DEEPFAKE DETECTION SYSTEM

PROJECT REPORT

Submitted by

JISTO KURIAKOSE (SNG20CS057)

MITHUN RAJU (SNG20CS065)

SREEJITH MOHAN (SNG20CS078)

VISHNUDAS T (SNG20CS083)

to

The APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Technology

In

Computer Science and Engineering

Department of Computer Science and Engineering

Sree Narayana Gurukulam College of Engineering

Kadayiruppu

682311

MAY 2024

DECLARATION

We undersigned hereby declare that the project report **" DEEPFAKE DETECTION SYSTEM "**, submitted for partial fulfilment of the requirements for the award of the degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a Bonafide work done by me under supervision of Prof**. (Dr.) Smitha Suresh** This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma, or similar title of any other University.

PLACE: KADAYIRUPPU

JISTO KURIAKOSE

MITHUN RAJU

SREEJITH MOHAN

VISHNUDAS T

B. TECH COMPUTER SCIENCE AND ENGINEERING 2020-2024

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING, KADAYIRUPU

(Affiliated to APJ Abdul Kalam Technological University & Approved by A.I.C.T.E)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2020-2024

CERTIFICATE

This is to certify that the main project report entitled, **“Deepfake Detection System”** is a bonafide work done by **Jisto Kuriakose, Mithun Raju, Sreejith Mohan, Vishnudas T** towards the partial fulfilment of the requirements for the award of the B. Tech degree in Computer Science and Engineering under KTU University.

Head of the Department	**Project Coordinator**	**Guided by**
Prof. Dr. Smitha Suresh	**Ms. Sindhu M. P**	**Ms. Archana P. S**
(HOD. CSE Dept.)	(Assoc. Prof. CSE Dept)	(Asst. Prof. CSE Dept)

Submitted for the University Evaluation on…………………………………………………….

University Register No……………………………………………………………………….

Internal Examiner External Examiner

ACKNOWLEDGEMENT

Dedicating this main project to the Almighty Universe whose abundant grace and mercy enabled its successful completion, I would like to express my profound gratitude to all the people who had inspired and motivated us to undertake this thesis work.

I wish to express my sincere thanks to our Head of the Department, Prof. Dr. Smitha Suresh, for providing me the opportunity to undertake this thesis work. I am deeply indebted to my guide Asst. Prof. Archana P. S in the Department of Computer Science and Engineering for providing me with valuable advice and guidance during the course of the thesis work. I also express my sincere thanks to Assoc. Prof. Sindhu M. P for their guidance and valuable support throughout the course of the thesis work. Then I would like to thank my friends who gave me suggestions and opinions to fulfil this thesis work.

Finally, I would like to express my gratitude to Sree Narayana Gurukulam College of Engineering for providing me with all the required facilities without which the successful completion of the project work would not have been possible.

ABSTRACT

The proliferation of deep learning algorithms and increasing computational power has facilitated the creation of indistinguishable human-synthesized videos, commonly known as deep fakes. The potential misuse of these realistic face-swapped deep fakes in scenarios such as political manipulation, fake terrorism events, revenge porn, and blackmail is a significant concern. In response, we present a novel deep learning-based method designed to effectively distinguish AI-generated fake videos from real ones.

Our approach utilizes a Res-Next convolutional neural network to extract frame-level features, which are then employed to train a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN). This architecture enables the classification of videos as either subject to manipulation or genuine. To ensure the practical applicability of our method in real-time scenarios, we evaluate its performance on a large, balanced dataset. This dataset is curated by amalgamating various existing datasets.

Furthermore, we demonstrate the effectiveness of our system through competitive results achieved via a simple and robust approach. By employing this method, we aim to leverage Artificial Intelligence (AI) to combat the adverse effects of AI-generated content.

COURSE OUTCOMES AND PROGRAM OUTCOMES

COURSE OUTCOMES: After the completion of the course, the student will be able to

CO1	Identify academic documents from the literature which are related to her/his areas of interest (Cognitive knowledge level: **Apply**).
CO2	Read and apprehend an academic document from the literature which is related to her/ his areas of interest (Cognitive knowledge level: **Analyze**).
CO3	Prepare a presentation about an academic document (Cognitive knowledge level: **Create**).
CO4	Give a presentation about an academic document (Cognitive knowledge level: **Apply**).
CO5	Prepare a technical report (Cognitive knowledge level: **Create**).

Program outcomes
Engineering Graduates will be able to:
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 analyse 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 contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO7. Environment and sustainability: Understand the impact of 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 engineering practice.

PO9. Individual and teamwork: 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.

PROGRAM SPECIFIC OUTCOMES (PSO's)

PSO1: Finding Innovative Solutions: Shall enhance the employability skills by finding innovative solutions for challenges and problems in various domains of CS.

PSO2: Software Development: Shall apply the acquired knowledge to develop software solutions and innovative mobile applications for various problems.

CO PO PSO MAPPING

	PO1(Engineering Knowledge)	PO2(Problem Analysis)	PO3(Design/Development of	PO4(Conduct Investigations of	PO5(Modern Tool Usage)	PO6(The Engineer and Society)	PO7(Environment and	PO8(Ethics)	PO9(Individual and Teamwork)	PO10(Communication)	PO11(Project Management and	PO12(Life-long Learning)	PSO1(Finding Innovative	PSO2(Software Development)
CO1	2	3	2	2	2	1	2	2	2	2	2	2	3	2
CO2	2	2		3	3	2	2	2	3	2	2	2		3
CO3	3	3	2	2	3	1		2	2	3	3	2	3	2
CO4	2	2	2	2		2	2	2	3	3	2	2	2	2
CO5	3	3	2	3	2	2	2	3	3	3	2	3	2	2
AVERAGE	2.4	2.6	2	2.4	2.5	1.2	1	2.2	1.5	2.6	1.4	2.2	2.4	2.4

PO PSO ATTAINMENT AND JUSTIFICATION

PO	Attained point (0/1/2/3)	Justification
PO1	3	We applied advanced deep learning algorithms, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), exemplifying a direct application of engineering knowledge. Our incorporation of ResNet50 for feature extraction and LSTM for sequence processing underscores a robust utilization of established engineering principles.
PO2	3	By delving deeply into the intricacies of identifying deepfake videos, we conducted a comprehensive analysis of the engineering problem. The literature review offered valuable insights into various methodologies, strengths, and limitations of existing deepfake detection systems, substantiating our meticulous problem analysis.
PO3	3	We formulated an advanced deepfake detection system, emphasizing key design elements such as face region focus, sequence processing, and the utilization of specific neural network architectures. Our structured approach to frame processing, feature extraction, sequence processing, and prediction aligns harmoniously with the principles of designing solutions for intricate engineering challenges.
PO4	2	Our initiative involved the systematic use of research-based knowledge, experimentation, and data analysis to devise a sophisticated deepfake detection system. The thorough exploration of various research papers and methodologies contributed to a well-informed and systematic investigation

PO5	3	We extensively employed modern engineering tools, including cutting-edge deep learning frameworks and neural network architectures. The amalgamation of ResNet50 CNN and LSTM for feature extraction and sequence processing exemplifies our adept proficiency in modern tool usage.
PO6	2	Acknowledging and addressing the societal implications of deepfake technology, we aimed to contribute to the prevention of misinformation and disinformation. Ethical considerations and responsible development were explicitly acknowledged, aligning with the ethical responsibilities of engineers to society.
PO7	0	
PO8	3	Our initiative incorporates ethical principles throughout the development process. We recognize the potential societal impacts of deepfake technology and have emphasized responsible development and ethical considerations. This aligns directly with PO8, demonstrating a commitment to professional ethics and responsibilities
PO9	2	While primarily an individual initiative, we acknowledged the interdisciplinary nature of deepfake detection and the necessity for collaboration with researchers from diverse domains. Furthermore, our initiative contributes to enhancing employability skills, aligning seamlessly with PSO1.
PO10	3	Through the presentation of a comprehensive literature review, our proposed innovative methodology, and the ultimate project report, effective communication was a cornerstone of our initiative. Proficient communication skills are essential for conveying intricate engineering concepts to both the engineering community and society at large

PO11	2	We demonstrated a clear understanding of project management principles by presenting a detailed schedule (Gantt chart) for different phases. This aligns with PO11 regarding project management competencies.
PO12	2	Our project reflects the spirit of lifelong learning by navigating the rapidly evolving landscape of deepfake technology. The comprehensive literature review, exploration of diverse methodologies, and adaptation to emerging challenges showcase our commitment to continuous learning in the broad context of technological change.

PSO1	2	Finding Innovative Solutions: Our initiative actively sought and proposed innovative solutions to address the challenges posed by deepfake technology, aligning perfectly with PSO1.
PSO2	2	The proposed deepfake detection system involved the development of software solutions utilizing advanced deep learning techniques, showcasing a tangible application of knowledge in software development.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVATION

Sl. No.	Title	Expansion
1	CNN	Convolutional Neural Network
2	RNN	Recurrent Neural Network
3	LSTM	Long-Short Term Memory
4	ResNet50	Residual Network
5	DNN	Deep Neural Network

CHAPTER 1

INTRODUCTION

In today's era of burgeoning social media platforms, the rise of deepfake technology poses a significant threat. Deepfakes, realistic face-swapped videos created using advanced AI algorithms, have the potential to disrupt political landscapes, fabricate terrorism events, propagate revenge porn, and facilitate blackmail schemes. These manipulative videos, often indistinguishable from authentic footage, can cause widespread confusion and harm. For instance, the circulation of purported nude videos featuring celebrities like Brad Pitt and Angelina Jolie underscores the urgency of developing robust methods to detect and combat deepfakes.

Deepfakes are typically created using tools such as FaceApp and Face Swap, which utilize pre-trained neural networks like Generative Adversarial Networks (GANs) or Autoencoders. These tools enable the seamless superimposition of target images onto source videos, resulting in deceptively realistic deepfake videos. To tackle this challenge, our approach employs a Long Short-Term Memory (LSTM) based artificial neural network for sequential temporal analysis of video frames, complemented by a pre-trained Res-Next Convolutional Neural Network (CNN) for extracting frame-level features.

Our methodology capitalizes on the inherent limitations of deepfake creation tools, which often leave discernible artifacts in the manipulated videos. While imperceptible to the human eye, these artifacts can be detected by trained neural networks. By leveraging these distinctive artifacts, we train our system to effectively discern between genuine and deepfake videos.

To ensure the practical applicability of our method in real-world scenarios, we extensively train and evaluate our model on a diverse dataset comprising deepfake videos from sources like FaceForensic++, the Deepfake Detection Challenge, and Celeb-DF. Additionally, we validate our model's performance against real-time data sourced from platforms like YouTube, striving to achieve competitive results under real-world conditions.

In summary, our project aims to provide a comprehensive solution to the deepfake menace by leveraging advanced AI techniques to detect and mitigate the proliferation of manipulated video content across social media platforms and online channels.

1.1 OVERVIEW

The advancement of mobile camera technology coupled with the widespread use of social media platforms has revolutionized the creation and sharing of digital videos. Deep learning has enabled the development of remarkable technologies, such as modern generative models capable of synthesizing highly realistic images, speech, music, and videos. These models have found applications in various fields, ranging from enhancing accessibility through text-to-speech systems to generating training data for medical imaging.

However, along with these advancements come new challenges. Deep generative models have enabled the creation of so-called "deep fakes," which are manipulated video and audio clips. Since their emergence in late 2017, numerous open-source deep fake generation methods and tools have become available, resulting in a proliferation of synthesized media clips. While some of these may be created for entertainment purposes, others pose significant risks to individuals and society. The increasing availability of editing tools and the demand for domain expertise have contributed to the rise in both the quantity and realism of fake videos.

The dissemination of deep fakes on social media platforms has become commonplace, leading to issues such as spamming and the dissemination of misinformation. For instance, envision a scenario where a deep fake of a political leader declares war against neighbouring countries or a deep fake featuring a well-known celebrity insulting their fans. Such instances could have severe consequences, including inciting panic and misleading the public.

To address this growing problem, deep fake detection is crucial. Therefore, we present a novel deep learning-based method designed to effectively distinguish AI-generated fake videos (Deep Fake Videos) from real ones. Developing technology capable of detecting and preventing the spread of deep fakes is of utmost importance to safeguard individuals and prevent the propagation of false information on the internet.

1.2 PROBLEM STATEMENT

While convincing manipulations of digital images and videos have existed for decades, recent advancements in deep learning have significantly enhanced the realism of fake content and its accessibility. This has led to the emergence of AI-synthesized media, commonly known as deep fakes. Creating deep fakes using AI tools has become increasingly simple. However, detecting these deep fakes poses a major challenge. Throughout history, deep fakes have been used for various malicious purposes, including creating political tension, fabricating terrorism events, spreading revenge porn, and blackmailing individuals. Consequently, detecting and preventing the dissemination of deep fakes through social media platforms has become imperative.

To address this challenge, we have developed a method utilizing LSTM-based artificial neural networks to detect deep fakes. This approach represents a significant step forward in combating the proliferation of manipulated media online.

1.3 OBJECTIVE

Our project is dedicated to uncovering the obscured truth behind deep fakes, aiming to mitigate the widespread abuse and misinformation that plagues the online world. By developing a robust system, our project endeavours to differentiate between authentic content and manipulated deep fakes, ultimately reducing the potential harm inflicted upon unsuspecting individuals on the internet. Through advanced AI techniques, our system will classify videos with precision, accurately determining whether they are genuine or deepfake. Furthermore, we prioritize user accessibility by providing an intuitive and user-friendly interface, allowing individuals to easily upload videos for analysis and receive prompt feedback on their authenticity.

1.4 SCOPE

While numerous tools exist for creating deep fakes, the availability of effective tools for detecting them is scarce. Our proposed approach to deep fake detection stands as a significant contribution to combating the proliferation of manipulated media on the internet. By offering a web-based platform, users will have the means to upload videos and determine whether they are authentic or deep fakes. This project holds the potential for scalability, with possibilities ranging from expanding the web-based platform to developing a browser plugin for seamless automatic deep fake detection. Additionally, prominent applications such as WhatsApp and Facebook could integrate this technology to enable pre-detection of deep fakes before sharing content with other users. The software description includes details such as input size, input validation, input dependency, and major inputs and outputs, providing a comprehensive overview of its functionality and capabilities without delving into implementation specifics.

