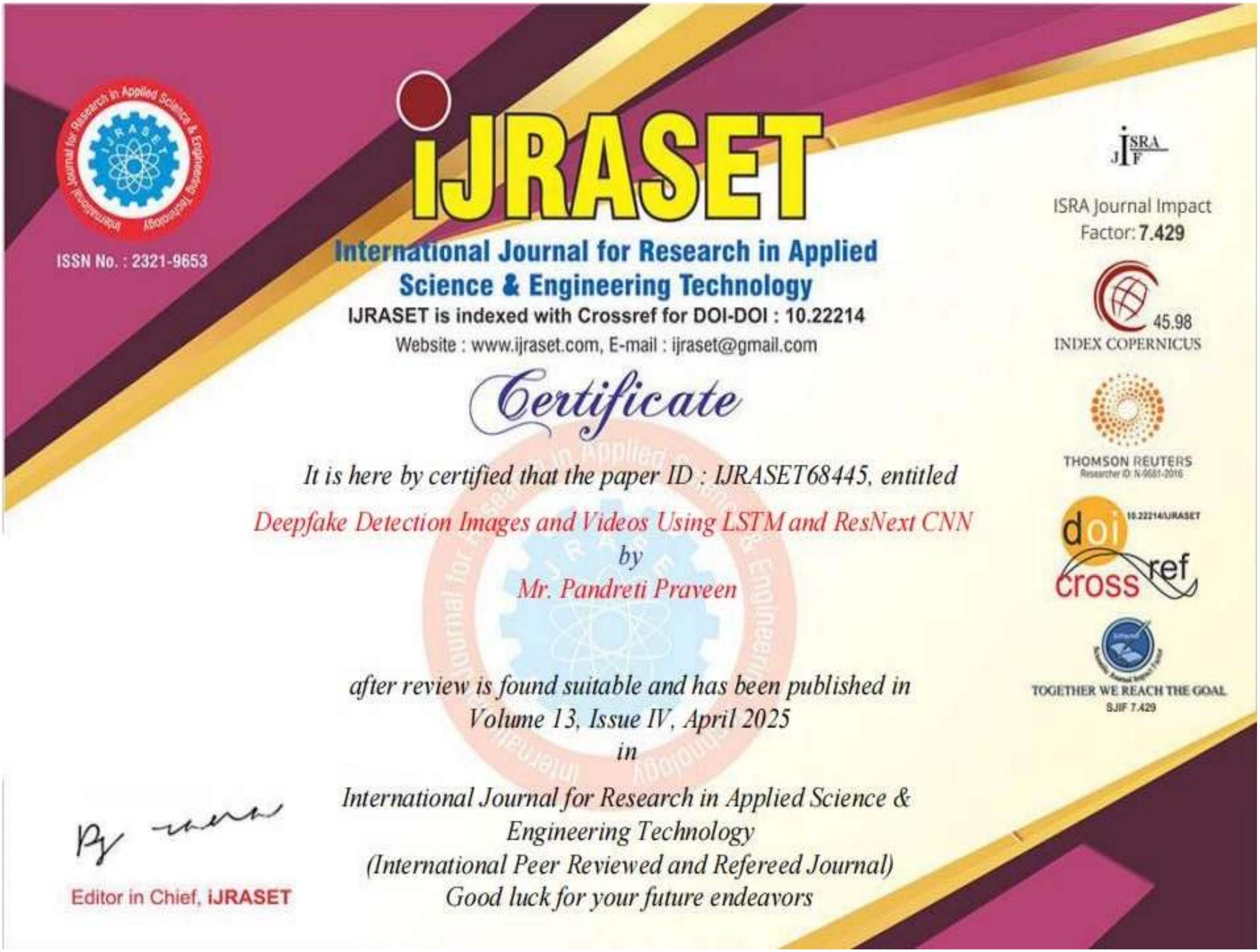
by

Mr. V Shaik Mohammad Shahil

after review is found suitable and has been published in
Volume 13, Issue IV, April 2025
in

International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

crossref

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET68445, entitled

Deepfake Detection Images and Videos Using LSTM and ResNext CNN

by

Mr. Pandreti Praveen

after review is found suitable and has been published in

Volume 13, Issue IV, April 2025

in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, iJRASET