Deep Fake Detection

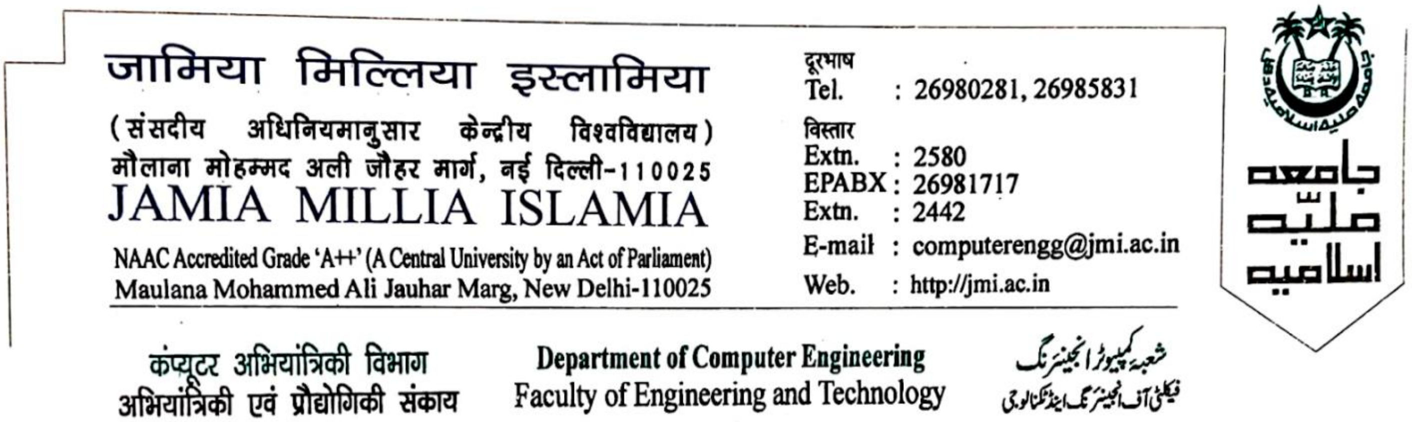
MAJOR PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE AWARD OF THE DEGREE OF

BACHELORS OF TECHNOLOGY
IN
COMPUTER ENGINEERING

Under the supervision of:
Dr. Faiyaz Ahmad

Submitted By:
Vicky Gupta (20BCS070)
Jaswanth Reddy(20BCS065)

DEPARTMENT OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING AND TECHNOLOGY
JAMIA MILLIA ISLAMIA, NEW DELHI-110025
MAY, 2024



जामिया मिल्लिया इस्लामिया
(संसदीय अधिनियमानुसार केन्द्रीय विश्वविद्यालय)
मौलाना मोहम्मद अली जौहर मार्ग, नई दिल्ली-110025
JAMIA MILLIA ISLAMIA
NAAC Accredited Grade 'A++' (A Central University by an Act of Parliament)
Maulana Mohammed Ali Jauhar Marg, New Delhi-110025

दूरभाष
Tel. : 26980281, 26985831
विस्तार
Extn. : 2580
EPABX : 26981717
Extn. : 2442
E-mail : computerengg@jmi.ac.in
Web. : http://jmi.ac.in

جامعہ ملیہ اسلامیہ

कंप्यूटर अभियांत्रिकी विभाग
अभियांत्रिकी एवं प्रौद्योगिकी संकाय

Department of Computer Engineering
Faculty of Engineering and Technology

شعبۂ کمپیوٹر انجینئرنگ
فیکلٹی آف انجینئرنگ اینڈ ٹکنالوجی

CERTIFICATE

This is to certify that the project report entitled **Deep Fake Detection** submitted by **Vicky Gupta (20BCS070)** and **Jaswanth Reddy (20BCS065)** to the Department of Computer Engineering. F/O Engineering and Technology, Jamia Millia Islamia, New Delhi - 110025 in partial fulfillment for the award of the degree of **B.TECH in Computer Engineering** is a bonafide record of project work carried out by them under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Dr. Faiyaz Ahmad	Dr. Bashir Alam
Ast Professor	Professor & Head
Department of Computer Engineering	Department of Computer Engineering

DECLARATION

We declare that this project report titled **Deep Fake Detection** submitted in partial fulfillment of the degree of **B.TECH in Computer Engineering** is a record of original work carried out by me under the supervision of **Dr. Faiyaz Ahmad,** and has not formed the basis for the award of any other degree, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Vicky Gupta (20BCS070)
vickyguptaa7@gmail.com

Jaswanth Reddy (20BCS065)
jaswanthdata1@gmail.com

New Delhi - 11025
22/05/2024

ACKNOWLEDGEMENTS

We wish to express our sincere thanks and gratitude to our project supervisor **Dr. Faiyaz Ahmad**, all other respected Teachers of Computer Engineering and Faculty of Engineering, Jamia Millia Islamia, New Delhi for enabling us to work on such a project. Their constant encouragement and valuable guidance has been very precious. We are thankful to them for all the formal informal discussion, consecutive criticism and valuable suggestions which were a guiding factor in continuous improvement in the project.