CHAPTER 2

LITERATURE REVIEW

[1] Deepfake Detection through Deep Learning by Deng Pan, Lixian Sun, Rui Wang, Xingjian Zhang, Richard O. Sinnott (2020):

In their pursuit to identify videos generated using deepfake technology, the authors employ a deep learning model, Xception, to detect fake videos generated by mainstream deepfake methods with high accuracy. However, they acknowledge a potential limitation as the model might not perform as effectively on fake videos generated using different approaches. The study underscores the challenge of adapting video input to deep learning models designed for images.

[2] Deepfake Video Detection using Neural Networks by Abhijit Jadhav, Abhishek Patange, Jay Patel, Hitendra Patil, Manjushri Mahajan (2020):

This work presents a web-based platform for users to upload videos and classify them as fake or real, addressing various types of deepfakes. The approach involves splitting the video into frames, followed by face cropping. Notably, the model processes cropped frames directly for detection. A notable drawback is the inability to detect audio deepfakes, emphasizing a potential limitation in the overall detection capabilities.

[3] Deep fake detection and classification using error-level analysis and deep learning by Rimsha Rafque, Rahma Gantassi, Rashid Amin, Jaroslav Frnda, Aida Mustapha, Asma Hassan Alshehri (2023):

The authors propose a novel deep fake detection and classification method combining Error Level Analysis (ELA) and deep learning. This methodology involves resizing images to CNN's input layer and performing ELA to identify digital manipulation at the pixel level. The proposed technique achieves high accuracy, especially with ResNet18 and KNN. However, a challenge highlighted is the need for regular updates to the dataset, pointing to the dynamic nature of deepfake generation techniques.

[4] DeepFakeDG: A Deep Learning Approach for Deep Fake Detection and Generation by Zeina Aymana, Natalie Sherifa, Mariam Mohamed, Mohamed Hazema, Diaa Salama (2023):

This paper introduces a web application for deepfake detection and generation. The detection side involves face extraction, training on a dataset, and employing a machine learning classifier. On the generation side, users upload two videos, and the model extracts frames for merging faces with enhancements. The study demonstrates slightly better accuracy with CNN compared to VGG. The authors intend to enhance dataset generalization and augmentation techniques to improve the system's ability to detect user-inserted data, emphasizing the importance of regular dataset updates.

[5] Video Manipulation Detection Using Stream Descriptors by Aleksander Dash, Nolan Handali (2019):

This study combines Convolutional Neural Networks (CNNs) for facial feature extraction with Recurrent Neural Networks (RNNs) to capture temporal patterns in video frames for deepfake detection. The model integrates facial detection systems to focus exclusively on faces within frames, achieving a promising 72.5% accuracy on the validation set. While demonstrating potential in deepfake detection, the study acknowledges challenges such as overfitting and computational cost associated with temporal information integration and adaptive face detection.

[6] Video Manipulation Detection and Localization using Deep Learning by Hemal Mamtora, Kevin Doshi, Shreya Gokhale, Surekha Dholay, Chandrashekhar Gajbhiye (2020):

This work focuses on developing a deep learning-based solution for detecting and localizing spatiotemporal manipulation in digital videos using an LSTM-based model. The proposed model aims to perform both temporal and spatial localization of forgery. The study uses the REWIND dataset for experiments, with evaluations conducted at pixel-level for spatial localization, frame-level for temporal localization, and video-level for forgery detection accuracy assessment. The model's effectiveness is highlighted, but challenges include data dependency and computational complexity.

[7] Deep Learning for Deepfakes Creation and Detection by Thanh Thi Nguyen, Tien Dung Nguyen, Cuong M. Nguyen, Saeid Nahavandi (2022):

This paper conducts a comprehensive review of existing literature on both deepfake creation algorithms and state-of-the-art methods for detecting deepfakes. The authors meticulously analyse various deepfake generation techniques and explore the associated threats to privacy, democracy, and national security. The study also delves into the current landscape of deepfake detection methods, examining their efficacy and limitations. Serving as a roadmap for the AI research community, the paper highlights challenges, trends, and future directions in the realm of deepfakes while emphasizing the importance of considering potential misuse and societal impacts for a holistic perspective.

[8] Deepfake Detection Algorithm Based on Dual-Branch Data Augmentation and Modified Attention Mechanism by Da Wan, Manchun Cai, Shufan Peng, Wenkai Qin, and Lanting Li (2023):

This paper introduces an innovative deepfake detection algorithm utilizing a dual-branch data augmentation framework and a modified attention mechanism within ResNet50. The approach combines traditional random sampling augmentation with adversarial samples during data preprocessing, enhancing image diversity. The model addresses attention distribution issues and allocates increased weight to forged traces in multi-scale feature maps. The study achieves a more robust model with higher accuracy, especially in unpredictable data distributions. However, the increased complexity and computational requirements during both training and inference pose challenges. Validation on standard and corrupted datasets demonstrates significant improvements in effectiveness and robustness compared to mainstream methods.

[9] DeepFake Detection Methods by Siwei Lyu (2022):

Siwei Lyu's paper categorizes deepfake detection methods into signal feature-based, physical/physiological-based, and data-driven approaches. Signal feature-based methods focus on anomalies in the generation process, being sensitive to disturbances. Physical/physiological-based methods expose deepfakes through violations of physics or human physiology, relying on robust Computer Vision algorithms. Data-driven methods, utilizing DNNs, achieve high performance with large datasets, but success depends on the quality and diversity of training data and model design. Each category has distinct strengths and limitations, influencing the overall effectiveness of deepfake detection. The paper emphasizes the importance of training on specific datasets for model relevance but notes the challenge of generalization for unseen models.

[10] DeepFake Detection by Analyzing Convolutional Traces by Luca Guarnera, Oliver Giudice, Sebastiano Battiato (2020):

This study proposes a unique approach for deepfake detection by analyzing convolutional traces, specifically focusing on local pixel correlation in deepfakes. The method suggests that this correlation depends on all layers, notably Transpose Convolution layers in GAN. Using unsupervised learning, the EM algorithm forms clusters to reveal hidden dataset structure, creating a spatially capturing model for pixel correlations. However, the operational dependence on the direct connection between local pixel correlation in deepfakes and the operations executed by Transpose Convolution layers is highlighted as a consideration, along with the model's capacity to generalize across diverse datasets.

[11] An Extensive Analysis of Deep Learning-based Deepfake Video Detection by D. Myvizhi, J. C. Miraclin Joyce Pamila (2022):

This comprehensive analysis of deep learning-based deepfake video detection involves diverse dataset selection, standardization, and preprocessing. State-of-the-art deep learning models, including 3D CNNs, are employed with a structured training/validation/testing split and data augmentation. The methodology focuses on feature extraction targeting temporal patterns and unique artifacts in deepfake generation. Fine-tuning based on performance evaluations enhances detection accuracy, and post-processing refines results while addressing ethical considerations. Despite the benefits of automating deepfake detection, the study acknowledges the challenge of balancing false positives and false negatives, impacting overall accuracy.

[12] Deepfake Video Detection Using Convolutional Neural Network by Aarti Karandikar, Vedita Deshpande, Sanjana Singh, Sayali Nagbhidkar, Saurabh Agrawal (2020):

This paper proposes a method for deepfake video detection using a Convolutional Neural Network (CNN). The approach involves training a video frame classifier with face extraction, alignment, and a fine-tuned VGG-16 convolutional model. The classifier integrates batch normalization, dropout, and a custom two-node dense layer for real and fake classes. Adam Optimizer enhances learning, and transfer learning evaluates simple features, detecting anomalies introduced during fake image creation. The study highlights the high accuracy of CNNs in detecting deepfake videos due to their ability to learn intricate patterns and features in image data, but notes the potential limitation of reduced generalization and detection performance with limited or biased training data.

[13] Deepfake Video Detection Using Recurrent Neural Networks by David Guera, Edward J. Delp (2019):

This paper introduces a temporal-aware system for automatically detecting deepfake videos using a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks

(RNNs). The study employs a CNN to extract frame-level features, which are then used to train an RNN that classifies if a video has been subject to manipulation. The proposed simple convolutional LSTM structure achieves fast and highly accurate detection of fake videos, especially those manipulated using popular deepfake generating techniques. However, the challenge lies in identifying videos manipulated using unseen and unorthodox methods.

[14] Deepfake Examiner by Hafas Ilyas, Aun Irtaza, Ali Javed, Khalid Malik (2022):

"Deepfake Examiner" proposes an end-to-end deep learning model for deepfake video detection, introducing a novel hybrid deep learning framework, Inception ResNet-BILSTM. The model aims to be robust to different ethnicities and varied illumination conditions. Faces extracted from videos are fed into Inception ResNet-BILSTM to extract frame-level learnable details, which are then used to classify real and fake videos. The approach demonstrates the ability to detect deepfake videos made using different techniques, varying illumination conditions, and diverse ethnicities. However, the model's limitation lies in its inapplicability for images and audio.

[15] Deepfakes Detection with Automatic Face Weighting by Daniel Mas Montserrat, Hanxiang Hao, S. K. Yarlagadda, Sriram Baireddy, Ruiting Shao, János Horváth, Emily Bartusiak, Justin Yang, David Guera, Fengqing Zhu, Edward J. Delp (2020):

This paper presents a deepfake detection method based on Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), extracting visual and temporal features from faces in videos. The method is evaluated with the DFDC dataset and introduces automatic face weighting, enabling the model to automatically weigh different face regions for improved accuracy. While the approach demonstrates fast processing, good accuracy, and an ability to dismiss any analysis of audio content, challenges arise in identifying manipulated videos when multiple faces are present.

CHAPTER 3

SYSTEM ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

Functional requirements are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users. Based on this, the functional requirements that the system must require areas follow:

- User of the application will be able detect the whether the uploaded video is fake or real, along with the model confidence of the prediction.
- The User will be able to see the playing video with the output on the face along with the confidence of the model.
- UI contains a browse tab to select the video for processing. It reduces the complications and at the same time enrich the user experience.

3.2 NON-FUNCTIONAL REQUIREMENTS

Nonfunctional requirements are a description of features, characteristics, and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. Mainly based on performance, information, economy, control and security efficiency and services. Requirements are as follows:

- The design is versatile and user friendly.
- The application is fast, reliable and time saving.
- The software should be efficiently designed to give reliable recognition of fake videos and so that it can be used for more pragmatic purpose.

3.3 HARDWARE REQUIREMENTS

1. Intel Xeon E5 2637: 3.5 GHz
2. RAM: 16 GB
3. Hard Disk: 100 GB
4. Graphic card: NVIDIA GeForce GTX Titan (12 GB RAM)

3.4 SOFTWARE REQUIREMENTS

1. Operating System: Windows 7+
2. Programming Language: Python 3.0
3. Framework: PyTorch 1.4, Django 3.0
4. Cloud platform: Google Cloud Platform
5. Libraries: OpenCV, Face-recognition

CHAPTER 4

DESIGN

4.1 USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The entities in the diagram are the Driver, the Camera, and the System. The information on driver is captured by the camera and fed to the system. The system then processes the information and alerts the driver.