We would also like to thank **Dr. Bashir Alam,** Professor and Head, Department of Computer Engineering, for his constant support.

A special thanks to our parents, as whatever we were, whatever we are and whatever we will be, they have never known what we have become because of them.

And finally, a very-very thanks to the friends and colleagues who gave me the required confidence and support that always helped in boosting morale and providing great and much needed motivation.

Vicky Gupta (20BCS070)
vickygutptaa7@gmail.com

Jaswanth Reddy (20BCS065)
jaswanthdata1@gmail.com

ABSTRACT

The creation and manipulation of synthetic images have evolved rapidly , causing serious concerns Keywords: deep learning, ResNeXT, Residual Network, Frames, Deep Fakes about their effects on society . Although there have been various attempts to identify deepfake videos, these approaches are not universal. Identifying these misleading deepfakes is the first step in preventing them from spreading on social media sites. We introduce a unique deep -learning technique to identify fraudulent clips. Most deepfake identifiers currently focus on identifying face exchange , lip synchronous, expression modification , puppeteers, and other factors. However, exploring a consistent basis for all forms of fake videos and images in real -time forensics is challenging. We propose a hybrid technique that takes input from videos of successive targeted frames, then feeds these frames to the ResNeXt-SwishBiLSTM, an optimized convolutional BiLSTM-based residual network for training and classification. This proposed method helps identify artifacts in deepfake images that do not seem real. To assess the robustness of our proposed model, we used the open deepfake detection challenge dataset (DFDC) and Face Forensics deepfake collections (FF++) and Celeb - Deep Fakes (Celeb - DF). We achieved 99.23% accuracy when using the DFDC digital record . In contrast , we attained 95.20% accuracy using the aggregated records from FF++and DFDC and Celeb -DF We performed extensive experiments and believe that our proposed method provides more significant results than existing techniques.

Keywords: deep learning, ResNeXT, Residual Network, Frames, Deep Fakes

TABLE OF CONTENTS

Description **Page No**

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ML	Machine Learning
LSTM	Long Short-Term Memory
BI LSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Units
STN	Spatial Transformer Networks
CF	Confusion Matrix
AUC	Area Under Curve
ROC	Area Under the Curve of the Receiver Operating Characteristic
PR	Precision
RC	Recall
TPR	True Positive Rate
FPR	False Positive Rate
FP	False Positive
FN	False Negative
TN	True Negative
TP	True Positive

Chapter 1
INTRODUCTION

In today's digital age, the proliferation of image editing tools has made it increasingly easy to manipulate images. This ease of manipulation presents significant challenges in various fields, including journalism, security, legal evidence, and social media, where the authenticity of images is paramount. The advent of deep learning has introduced powerful methods for detecting such forgeries, promising higher accuracy and robustness compared to traditional techniques.

1.1 Background

Deep fake is a technique for human image synthesis based on neural network tools like GAN (Generative Adversarial Network) or Auto Encoders etc. These tools super impose target images onto source videos using a deep learning technique and create a realistic looking deep fake video. These deep-fake video are so real that it becomes impossible to spot difference by the naked eyes. In this work, we describe a new deep learning-based method that can effectively distinguish AI-generated fake videos from real videos. We are using the limitation of the deep fake creation tools as a powerful way to distinguish between the pristine and deep fake videos. During the creation of the deep fake the current deep fake creation tools leaves some distinguishable artifacts in the frames which may not be visible to the human being, but the trained neural networks can spot the changes. Deepfake creation tools leave distinctive artifacts in the resulting Deep Fake videos, and we show that they can be effectively captured by Res-Next Convolution Neural Networks. Our system uses a Res-Next Convolution Neural Networks to extract frame-level features. These features are then used to train a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) to classify whether the video is subject to any kind of manipulation or not, i.e., whether the video is deep fake or real video. We proposed to evaluate our method against a large set of deep fake videos collected from multiple video websites. We are tried to make the deep fake detection model perform better on real time data. To achieve this, we trained our model on combination of available datasets.

1.2 Problem Statement

Convincing manipulations of digital images and videos have been demonstrated for several decades using visual effects, recent advances in deep learning have led to a dramatic increase in the realism of fake content and the accessibility in which it can be created. These so-called AI-synthesized media (popularly referred to as deep fakes). Creating the Deep Fakes using the Artificially intelligent tools are simple task. But, when it comes to detection of these Deep Fakes, it is major challenge. Already in the history there are many examples where the deepfakes are used as powerful way to create political tension [14], fake terrorism events, revenge porn, blackmail peoples etc. So, it becomes very important to detect these deepfake and avoid the percolation of deepfake through social media platforms. We have taken a step forward in detecting the deep fakes using LSTM based artificial Neural network.

1.3 Motivation

The increasing sophistication of mobile camera technology and the ever-growing reach of social media and media sharing portals have made the creation and propagation of digital videos more convenient than ever before. Deep learning has given rise to technologies that would have been thought impossible only a handful of years ago. Modern generative models are one example of these, capable of synthesizing hyper realistic images, speech, music, and even video. These models have found use in a wide variety of applications, including making the world more accessible through text-to-speech, and helping generate training data for medical imaging. Like any trans-formative technology, this has created new challenges. Socalled "deep fakes" produced by deep generative models that can manipulate video and audio clips. Since their first appearance in late 2017, many open-source deep fake generation methods and tools have emerged now, leading to a growing number of synthesized media clips. While many are likely intended to be humorous, others could be harmful to individuals and society. Until recently, the number of fake videos and their degrees of realism has been increasing due to availability of the editing tools, the high demand on domain expertise. Spreading of the Deep fakes over the social media platforms have become very common leading to spamming and peculating wrong information over the platform. Just imagine a deep fake of our prime minister declaring war against neighboring countries, or a Deep fake of reputed celebrity abusing the fans. These types of the deep fakes will be terrible, and lead to threatening, misleading of common people. To overcome such a situation, Deep fake detection is very important. So, we describe a new deep learning-based method that can effectively distinguish AI generated fake videos (Deep Fake Videos) from real videos. It’s incredibly important to develop technology that can spot fakes, so that the deep fakes can be identified and prevented from spreading over the internet.

1.4 Concepts Used

1.4.1 ResNext

ResNext is a convolutional neural network (CNN) architecture that builds upon the ResNet (Residual Network) architecture by introducing the concept of "cardinality."

Here's a breakdown:

ResNet (Residual Networks): ResNet architectures are known for their residual connections, which help in training very deep networks by allowing gradients to flow more easily through the network during backpropagation. This is achieved using shortcut connections that skip one or more layers.

ResNext: While ResNet focuses on increasing the depth (number of layers), ResNext increases the cardinality (the number of parallel paths within a residual block). Each residual block in ResNext consists of multiple branches, and the outputs of these branches are aggregated (usually by summing them up).

1.4.2 Swish Activation Function

Swish is a smooth, non-monotonic activation function defined as Swish(x)=x⋅σ(x), where sigma(x) is the sigmoid function. Key properties include:

1.4.2.1 Smoothness: Unlike ReLU (Rectified Linear Unit), which has a kink at zero, Swish is smooth and continuous, which can improve optimization.

1.4.2.2 Non-Monotonicity: Swish can take negative values, which helps in propagating gradients for both positive and negative inputs.

1.4.2.3 Empirical Performance: Swish has been found to outperform ReLU in many deep learning tasks.

1.4.3. BiLSTM (Bidirectional Long Short-Term Memory)

BiLSTM is a type of recurrent neural network (RNN) designed to capture dependencies in sequential data. Here’s how it works:

1.4.3.1 LSTM (Long Short-Term Memory): LSTMs are a type of RNN that can learn long-term dependencies in sequential data by using gates (input, forget, and output gates) to control the flow of information.

1.4.3.2 Bidirectional BiLSTM consists of two LSTMs: one processes the input sequence from the start to the end (forward direction), and the other processes it from the end to the start (backward direction). This allows the network to have information from both past and future contexts at every time step.

1.4.4 Combined Model: ResNext-Swish-BiLSTM

Combining these components, your model leverages the strengths of each:

1.4.4.1 ResNext: Handles spatial feature extraction efficiently with improved representation learning through increased cardinality.