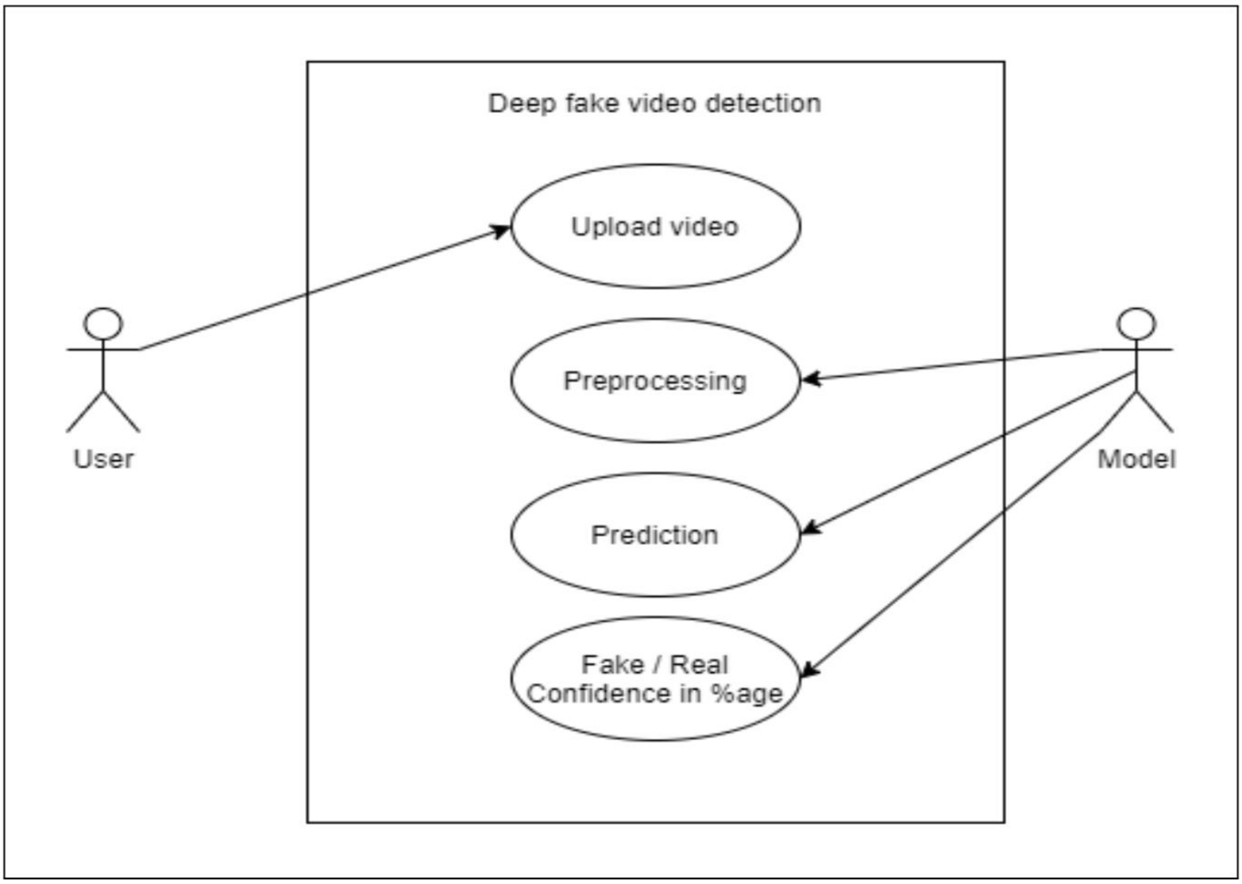


Fig 4.1:Use Case Diagram

4.2 DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.

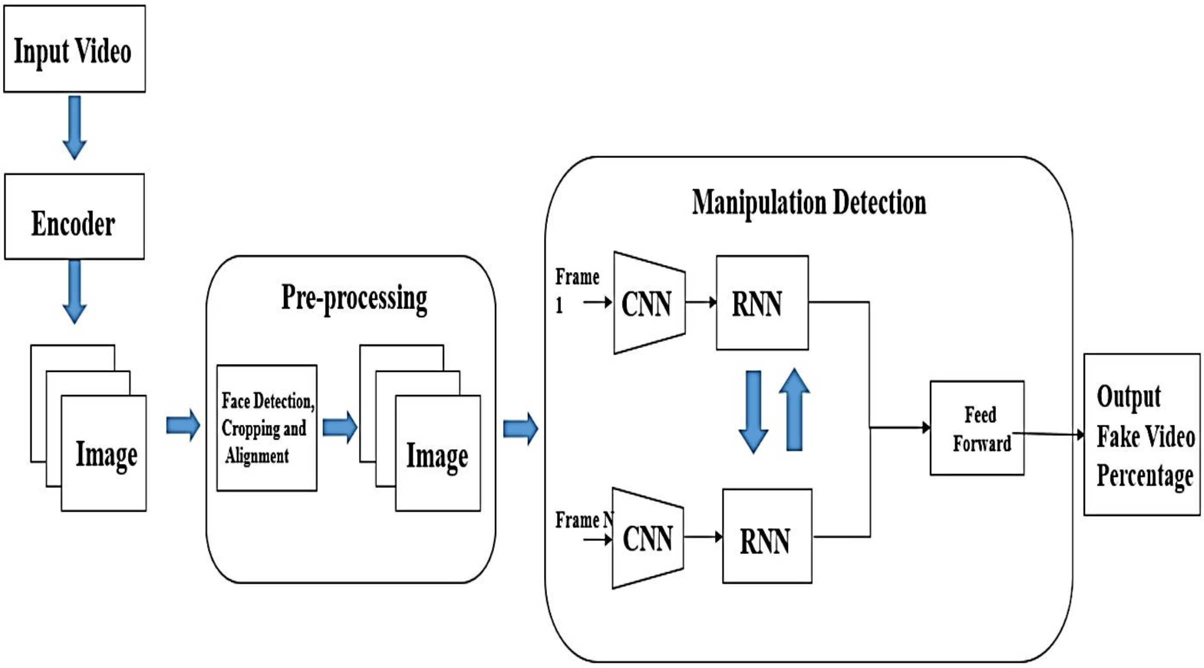


Fig 4.2: Data Flow Diagram

4.3 SEQUENTIAL DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

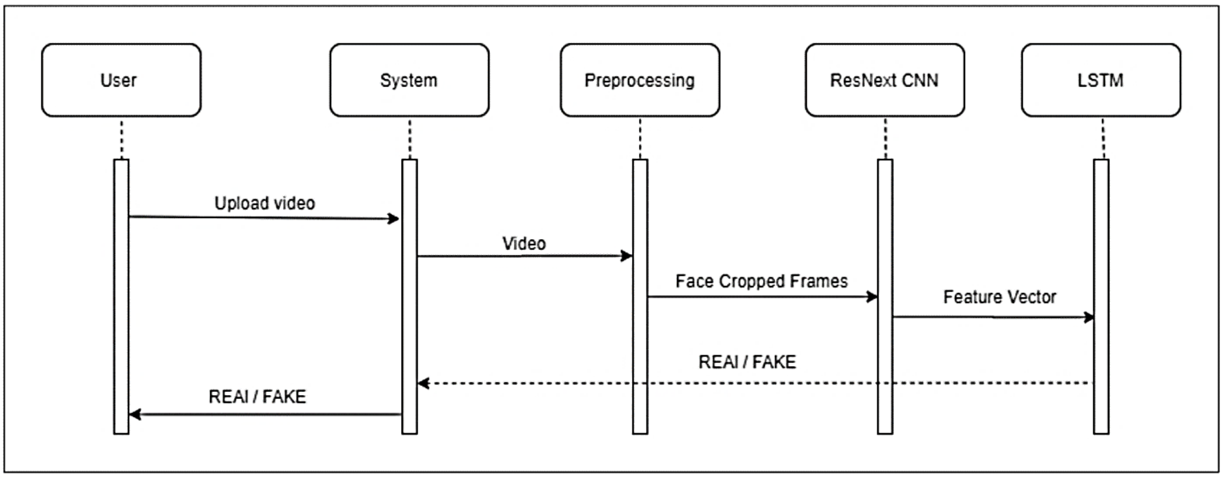


Fig 4.2: Sequential Diagram

4.4 ACTIVITY DIAGRAM

An activity diagram is a behavioural diagram i.e. it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

Training Workflow:

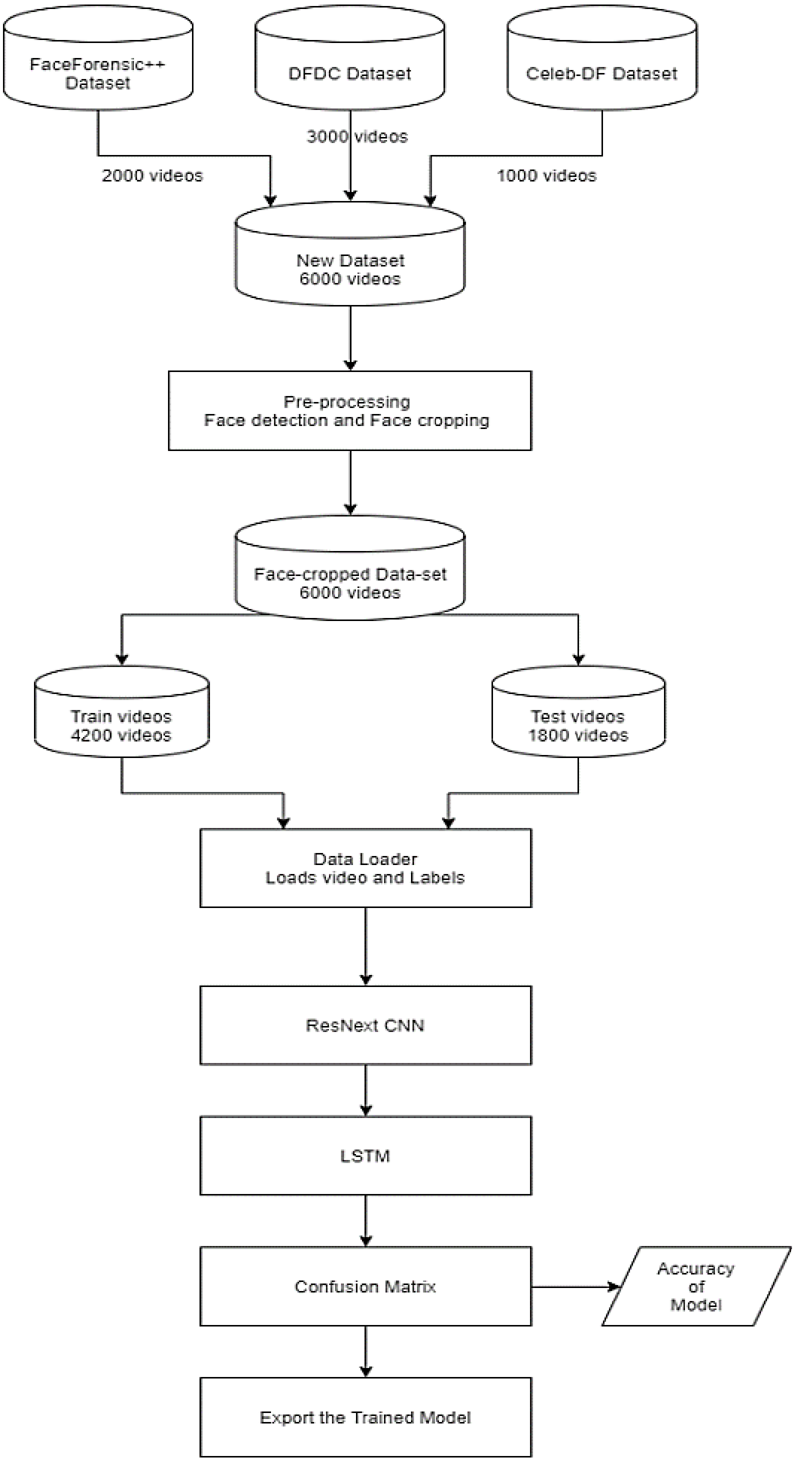


Fig 4.4: Training Workflow

Testing Workflow:

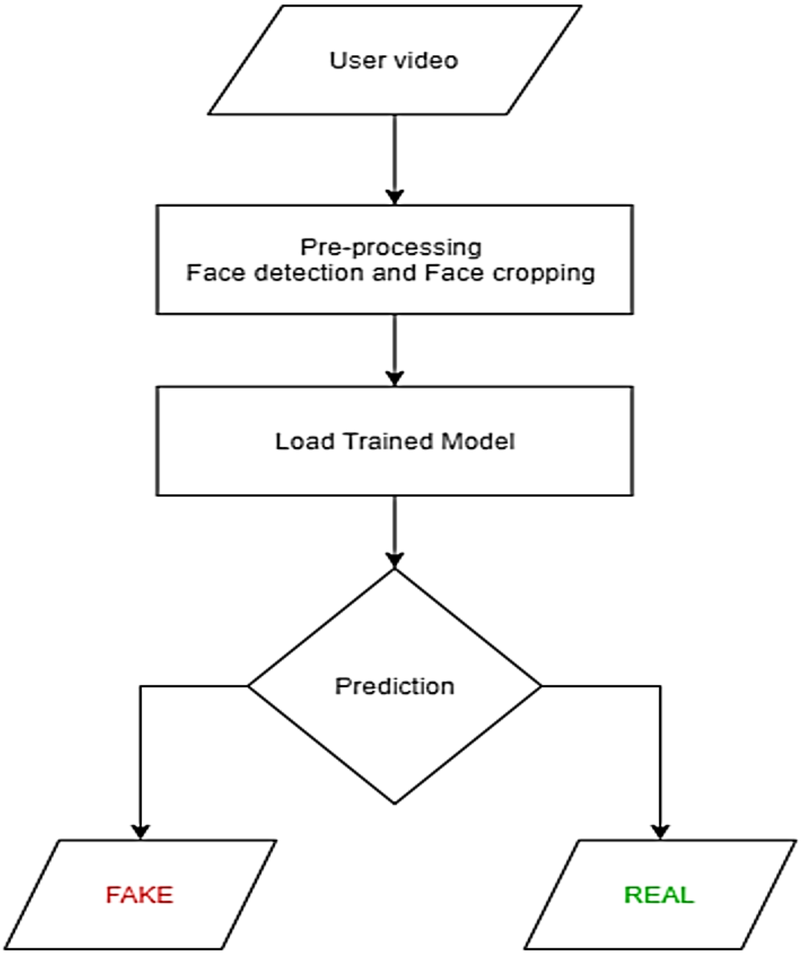


Fig 4.5: Testing Workflow

CHAPTER 5

DETAILED DESIGN DOCUMENT

5.1 SYSTEM ARCHITECTURE

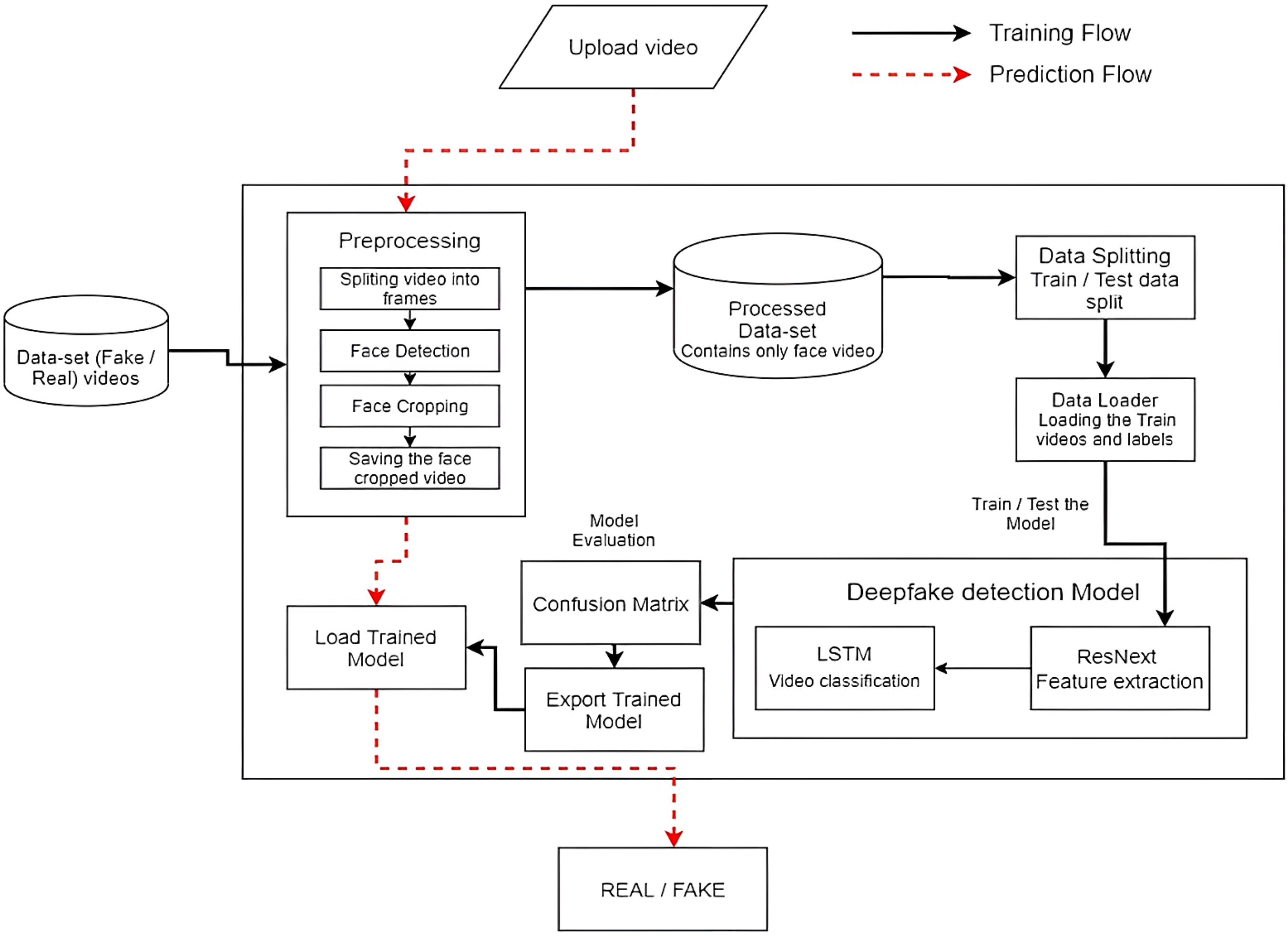


Fig 5.1: System Architecture

In this system, the PyTorch deepfake detection model has been trained on an equal number of genuine and synthetic videos to mitigate bias. The model's architecture is illustrated in the accompanying figure. During the developmental phase, a dataset was selected, pre-processed, and refined to exclusively feature videos with cropped faces.