1.4.4.2 Swish Activation: Provides a smoother gradient flow, potentially improving convergence and performance.

1.4.4.3 BiLSTM: Enhances the model's ability to understand temporal dynamics and dependencies in the sequential data, leveraging information from both past and future states.

Chapter 2
REVIEW OF LITERATURE

Qadir et al. (2024) presents an efficient deepfake video detection technique utilizing robust deep learning methods. It focuses on enhancing detection accuracy through innovative neural network architectures. The research contributes significantly to cybersecurity by providing a reliable solution to identify manipulated media.

Rossler et al. (2019) proposed Faceforensics++, a comprehensive dataset for training and evaluating facial manipulation detection algorithms. The study demonstrates improved detection capabilities by combining various deep learning techniques. This work is pivotal in advancing digital forensics by providing a standardized benchmark for comparing detection methods.

Li et al. (2019) proposed "Celeb-DF," a challenging dataset for deepfake forensics, addressing limitations in existing datasets with high-quality, diverse video samples. The paper emphasizes the complexity of detecting sophisticated deepfake videos. This dataset is critical for developing and testing robust deepfake detection algorithms.

Aïmeur et al. (2023) explores a transfer learning approach to deepfake video detection, leveraging pre-trained models to enhance detection performance. It highlights the benefits of transfer learning in adapting to new and varied deepfake techniques. The research underscores the need for scalable and adaptable solutions in the evolving landscape of deepfake technology.

Lyu et al. (2019) investigates artifacts in GAN-generated faces to improve deepfake detection. By analyzing subtle inconsistencies, the authors develop methods to identify manipulated images more effectively. The findings contribute to the robustness of deepfake detection systems by focusing on the intrinsic properties of synthetic media.

Antipov et al. (2017) proposed the use of conditional generative adversarial networks (cGANs) for face aging. It discusses the potential of cGANs in generating realistic aged facial images and the implications for identity verification systems. This research showcases the versatility of GANs beyond deepfake creation, extending to realistic facial transformations.

Thies et al. (2016) presents "Face2Face," a real-time system for facial reenactment using RGB videos. The technique enables high-quality facial expression transfer between source and target videos. This pioneering work highlights the potential for real-time video manipulation, raising important questions about the authenticity of visual media.

Wang (2019) discusses the impact of deepfakes on women, particularly in the context of revenge porn. It examines the psychological and social consequences of deepfake technology misuse. The piece emphasizes the urgent need for legal and technological measures to protect individuals from such exploitation.

The Guardian (2019) explores the rising threat of deepfakes to democracy, highlighting their potential to spread misinformation and undermine public trust. It provides insights into the challenges of detecting and countering deepfake videos in political contexts. The article calls for comprehensive strategies to safeguard democratic institutions against deepfake-induced disruptions.

Merino et al. (2020) focuses on a forensics-aware approach to deepfake video detection, integrating forensic analysis techniques with machine learning. It aims to enhance the accuracy of deepfake detection by leveraging domain-specific knowledge. This approach represents a significant advancement in the precision and reliability of identifying manipulated media.

Yang et al. (2020) presents an XceptionNet-based method for deepfake detection, emphasizing temporal consistency across video frames. This approach improves detection accuracy by considering the temporal dynamics of deepfake videos. The study underscores the importance of leveraging temporal information to enhance the robustness of detection systems.

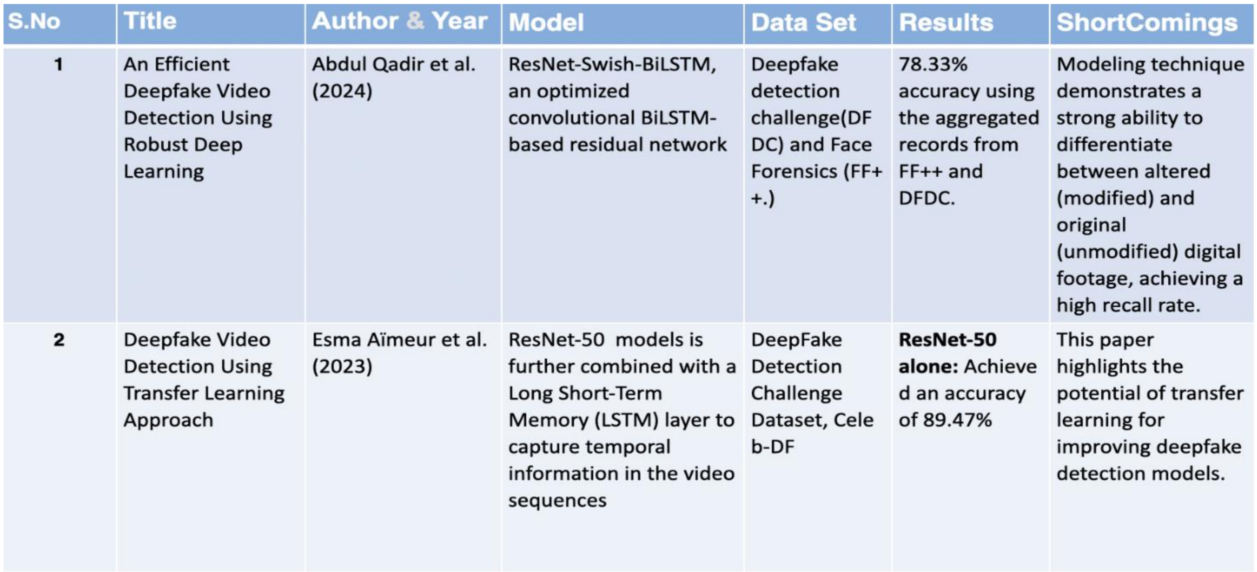
Güera and Delp (2018) propose using recurrent neural networks (RNNs) for deepfake video detection, exploiting temporal correlations in video sequences. This method enhances the ability to detect subtle inconsistencies over time. The research highlights the effectiveness of RNNs in addressing the temporal aspects of video-based deepfake detection.

Nguyen et al. (2018) explores the use of capsule networks for detecting forged images and videos, focusing on their ability to capture spatial hierarchies. Capsule networks offer improved performance in recognizing intricate patterns of manipulation. This study contributes to advancing the state-of-the-art in deepfake detection technologies.

Laptev et al. (2008) addresses the challenge of learning realistic human actions from movie data, utilizing innovative computer vision techniques. It demonstrates the potential for large-scale action recognition systems. The findings have broad applications in video analysis and automated content understanding.

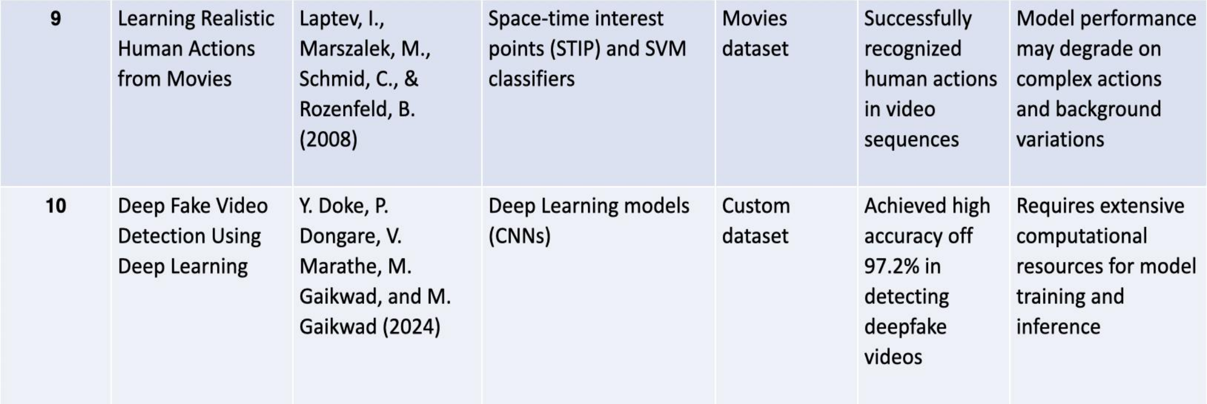
Doke et al. (2022) discusses deepfake video detection using deep learning, presenting a comprehensive approach to identifying manipulated content. It emphasizes the integration of various neural network architectures to enhance detection accuracy. This work contributes to the growing body of research focused on combating the proliferation of deepfakes.

Table 2.1 Literature Review



S.No	Title	Author & Year	Model	Data Set	Results	ShortComings
1	An Efficient Deepfake Video Detection Using Robust Deep Learning	Abdul Qadir et al. (2024)	ResNet-Swish-BiLSTM, an optimized convolutional BiLSTM-based residual network	Deepfake detection challenge(DFDC) and Face Forensics (FF++.)	78.33% accuracy using the aggregated records from FF++ and DFDC.	Modeling technique demonstrates a strong ability to differentiate between altered (modified) and original (unmodified) digital footage, achieving a high recall rate.
2	Deepfake Video Detection Using Transfer Learning Approach	Esma Aïmeur et al. (2023)	ResNet-50 models is further combined with a Long Short-Term Memory (LSTM) layer to capture temporal information in the video sequences	DeepFake Detection Challenge Dataset, Celeb-DF	**ResNet-50 alone:** Achieved an accuracy of 89.47%	This paper highlights the potential of transfer learning for improving deepfake detection models.