- **Creating Deepfake Videos**

To effectively detect deepfake videos, understanding the process of their creation is crucial. Most tools, such as GANs and autoencoders, require a source image and target video as input. These tools break down the video into frames, identify faces within the video, and replace the source face with the target face in each frame. Subsequently, these altered frames are merged using various pre-trained models. These models further refine the video quality by eliminating any remaining traces left by the deepfake generation process, resulting in a deepfake that appears highly realistic.

Our approach to detecting deepfakes follows a similar methodology. Deepfakes generated using pre-trained neural network models can be incredibly convincing, making them nearly indistinguishable to the naked eye. However, despite their realism, the tools used to create deepfakes often leave behind subtle traces or artifacts in the video, which may go unnoticed by casual observation. The objective of our research is to identify these imperceptible traces and distinct artifacts present in deepfake videos, enabling us to classify them accurately as either deepfake or genuine.

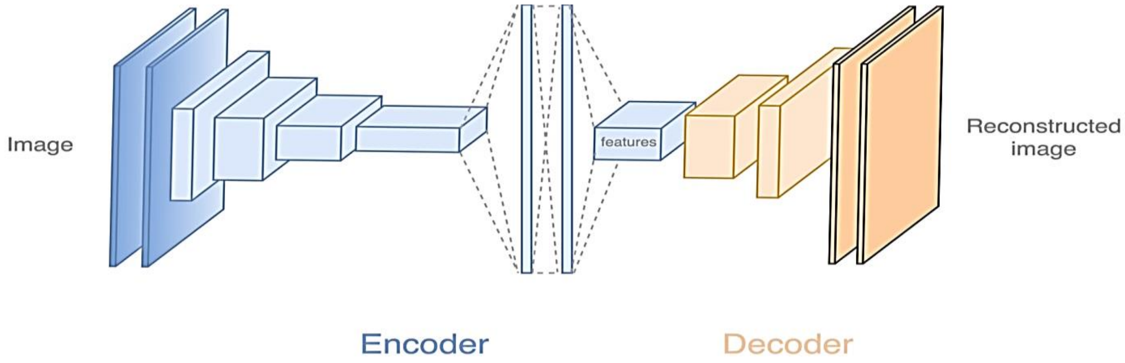


Fig 5.2: Deepfake Generation

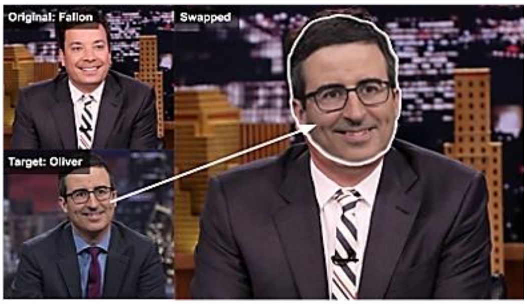


Fig 5.3: Face Swapped Deepfake Generation

- **Tools for deep fake creation**

1. Faceswap
2. Faceit
3. Deep Face Lab
4. Deepfake Capsule GAN
5. Large resolution face masked

5.2 ARCHITECTURAL DESIGN

5.2.1 MODULE 1: DATA-SET GATHERING

To ensure efficient real-time prediction, we've compiled data from various sources, including FaceForensic++ (FF), the Deepfake Detection Challenge (DFDC), and Celeb-DF. Combining these datasets, we've curated a new dataset tailored for accurate and rapid detection across different video types. To prevent model bias during training, we've maintained a balanced ratio of 50% genuine and 50% synthetic videos.

The DFDC dataset contains certain audio-altered videos, which fall outside the scope of our paper focused on visual deepfake detection. We've pre-processed the DFDC dataset by removing these audio-altered videos using a Python script.

Following preprocessing, we selected 1500 genuine and 1500 synthetic videos from the DFDC dataset, 1000 genuine and 1000 synthetic videos from FF, and 500 genuine and 500 synthetic videos from Celeb-DF. This brings our total dataset to 3000 genuine, 3000 synthetic, and 6000 videos overall. The distribution of these datasets is illustrated in Fig 5.4.

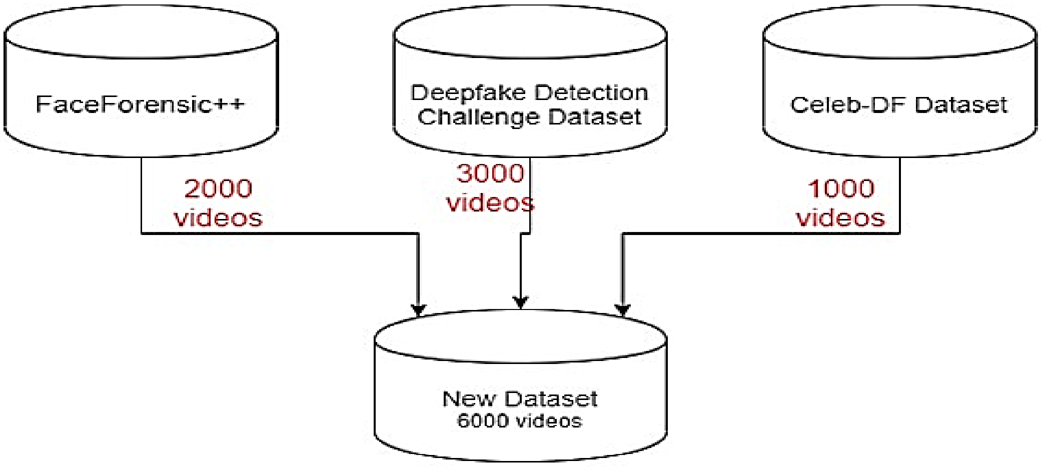


Fig 5.4: Dataset Distribution

5.2.2 MODULE 2: PRE-PROCESSING

In this stage of preprocessing, the videos undergo several steps to extract only the necessary information, particularly focusing on detecting and cropping faces. Here's a paraphrased version: The initial step involves splitting the video into frames. Each frame is then analysed to detect and crop the face, discarding any non-face elements and noise. Subsequently, the cropped frames are compiled back into a new video, containing only the face portions.

To ensure consistency in frame counts and manage computational resources effectively, a threshold value is determined based on the average frame count per video. This threshold value is crucial due to computational limitations, especially in processing large volumes of frames simultaneously. For instance, a 10-second video at 30 frames per second would yield 300 frames, which might overwhelm the system's computational capabilities. Hence, to accommodate the computational constraints of the GPU in the experimental setup, a threshold of 150 frames is chosen.

During the creation of the processed dataset, only the initial 150 frames of each video are retained, maintaining sequential order for proper utilization in Long Short-Term Memory (LSTM) models. The resultant videos are saved at a frame rate of 30 frames per second and a resolution of 112 x 112 pixels.

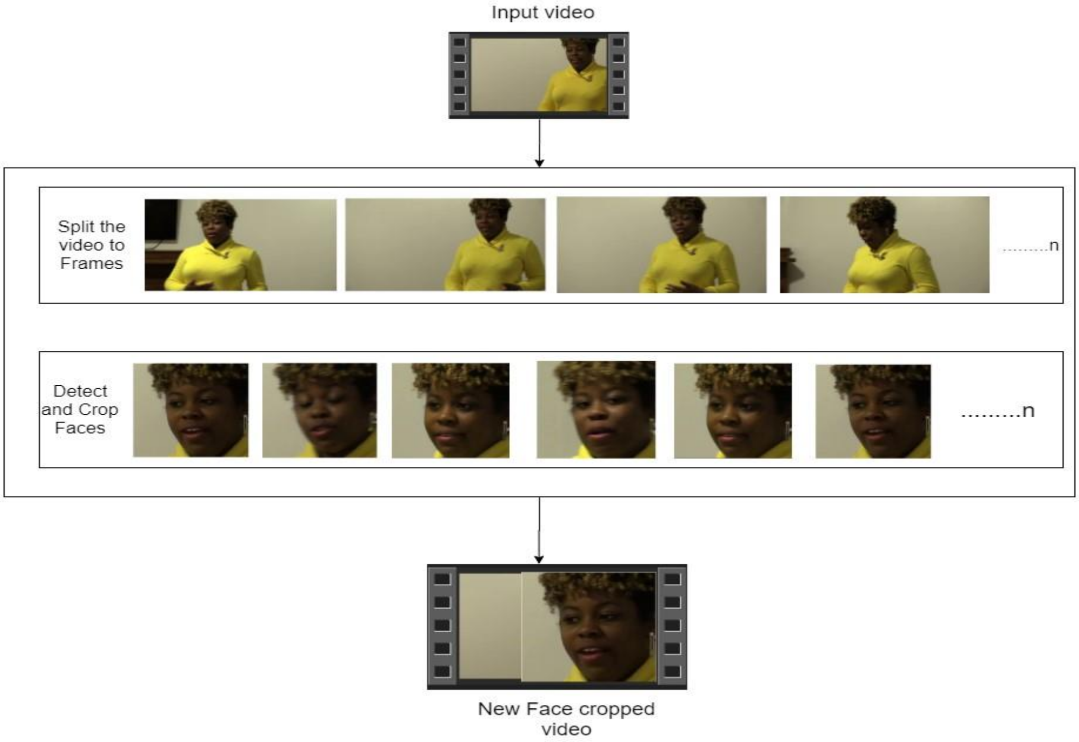


Fig 5.5: Pre-processing of video

5.2.3 MODULE 3: DATA-SET SPLIT

The dataset is divided into a training set and a test set, maintaining a ratio of 70% for training videos (4,200 videos) and 30% for test videos (1,800 videos). To ensure a balanced representation of real and fake videos in both the training and test sets, each split contains an equal distribution of 50% real videos and 50% fake videos. This balanced split helps prevent biases and ensures that the model learns to generalize well across both real and fake video data.

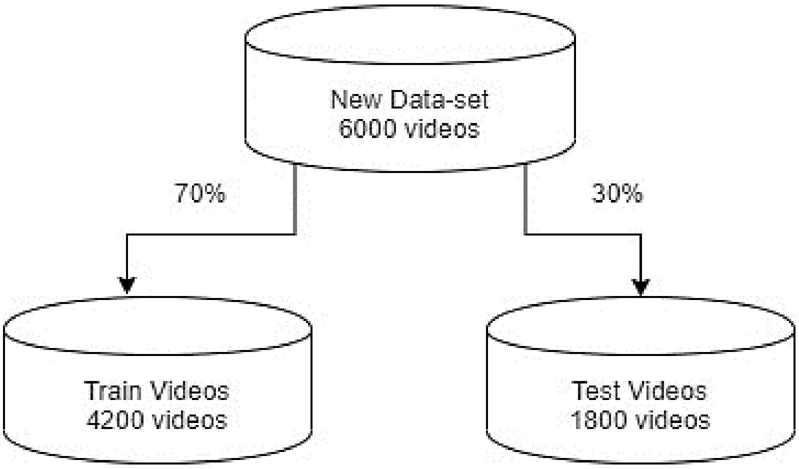


Fig 5.6: Train-test split

5.2.4 MODULE 4: MODEL ARCHITECTURE

Our model combines Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for video classification, specifically focusing on distinguishing between deepfake and pristine videos. We leverage a pre-trained ResNext CNN model for feature extraction at the frame level, followed by training a Long Short-Term Memory (LSTM) network to classify the videos.

ResNext:

Instead of developing the model architecture from scratch, we employ the pre-trained ResNext model for feature extraction. ResNext is a variant of the Residual CNN network designed to excel in deeper neural networks. For our experiments, we utilize the resnext50_32x4d model, which consists of 50 layers and features a 32 x 4-dimension setup.

We fine-tune the ResNext model by adding necessary additional layers and adjusting the learning rate for optimal convergence of the gradient descent during training. The 2048-dimensional feature vectors obtained after the final pooling layers of ResNext serve as the input to the sequential LSTM.

LSTM for sequence processing:

The 2048-dimensional feature vectors are fed into the LSTM for sequential processing. We employ a single LSTM layer with 2048 latent dimensions and 2048 hidden layers, along with a dropout probability of 0.4 to prevent overfitting. The LSTM analyses the video frames sequentially, enabling temporal analysis by comparing frames at different time points.

Model architecture:

The model incorporates Leaky ReLU activation functions and a linear layer with 2048 input features and 2 output features to facilitate learning the correlation between input and output. An adaptive average pooling layer with an output parameter of 1 is utilized to reshape the output to the desired dimensions (H x W). Sequential layer processing is facilitated using a Sequential Layer. Batch training is performed with a batch size of 4. Finally, a SoftMax layer is employed to obtain confidence scores for model predictions.