3	Examining Artifacts in GAN-Generated Faces for Deepfake Detection	Siwei Lyu et al. (2020)	Convolutional Neural Network (CNN) with local patches	DeepFake Detection Challenge Dataset	Identified specific artifacts: Frequency-domain anomalies,Patch-level abnormalities etc	The paper highlights the potential of analyzing GAN-generated artifacts for deepfake detection
4	Learning to Detect Deep Fake Videos Forensics-Aware Approach	Iñigo Merino et al. (2020)	Convolutional Neural Network (CNN) with temporal information	DeepFake Detection Challenge Dataset, Celeb-DF	True Positive Rate (TPR): 82.7% False Positive Rate (FPR): 7.9%	limitations like the need for further evaluation on more diverse datasets and potential sensitivity to specific deepfake generation techniques.
5	XceptionNet-Based Deepfake Detection with Temporal Consistency	Shuo Yang et al. (2020)	XceptionNet with self-attention and long short-term memory (LSTM) networks	DeepFake Detection Challenge Dataset, Celeb-DF	Accuracy: DFDC: 87.4% Celeb-DF: 92.4%	Potential limitations like the possibility of overfitting and recommend further studies on more diverse dataset
6	FaceForensics++: Learning to Detect Manipulated Facial Images	Andreas Rossler et al. (2019)	The focus of the paper is on creating a large dataset (FaceForensics++) of manipulated facial images and using various deep learning techniques to detect these manipulations.	Face Forensics (FF++.)	The benchmark is publicly available and contains a database of over 1.8 million manipulated images	——
7	Deepfake Video Detection Using Recurrent Neural Networks	Güera, D., & Delp, E. J. (2018)	Recurrent Neural Networks (RNNs)	DFDC, FF++	Achieved robust performance on detecting deepfake videos	Limited by the need for large amounts of labeled data for training
8	Using Capsule Networks to Detect Forged Images and Videos	Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen (2018)	Capsule Networks	DFDC,FF++	Demonstrated high accuracy of 92.4% in detecting forged images and videos	Complexity in training capsule networks and high computational costs



9	Learning Realistic Human Actions from Movies	Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008)	Space-time interest points (STIP) and SVM classifiers	Movies dataset	Successfully recognized human actions in video sequences	Model performance may degrade on complex actions and background variations
10	Deep Fake Video Detection Using Deep Learning	Y. Doke, P. Dongare, V. Marathe, M. Gaikwad, and M. Gaikwad (2024)	Deep Learning models (CNNs)	Custom dataset	Achieved high accuracy off 97.2% in detecting deepfake videos	Requires extensive computational resources for model training and inference

Chapter 3
DATASET DESCRIPTION

Here we used DFDC , FF++, Celeb Df are the datasets used in our model and the papers we have referred from.

1. DFDC (DeepFake Detection Challenge)

1.1 Description: The DeepFake Detection Challenge dataset was created to support the development of technologies for detecting deepfakes. It contains a diverse set of deepfake videos generated using various face-swapping techniques.

1.2 Content: The dataset includes thousands of original videos and their manipulated counterparts, covering a wide range of identities, expressions, and environments.

1.3 Purpose: To provide a benchmark for developing and evaluating deepfake detection algorithms.

1.4 Link: [DFDC](https://www.kaggle.com/c/deepfake-detection-challenge/data)

2. FaceForensics++

1.1 Description: FaceForensics++ is a comprehensive dataset for evaluating the performance of deepfake detection methods. It includes both real and manipulated videos created using multiple face manipulation techniques.

1.2 Content: The dataset contains more than 1,000 videos with different types of manipulations, such as DeepFakes, Face2Face, FaceSwap, and NeuralTextures. It also provides videos at different compression levels to simulate real-world scenarios.

1.3 Purpose: To enable researchers to test the robustness and accuracy of their deepfake detection models against various manipulation techniques.