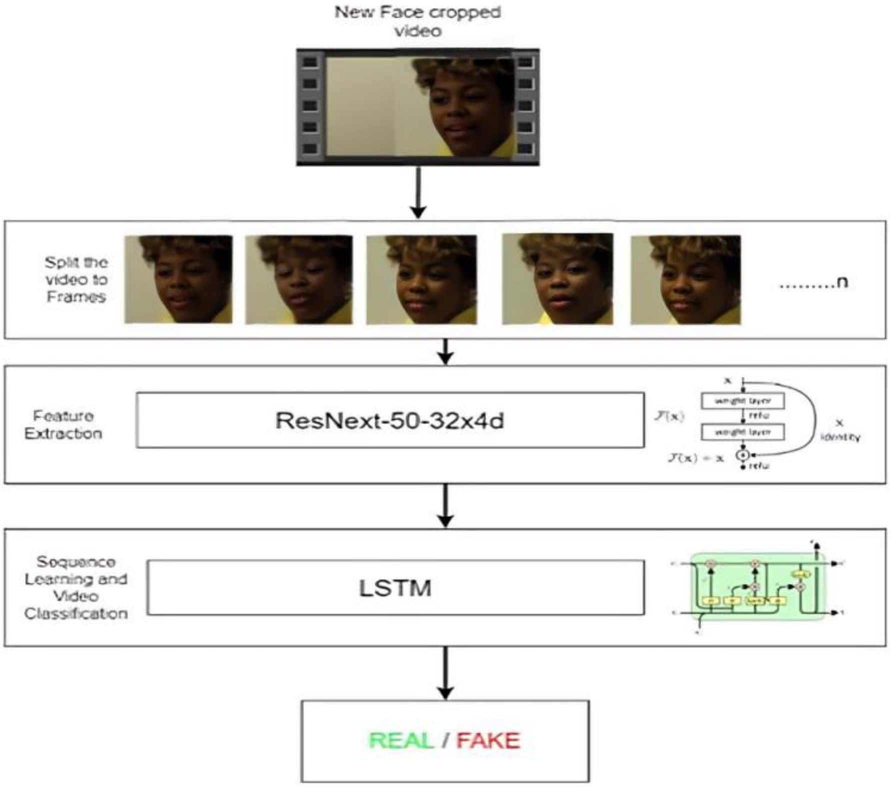


Fig 5.7: Overview of Model

5.2.5 MODULE 5: HYPER-PARAMETER TUNING

In the process of hyperparameter tuning, the goal is to select the optimal parameters to achieve maximum accuracy. After multiple iterations on the model, the best hyperparameters for our dataset are identified. To facilitate adaptive learning rate, we employ the Adam optimizer along with the model parameters. The learning rate is finely tuned to 1e-5 (0.00001) to reach a better global minimum during gradient descent. Additionally, a weight decay of 1e-3 is utilized.

Given that this is a classification problem, we employ the cross-entropy loss function. To efficiently utilize available computational resources, we opt for batch training with a batch size of 4. This size has been determined to be optimal for training in our development environment.

For the user interface, we utilize the Django framework to ensure scalability and maintainability of the application in the future. The initial page of the user interface, index.html, features a tab allowing users to browse and upload videos. Upon upload, the video is passed to the model for prediction. The model returns an output indicating whether the video is real or fake, along with the confidence level. This output is then displayed on the predict.html page, overlaid on the playing video.

CHAPTER 6

IMPLEMENTATION

6.1 TOOLS AND TECHNOLOGIES USED

6.1.1 PROGRAMMING LANGUAGES

- **PYTHON3:**

Python is the go-to language for implementing a deepfake detection system using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Its rich ecosystem of machine learning libraries like TensorFlow and PyTorch simplifies complex model development. Python's readability and simplicity facilitate collaborative development and experimentation, while its extensive community support ensures access to resources and insights. Its versatility enables seamless integration with other tools, streamlining the entire development process from data preprocessing to model deployment. Overall, Python's combination of powerful libraries, simplicity, community support, and interoperability make it the ideal choice for building effective deepfake detection systems.

- **JAVASCRIPT:**

JavaScript, widely known for web development, isn't commonly used for deep learning compared to Python. Python's specialized libraries like TensorFlow and PyTorch are preferred for machine learning tasks due to their extensive support and performance. Despite this, emerging libraries such as TensorFlow.js and Brain.js enable JavaScript to tackle some deep learning tasks, albeit with less maturity and performance than Python. Overall, while JavaScript offers possibilities for deep learning, Python remains the primary choice for implementing complex algorithms like CNNs and RNNs due to its established ecosystem and better-suited syntax.

6.1.2 PROGRAMMING FRAMEWORKS

- **PyTorch:**

PyTorch is a machine learning library developed by Facebook's AI Research lab (FAIR). It's widely used for deep learning tasks, offering a flexible framework for building and training neural networks. With its dynamic computation graph feature, PyTorch simplifies model development and debugging. It provides seamless integration with Python and other libraries like NumPy. PyTorch supports GPU acceleration, enabling efficient training on GPUs for handling large datasets and complex models. Its ease of use, flexibility, and active community have made PyTorch a popular choice among researchers and practitioners in the field of machine learning.

- **Django:**

Django is a Python web framework designed for rapid development of web applications. It provides a set of tools and functionalities that streamline common web development tasks, such as URL routing, database interaction, and user authentication. Django follows the "batteries-included" approach, offering a comprehensive set of features out of the box, including an admin interface for managing site content, a powerful ORM for database interactions, and built-in security features. Its clean and pragmatic design allows developers to focus on building their applications without getting bogged down in repetitive tasks. Overall, Django is a robust and versatile framework that simplifies the process of creating complex web applications in Python.

6.1.3 IDE

- **Google Colab:**

Google Colab is a cloud-based platform for writing and executing Python code in a browser environment. It provides free access to GPUs and TPUs for accelerating computations, integrates with Google Drive, and supports Jupyter Notebooks for interactive coding. With pre-installed libraries and collaboration features, it's popular among data scientists and machine learning practitioners for its convenience and accessibility.

- **Jupyter Notebook:**

Jupyter Notebook is an open-source web application that allows users to create and share documents containing live code, equations, visualizations, and narrative text. It supports multiple programming languages, including Python, R, and Julia, and enables interactive computing through code execution in individual cells. With rich output capabilities, Markdown support, and flexible export options, Jupyter Notebook is widely used for data analysis, machine learning, scientific research, and educational purposes.

- **Visual Studio Code:**

Visual Studio Code, often abbreviated as VS Code, is a popular source-code editor developed by Microsoft. It's renowned for its versatility, speed, and extensibility through plugins available in the Visual Studio Code Marketplace. It supports various programming languages and offers features like syntax highlighting, code completion, debugging capabilities, Git integration, and more. Its lightweight nature, combined with its powerful features and active community support, has made it a favourite among developers across different platforms.

6.1.4 VERSIONING CONTROL

- **Git:**

Git is a distributed version control system for tracking changes in source code during software development. It allows multiple developers to collaborate on projects efficiently by managing changes to files, creating snapshots of code (commits), branching for parallel development, merging changes, and facilitating collaboration through pull requests. Git is widely used for its flexibility, speed, and robust version control capabilities.

6.1.5 CLOUD SERVICES

- **Google Cloud Platform**

6.1.6 APPLICATION AND WEB SERVICES

- **Google Cloud Engine**

6.1.7 LIBRARIES

- **torch**
- **torchvision**
- **os**
- **numpy**

- **cv2**
- **matplotlib**
- **face_recognition**
- **json**
- **pandas**
- **copy**
- **glob**
- **random**
- **sklearn**

6.2 ALGORITHM DETAILS

6.2.1 DATASET DETAILS

Refer 5.2.1 (DATASET GATHERING)

6.2.2 PREPROCESSING DETAILS

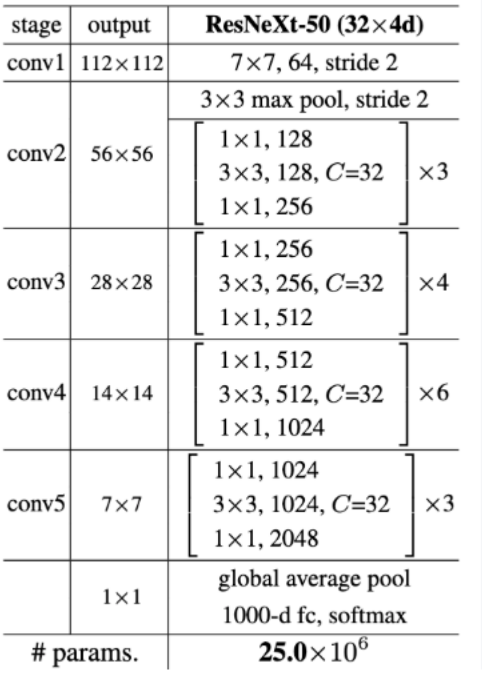
- We utilized the `glob` module to gather all video files within a directory and stored them in a Python list.
- Employing `cv2.VideoCapture`, we accessed each video file to calculate the mean number of frames in each video.
- To ensure consistency across videos, we determined a target value of 150 frames as ideal for creating a new dataset.
- Videos were decomposed into individual frames, with subsequent facial cropping performed on each frame.
- The cropped facial frames were then assembled into new videos using the `VideoWriter` function.
- These new videos were encoded with a frame rate of 30 frames per second and a resolution of 112 x 112 pixels, saved in the mp4 format.

- Instead of opting for random frame selection, the initial 150 frames were chosen to align with the requirements for LSTM-based temporal sequence analysis.

6.2.3 MODEL DETAILS

The model consists of following layers:

- The ResNext CNN model, specifically the pre-trained `resnext50_32x4d()` model, is employed for the task at hand. With its architecture composed of 50 layers and dimensions of 32 x 4, this model offers a robust framework for deep learning tasks. The detailed implementation of the model, as illustrated in the accompanying figure, showcases its intricate design and structure. By leveraging the capabilities of this pre-trained model, we aim to achieve high-performance results in our application, benefiting from the depth and complexity inherent in its architecture.



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

Fig 6.1: ResNext Architecture

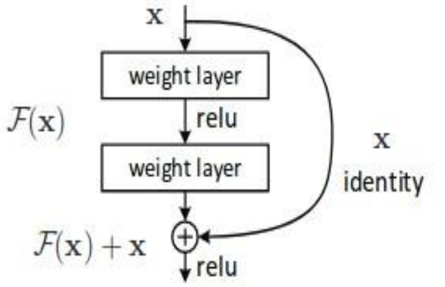

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig 6.2: ResNext Working

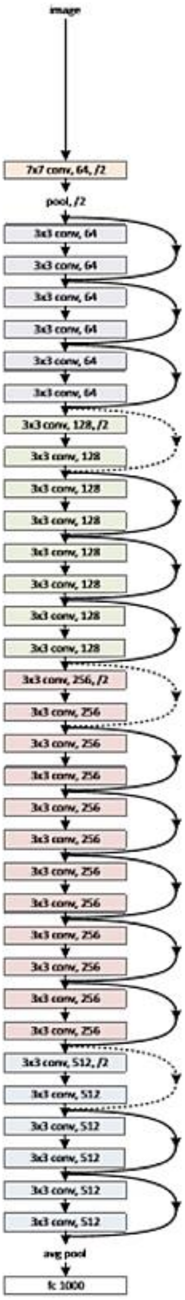


Fig 6.3: Overview of ResNext Architecture

- Sequential Layer: The Sequential layer serves as a container for Modules, allowing them to be stacked and executed simultaneously. In our context, the Sequential layer is utilized to organize and store the feature vector returned by the ResNext model in a structured manner. This ensures that the feature vector can be sequentially passed to the LSTM (Long Short-Term Memory) model, facilitating the temporal sequence analysis required for our task. By employing the Sequential layer, we maintain the order of features and enable seamless integration with subsequent layers, enhancing the efficiency and effectiveness of our deep learning architecture.
- The LSTM (Long Short-Term Memory) layer is crucial for sequence processing and capturing temporal changes between frames in our task. We input 2048-dimensional feature vectors into the LSTM, which aids in comprehensively analysing the temporal dynamics of the video data. Our LSTM architecture consists of a single layer with 2048 latent dimensions and 2048 hidden layers, complemented by a dropout probability of 0.4. This configuration is adept at fulfilling our objectives effectively. By leveraging the LSTM layer, we process frames in a sequential manner, enabling a detailed temporal analysis of the video content. This involves comparing the frame at time 't' with frames at earlier instances ('t-n' seconds), where 'n' represents any number of frames preceding time 't'. Such temporal analysis is instrumental in identifying patterns, trends, and changes over time within the video dataset.

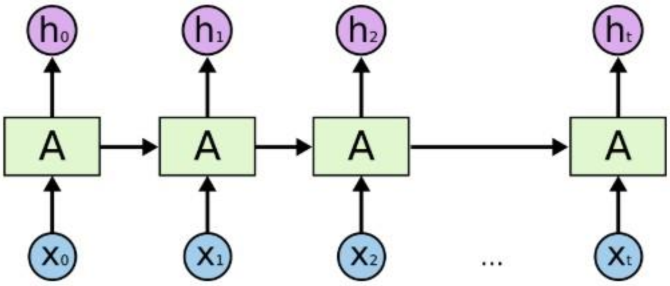


Fig 6.4: Overview of LSTM Architecture

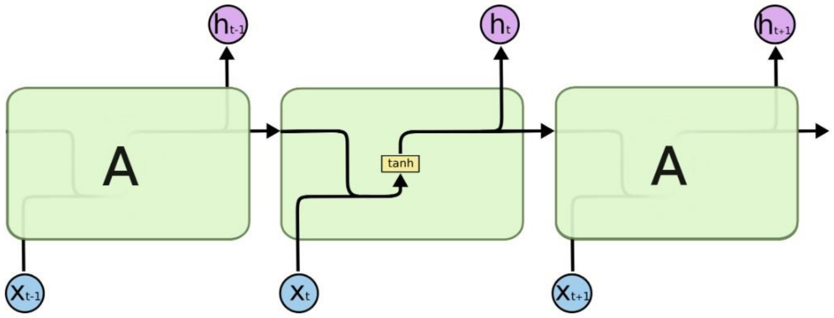


Fig 6.5: Internal LSTM Architecture

- ReLU (Rectified Linear Unit) is an activation function commonly used in neural networks. It operates by outputting 0 if the input is less than 0, and the raw input otherwise. In essence, if the input is greater than 0, the output equals the input. This activation function closely mimics the behaviour of biological neurons, enhancing the biological plausibility of neural network models. ReLU is non-linear, allowing neural networks to learn complex relationships in data, and it circumvents the vanishing gradient problem associated with activation functions like the sigmoid function. Moreover, ReLU's simple mathematical operation enables rapid model building, making it particularly advantageous for training larger neural networks efficiently.

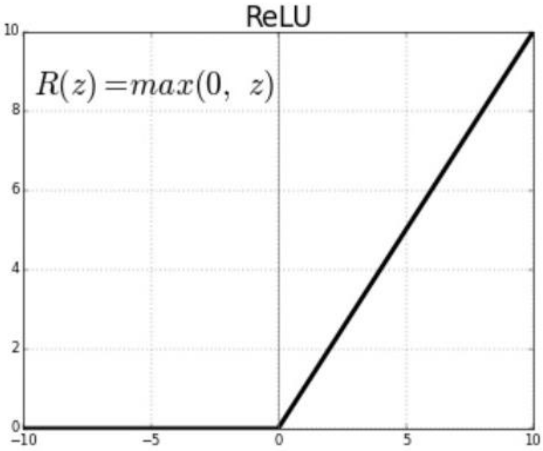


Fig 6.6: ReLU Activation Function

- The Dropout layer, set at a rate of 0.4, is incorporated into the model to counteract overfitting. By randomly deactivating 40% of neurons during training, it encourages the model to generalize better by relying on a wider array of features. This regularization technique introduces noise, prompting neighbouring neurons to adapt and learn more robust representations of the data. Consequently, during backpropagation, the cost function becomes more sensitive to the contributions of surrounding neurons, optimizing weight updates. Overall, Dropout enhances the model's ability to generalize and prevents it from memorizing noise or outliers in the training data.

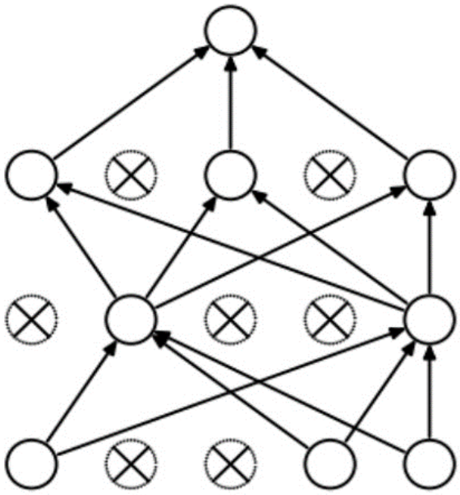


Fig 6.7: Dropout Layer Overview

- The Adaptive Average Pooling Layer, implemented as a 2-dimensional component in our model, dynamically adjusts pooling regions to reduce variance, computational complexity, and extract low-level features from local neighbourhoods.

6.2.4 MODEL TRAINING DETAILS

- Train Test Split: The dataset undergoes a balanced split into training and testing sets, with 70% (4,200) allocated to training and 30% (1,800) to testing. Each split comprises an equal distribution of real and fake videos, ensuring balance.
- Data Loader: A data loader is employed to efficiently load videos and their corresponding labels, utilizing a batch size of 4 to enhance processing speed.
- Training: The model is trained over 20 epochs, utilizing a learning rate of 1e-5 (0.00001) and a weight decay of 1e-3 (0.001). The Adam optimizer is chosen for its adaptive learning rate capabilities, facilitating efficient optimization of model parameters.
- Adam Optimizer: The Adam optimizer is selected to optimize the model parameters by adjusting the learning rate adaptively, enhancing convergence speed and performance.
- Cross Entropy: The Cross Entropy approach is adopted to compute the loss function, suitable for training the classification problem at hand.

- Softmax Layer: A Softmax function, serving as a type of squashing function, is employed. It confines the output within the range of 0 to 1, enabling direct interpretation as probabilities. In our model, the Softmax layer, with two output nodes representing "REAL" or "FAKE," facilitates confidence-based predictions.

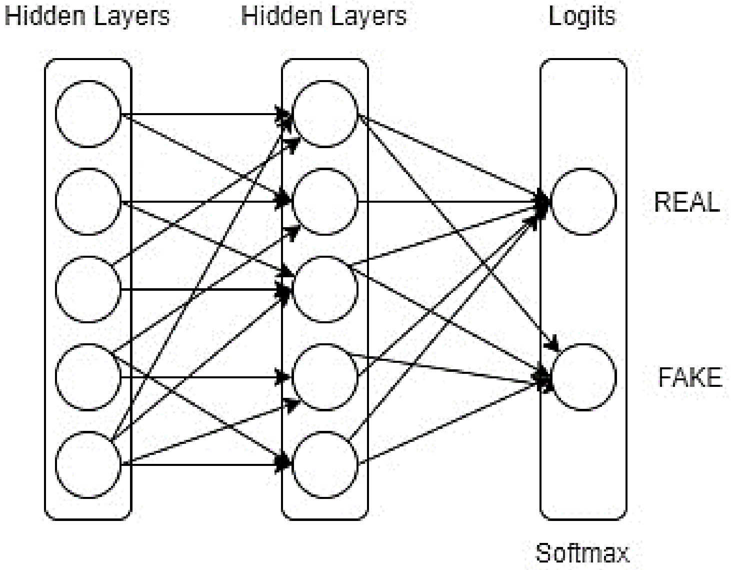


Fig 6.8: Softmax Layer

- Confusion Matrix: A confusion matrix provides a comprehensive summary of prediction results in a classification problem, detailing the count of correct and incorrect predictions for each class. It elucidates the manner in which the classification model is confused during predictions, offering valuable insights into the types of errors made. By analysing the confusion matrix, we gain a deeper understanding of the classifier's performance and identify specific areas for improvement. Additionally, the confusion matrix is instrumental in calculating the accuracy of the model, serving as a key evaluation metric.
- Export Model: Upon completion of training, the model is exported for future use in real-time data prediction scenarios. This ensures that the trained model can be readily deployed and applied to new data instances, enabling efficient and effective predictions. Exporting the model facilitates its integration into various applications and systems, enhancing accessibility and usability.

6.2.5 MODEL PREDICTION DETAILS

- Loading the Model: The application loads the pre-trained model, ensuring that it is ready for prediction tasks.

- Preprocessing New Video: The new video intended for prediction undergoes preprocessing steps as outlined in sections 8.3.2 and 7.2.2. These steps may include tasks such as frame extraction, resizing, normalization, or any other necessary preprocessing techniques.
- Prediction Process: The pre-processed video data is then passed to the loaded model for prediction. The model utilizes its learned parameters to make predictions regarding the authenticity of the video (real or fake).
- Prediction Output: Upon completion of the prediction process, the trained model returns the prediction results. This includes indicating whether the video is classified as real or fake, along with the confidence level associated with the prediction. The confidence level represents the model's certainty in its prediction, typically expressed as a probability score.

CHAPTER 7

TESTING

7.1 RESULT SCREENSHOTS

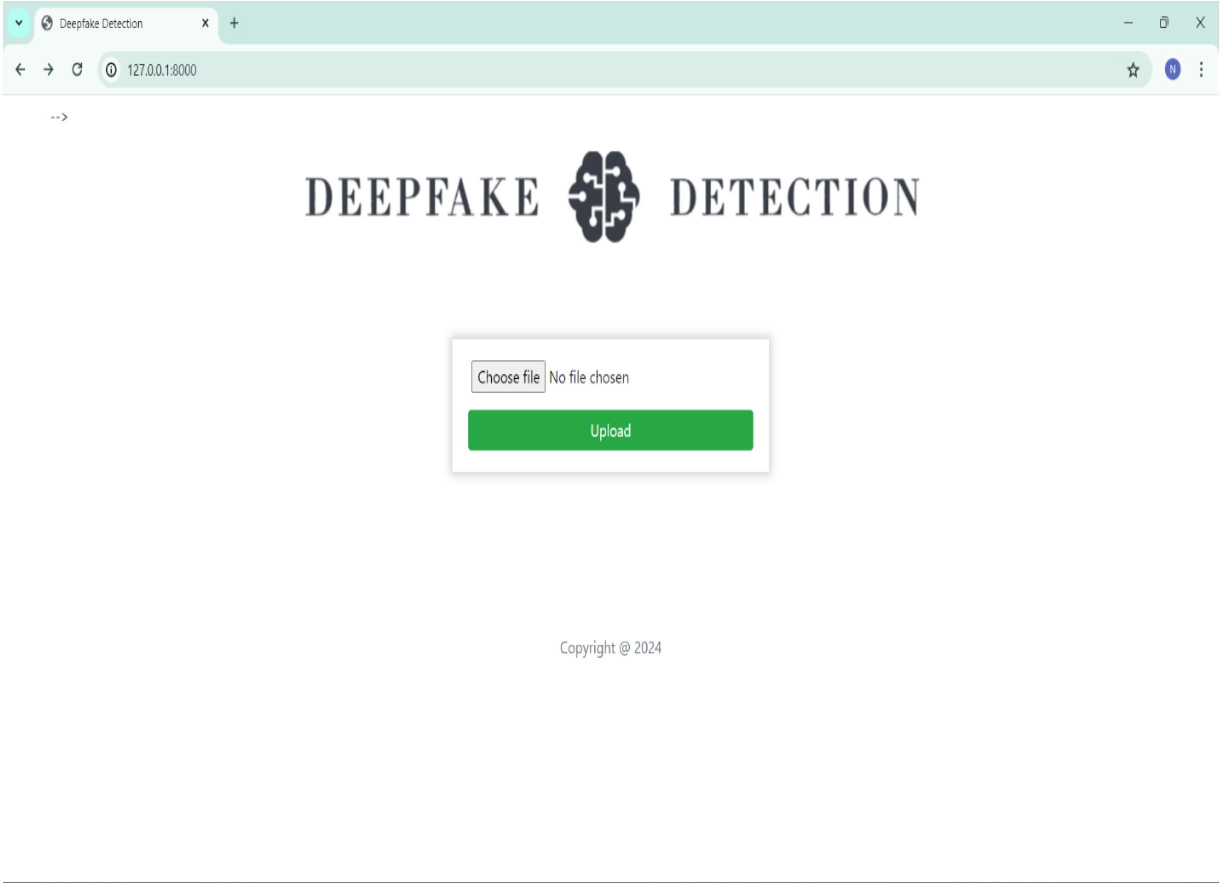


Fig 7.1: Home Page

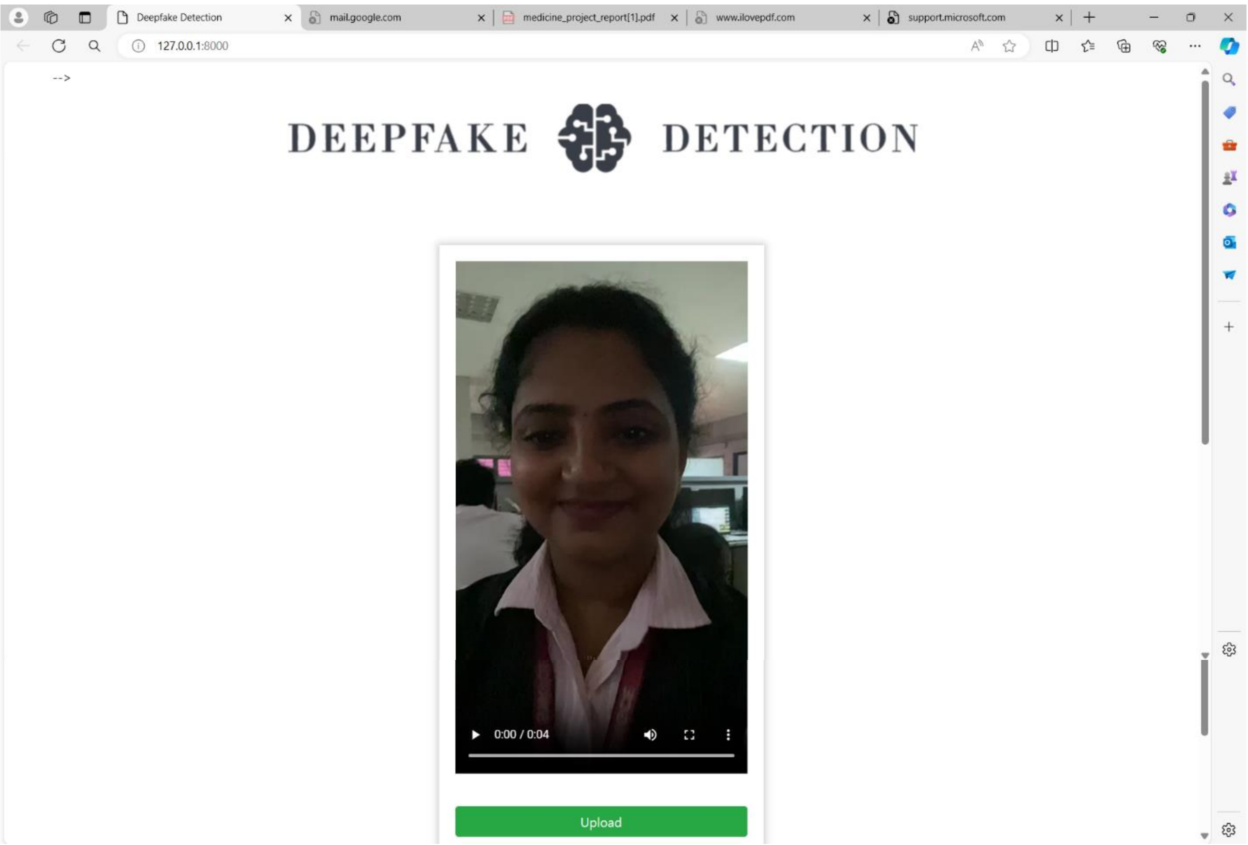
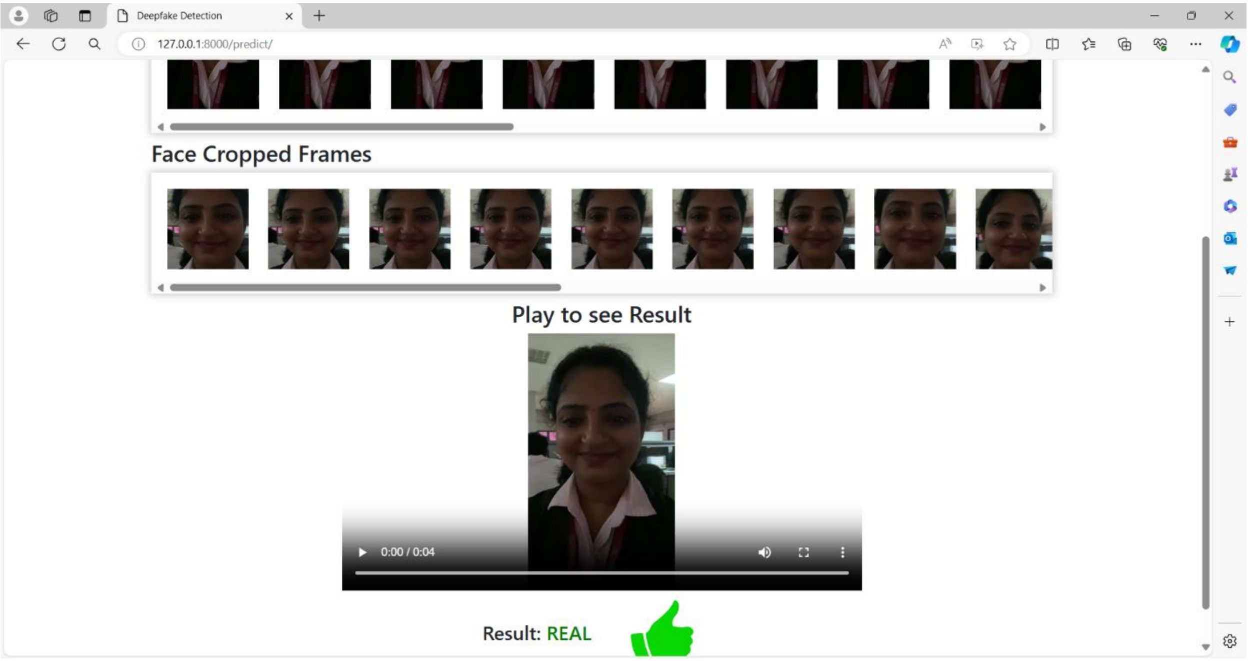