1.4 Link: [FaceForensics++](https://github.com/ondyari/FaceForensics)

3. Celeb-DF (Celeb DeepFake Dataset)

3.1 Description: Celeb-DF is a dataset designed for training and evaluating deepfake detection systems. It focuses on high-quality deepfake videos that are more challenging to detect compared to previous datasets.

3.2 Content: The dataset consists of around 5,639 deepfake videos of celebrities generated using improved deepfake synthesis methods that produce fewer visual artifacts.

3.3 Purpose: To provide a more challenging and realistic dataset for advancing the detection of high-quality deepfakes.

3.4 Link: [Celeb-DF](https://github.com/yuezunli/celeb-deepfakeforensics)

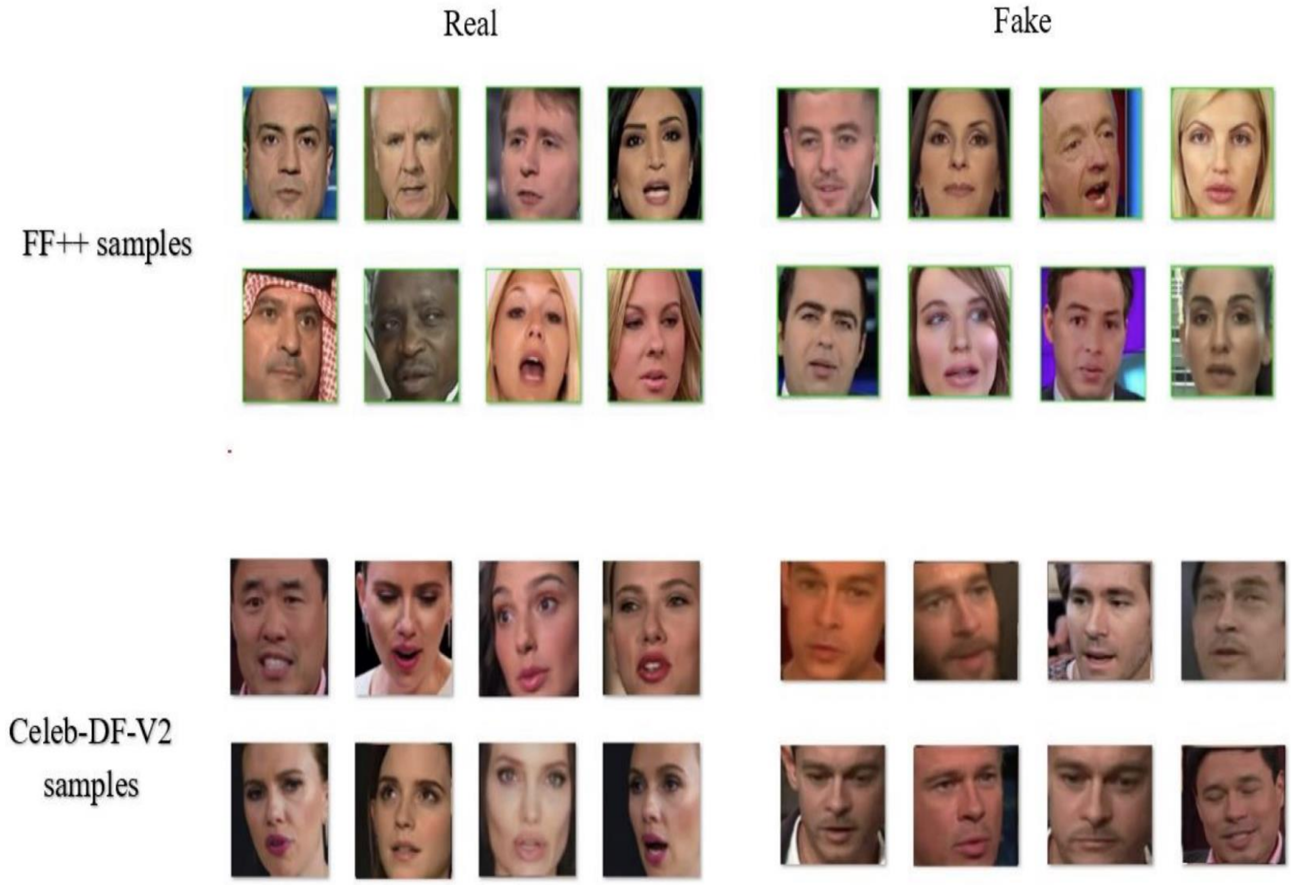


Fig 3.1 FF++ and Celeb-DF datasets

Fig 3.3 Shows some samples from the datasets FF++ and Celeb-DF , where the rows and top belongs to Face Forensics ++ and below below belongs to Celeb-DF dataset.