Fig 7.2: Upload Real Video



Result: REAL

Fig 7.3: Real Video Output

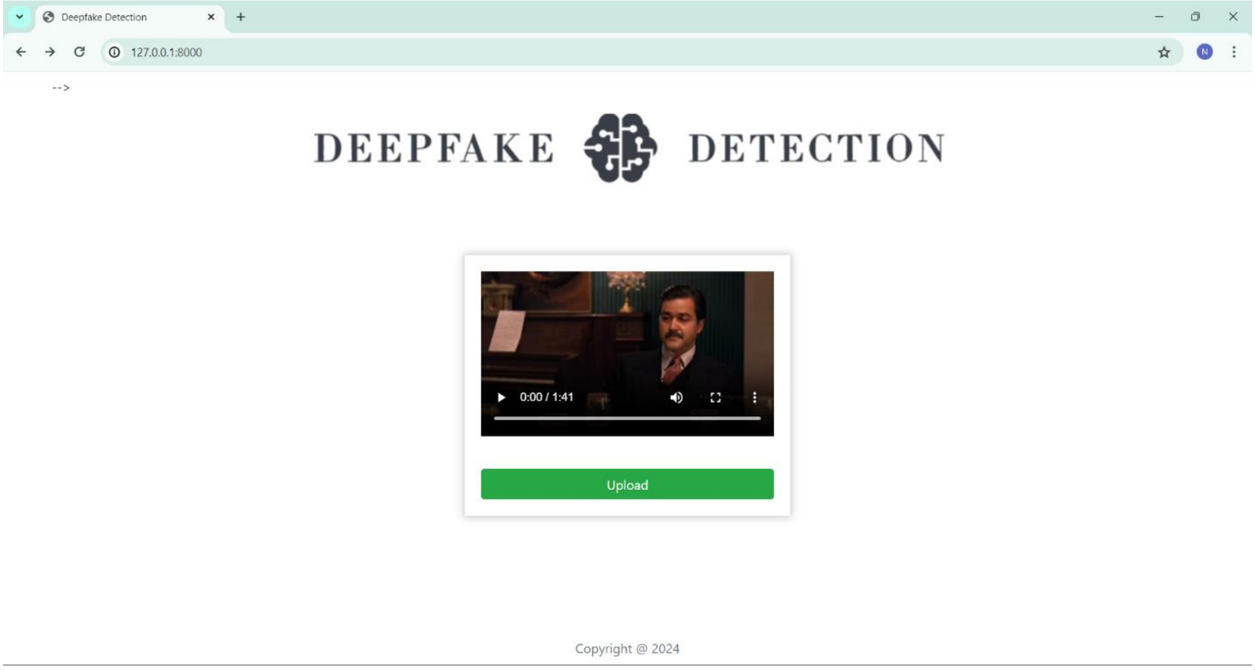


Fig 7.4: Uploading Deepfake Video

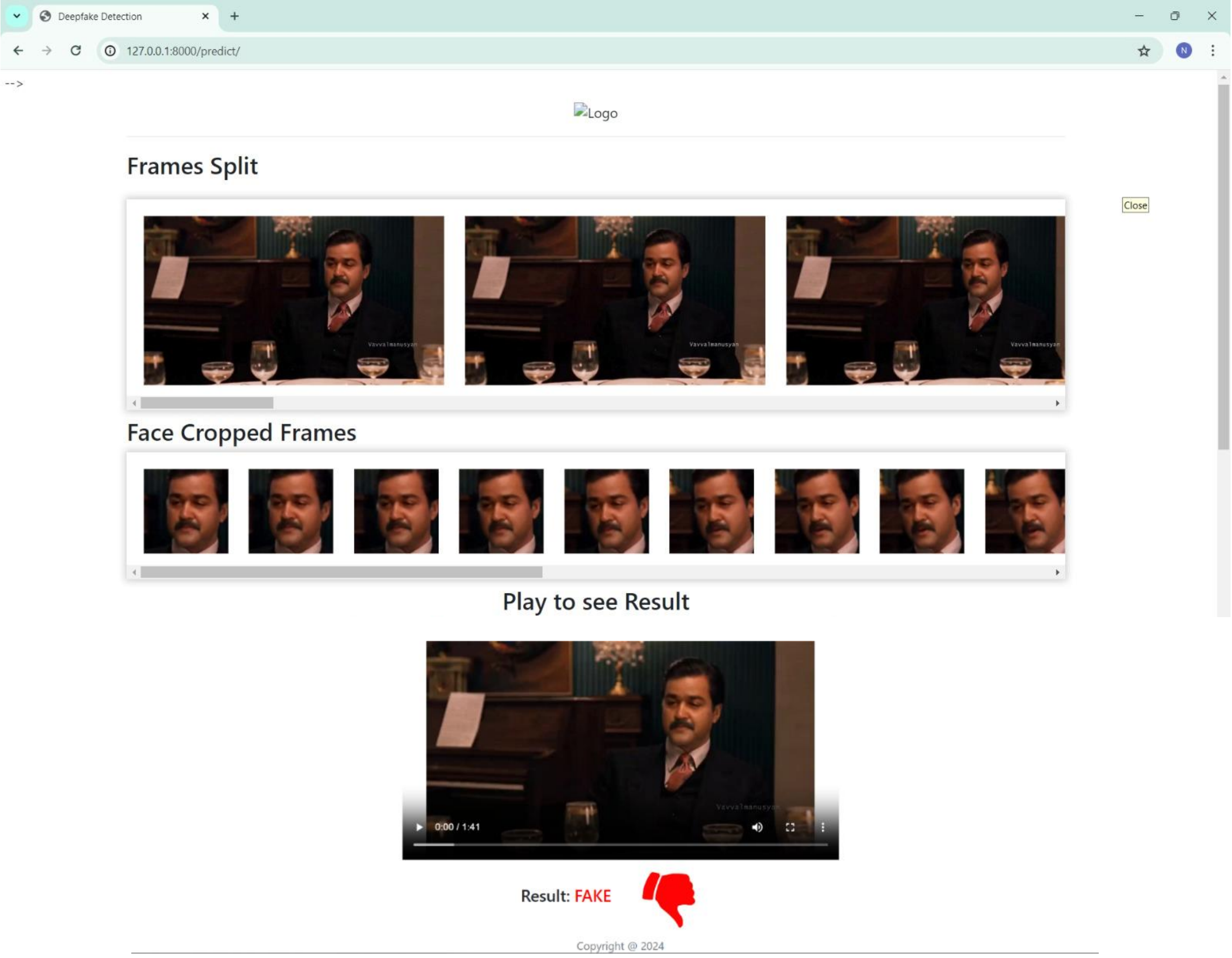


Fig 7.5: Deepfake Video Output

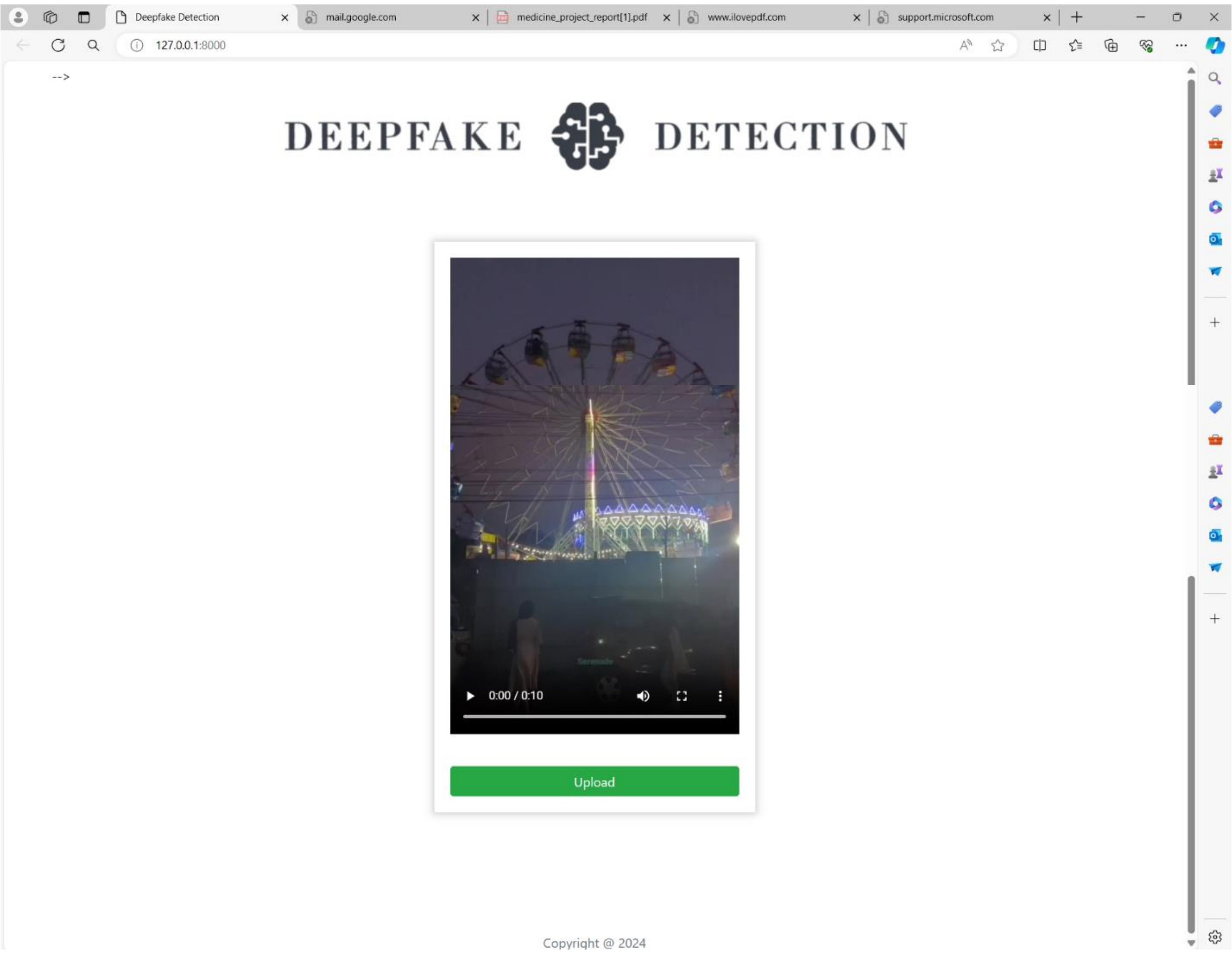
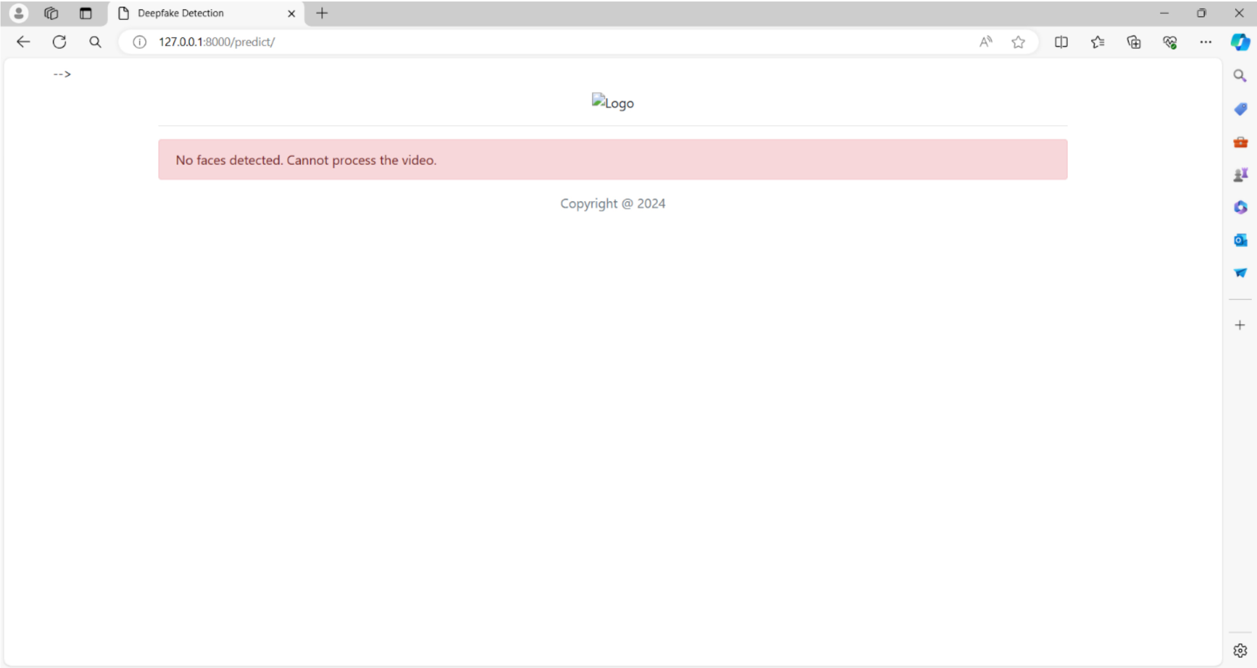
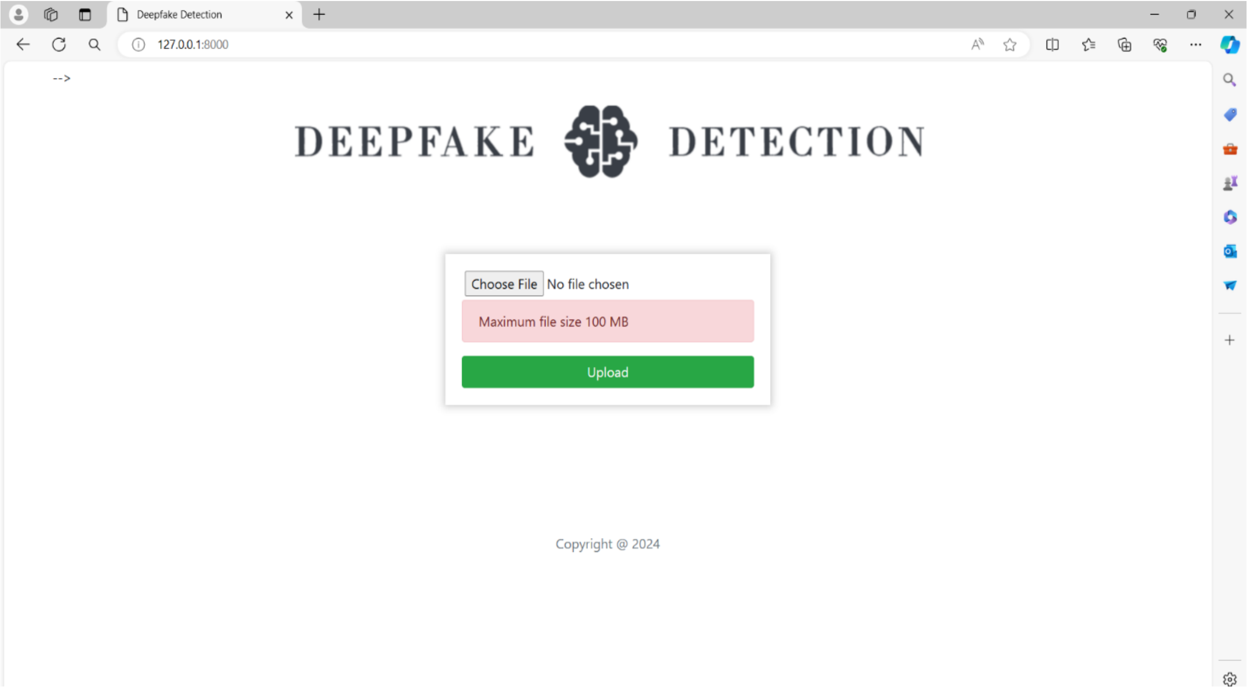


Fig 7.6: Upload Video with No Face



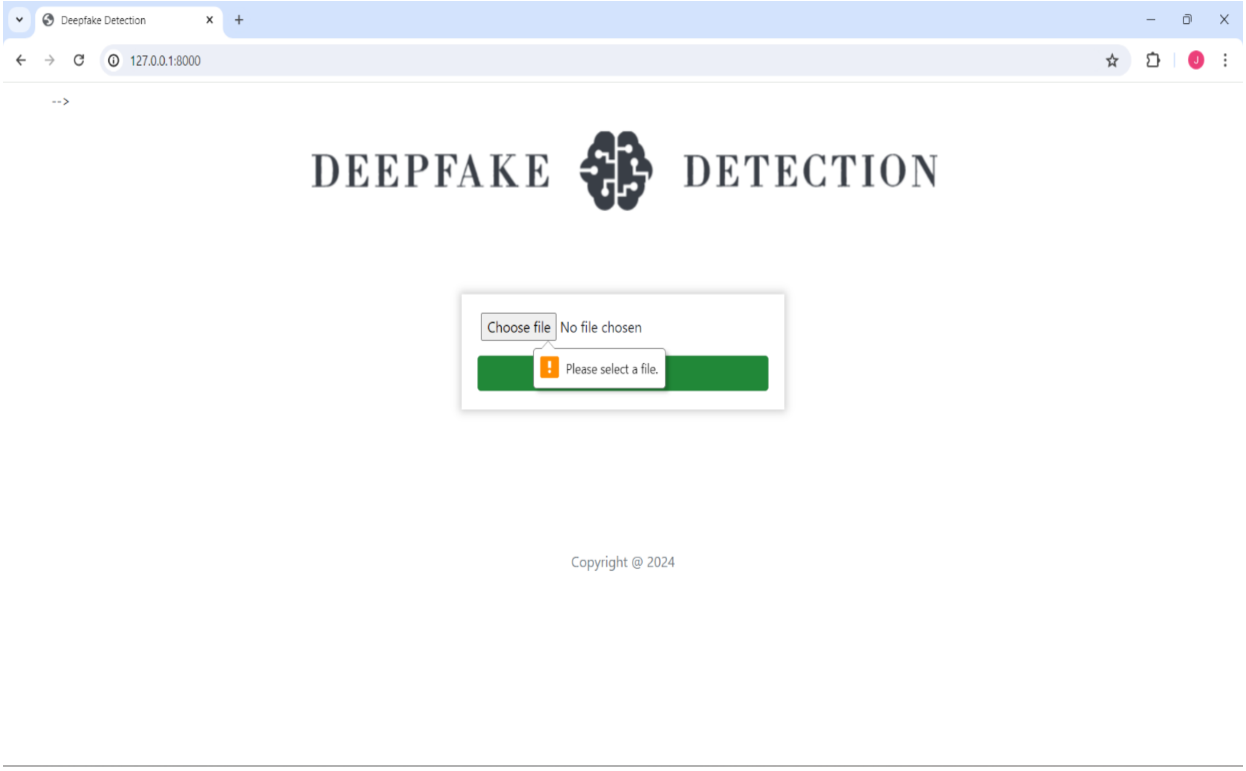
No faces detected. Cannot process the video.

Fig 7.7: Output of Uploaded Video with No Faces



Maximum file size 100 MB

Fig 7.8: Uploading File Greater Than 100MB



Please select a file.

Fig 7.9: Pressing Upload Button Without Selecting Video

7.2 TEST CASE REPORT

Case id	Test Case Description	Expected Result	Actual Result	Status
1	Upload a word file instead of video	Error message: Only video files allowed	Error message: Only video files allowed	Pass
2	Upload a 200MB video file	Error message: Max limit 100MB	Error message: Max limit 100MB	Pass
3	Upload a file without any faces	Error message:No faces detected. Cannot process the video.	Error message:No faces detected. Cannot process the video.	Pass
4	Videos with many faces	Fake / Real	Fake	Pass
5	Deepfake video	Fake	Fake	Pass
6	Enter /predict in URL	Redirect to /upload	Redirect to /upload	Pass
7	Press upload button without selecting video	Alert message: Please select video	Alert message: Please select video	Pass
8	Upload a Real video	Real	Real	Pass
9	Upload a face cropped real video	Real	Real	Pass
10	Upload a face cropped fake video	Fake	Fake	Pass

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

In our project, we introduced a neural network-based approach aimed at classifying videos as either deep fakes or real, while also providing the confidence level associated with our model's predictions. Our methodology boasts the capability to process 1 second of video data, equivalent to 10 frames per second, with a commendable level of accuracy.

At the core of our approach lies the fusion of two powerful components: a pre-trained ResNext CNN model and an LSTM network. The ResNext CNN model effectively extracts frame-level features from the input video data, leveraging its depth and complexity to capture intricate patterns. Subsequently, the LSTM network facilitates temporal sequence processing, enabling our model to discern temporal changes between consecutive frames (t and t-1).

Notably, our model operates on a frame sequence of 20, allowing it to analyze video data comprehensively and make informed predictions. By combining the strengths of both the CNN and LSTM architectures, our model demonstrates robust performance in discriminating between real and deep fake videos.

In summary, our approach offers a sophisticated solution for video classification tasks, leveraging deep learning techniques to effectively identify deep fakes while providing confidence estimates for each prediction. With its ability to process video data at a granular level and achieve high accuracy, our model holds promise for various applications in detecting and combating the proliferation of deep fake content.

8.2 FUTURE SCOPE

- Increasing Sequence Length for Improved Accuracy: Exploring the impact of increasing the sequence length beyond 20 frames could potentially lead to significant improvements in accuracy. By incorporating a longer sequence of frames for analysis, the model can capture more extensive temporal dependencies and subtle nuances within the video data. Experimenting with longer sequence lengths and evaluating their effect on model performance could uncover meaningful insights and refine the algorithm's ability to discern between real and deep fake videos with greater precision.

- Upscaling to a Browser Plugin: Expanding the current web-based platform into a browser plugin could significantly enhance user accessibility and convenience. By integrating the deep fake detection functionality directly into web browsers, users can seamlessly verify the authenticity of videos encountered online without the need to navigate to a separate platform.
- Detection of Full Body Deep Fakes: While the current algorithm focuses on detecting face deep fakes, there is considerable potential for enhancement to encompass the detection of full-body deep fakes. By extending the algorithm's capabilities to analyse and identify manipulations beyond facial features, such as body movements and gestures, the system could offer more comprehensive protection against the proliferation of deceptive content across various video formats and scenarios.

CHAPTER 9

REFERENCE

[1] Deng Pan, Lixian Sun, Rui Wang, Xingjian Zhang, Richard O. Sinnott "Deepfake Detection through Deep Learning" 2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), DOI 10.1109/BDCAT50828.2020.00001

[2] Abhijit Jadhav, Abhishek Patange, Jay Patel, Hitendra Patil, Manjushri Mahajan "Deepfake Video Detection using Neural Networks" IJSRD - International Journal for Scientific Research & Development| Vol. 8, Issue 1, 2020 | ISSN (online): 2321-0613

[3] Rimsha Rafque, RahmaGantassi, RashidAmin, Jaroslav Frnda, Aida Mustapha, Asma HassanAlshehri "Deep fake detection and classifcation using error-level analysis and deep learning" https://doi.org/10.1038/s41598-023-34629-3

[4] Zeina Ayman, Natalie Sherif, Mariam Mohamed, Mohamed Hazem, Diaa Salama " DeepFakeDG: A Deep Learning Approach for Deep Fake Detection and Generation" Journal of Computing and Communication Vol.2 , No.2 , PP. 31-37 , 2023

[5] David Guera, Sriram Baireddy, Paolo Bestagini, Stefano Tubaro, Edward J. Delp "We Need No Pixels: Video Manipulation Detection Using Stream Descriptors" arXiv:1906.08743v1 [cs.LG] 20 Jun 2019 https://doi.org/10.1007/978-3-030-01228-1_8

[6] Hemal Mamtora, Kevin Doshi, Shreya Gokhale, Surekha Dholay, Chandrashekhar Gajbhiye "Video Manipulation Detection and Localization Using Deep Learning" 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) DOI: 10.1109/ICACCCN51052.2020.9362923

[7] Thanh Thi Nguyen, Tien Dung Nguyen, Cuong M. Nguyen, Saeid Nahavandi ”Deep Learning for Deepfakes Creation and Detection” https://www.researchgate.net/publication/336058980

[8] Da Wan, Manchun Cai, Shufan Peng, Wenkai Qin and Lanting Li "Deepfake Detection Algorithm Based on Dual-Branch Data Augmentation and Modified Attention Mechanism" Appl. Sci. 2023,13, 8313. https://doi.org/10.3390/app13148313

[9] Siwei lyu "Deepfake detection" 2022 H.T.Sencaretal.(eds.), MultimediaForensics, Advances in Computer Vision and Pattern, https://doi.org/10.1007/978-981-16-7621-5_12

[10] Luca Guarnera,Oliver Giudice,Sebastiano Battiato "DeepFake Detection by Analyzing Convolutional Traces" 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),DOI 10.1109/CVPRW50498.2020.00341

[11] D. Myvizhi1, J. C. Miraclin Joyce Pamila2 "Extensive Analysis of Deep Learning-based Deepfake Video Detection " 2022 Journal of Ubiquitous Computing and Communication Technologies, March 2022, Volume 4, Issue 1, Pages 1-8 DOI: https://doi.org/10.36548/jucct.2022.1.001

[12] Aarti Karandikar, Vedita Deshpande, Sanjana Singh, Sayali Nagbhidkar, Saurabh Agrawal " Deepfake Video Detection Using Convolutional Neural Network " Volume 9 No.2, March - April 2020 International Journal of Advanced Trends in Computer Science and Engineering, https://doi.org/10.30534/ijatcse/2020/62922020

[13] David Guera, Edward J. Delp " Deepfake Video Detection Using Recurrent Neural Networks" Deep feature interpolation for image content changes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6090–6099, July 2021

[14] Hafsa Ilyas, Aun Irtaza, Ali Javed, Khalid Mahmood Malik " Deepfakes Examiner: An End-to-End Deep Learning Model for Deepfakes Videos Detection" 2022 16th International Conference on Open Source Systems and Technologies (ICOSST) | 978-1-6654-6477-2/22/2022 IEEE | DOI: 10.1109/ICOSST57195.2022.10016871

[15] Daniel Mas Montserrat, Hanxiang Hao, S. K. Yarlagadda, Sriram Baireddy, Ruiting Shao, Janos Horv’ath, Emily Bartusiak, Justin Yang, David G ´ uera, Fengqing Zhu, Edward J. Delp "Deepfakes Detection with Automatic Face Weighting" 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)/DOI 10.1109/CVPRW50498.2020.00342