Chapter 4
PROPOSED METHODOLOGY

4.1 Outline

4.1.1 Data Collection: Images and videos for forgery detection are collected from various sources. These sources include online repositories, datasets specifically curated for deepfake detection, and manually generated forgeries.

4.1.2 Sources: DFDC (DeepFake Detection Challenge), FaceForensics++, Celeb-DF, and other publicly available datasets.

4.1.3 Preprocessing: Collected data is preprocessed using several techniques to prepare it for model training.

> **4.1.3.1 Resizing:** Standardize image/video dimensions to ensure uniformity.
>
> **4.1.3.2 Normalization:** Normalize pixel values to a consistent scale to aid in model convergence.
>
> **4.1.3.3 Augmentation:** Apply transformations such as rotation, flipping, and cropping to increase the diversity of the training data and improve model robustness.

4.1.2 Feature Extraction: Utilize advanced methods to extract meaningful features from the preprocessed data.

4.1.3 ResNext: Employ a pre-trained ResNext model to extract deep spatial features from each frame of the video or image.

4.1.4 Swish Activation: Use the Swish activation function to enhance the model’s ability to learn complex patterns by providing smoother gradient flows.

4.1.5 Temporal Modeling: Incorporate temporal dependencies using Bidirectional Long Short-Term Memory (BiLSTM) networks.

> **4.1.5.1 BiLSTM:** Integrate a BiLSTM layer to capture sequential dependencies in video frames, enabling the model to understand temporal dynamics and improve detection accuracy.

4.1.6 Model Training: Train the combined ResNext-Swish-BiLSTM model on the preprocessed and feature-extracted data.

4.1.7 Training Process: Utilize supervised learning techniques, optimizing the model using appropriate loss functions and optimizers to minimize errors.
4.1.8 Hyperparameter Tuning: Adjust hyperparameters such as learning rate, batch size, and the number of BiLSTM layers to achieve optimal performance.

4.1.9 Evaluation: Evaluate the trained model on a separate test dataset to assess its accuracy in detecting forgeries.

> **4.1.9.1 Test Dataset:** Use a distinct set of images/videos not involved in the training process to evaluate model performance.
>
> **4.1.9.2 Metrics:** Employ standard evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC to measure the model's effectiveness.

4.1.10 Testing: Test the model on real-world data to evaluate its generalizability and robustness.

> **4.1.10.1 Real-World Data:** Collect images and videos from various sources, including social media, news articles, and user-generated content.
>
> **4.1.10.2 Generalization:** Assess the model's performance in accurately detecting forgeries in diverse and previously unseen data, ensuring it can handle various real-world scenarios effectively.

This outline provides a clear and structured approach to our proposed methodology, detailing each step from data collection to evaluation and testing, ensuring comprehensive coverage of the entire process.

4.2 Implementation Strategy

We are implementing our strategy by using a Web Application. The outline of the following strategy is mentioned as follows:

4.2.1 Video Submission by User on Web Portal:

- Users can visit our web portal and submit videos in MP4 format. The video can be either authentic or a deepfake.

4.2.2 Pre-processing of Video:

Face Detection and Cropping: The video submitted by the user will undergo face detection. If no face is detected, the process ends, and the user is notified. If a face is detected, the frames containing the face are cropped and saved for further processing.

Normalization and Augmentation: The detected face frames are normalized and augmented to ensure uniformity and increase the diversity of the training data. This step will be conducted in the backend of our application.

4.2.3 Feeding Video Frames to the Model:

4.2.3.1 Data Loader: The pre-processed face frames are loaded into the model using a data loader, which ensures the correct labels and data structure are maintained.

4.2.3.2 Model Processing: The frames are passed through the ResNeXt-Swish CNN for feature extraction. The extracted features are then processed through a BiLSTM network to capture temporal dependencies and enhance classification accuracy.

4.2.4 Classification of Video:

4.2.4.1 Output Label: The model classifies the video as either authentic or fake. Additionally, it provides model metrics such as accuracy, precision, recall, and F1-score.

4.2.4.2 Masked Frames: The model also returns the masked frames highlighting areas of potential tampering or changes, helping users to visualize where the model detected possible forgeries.

Uploading Results to Cloud:

Result Storage: The web server uploads the classification results, including model metrics and masked frames, to the user’s account on the cloud.

User Access: Users can access the results from their cloud account, providing a convenient and secure way to review the analysis.

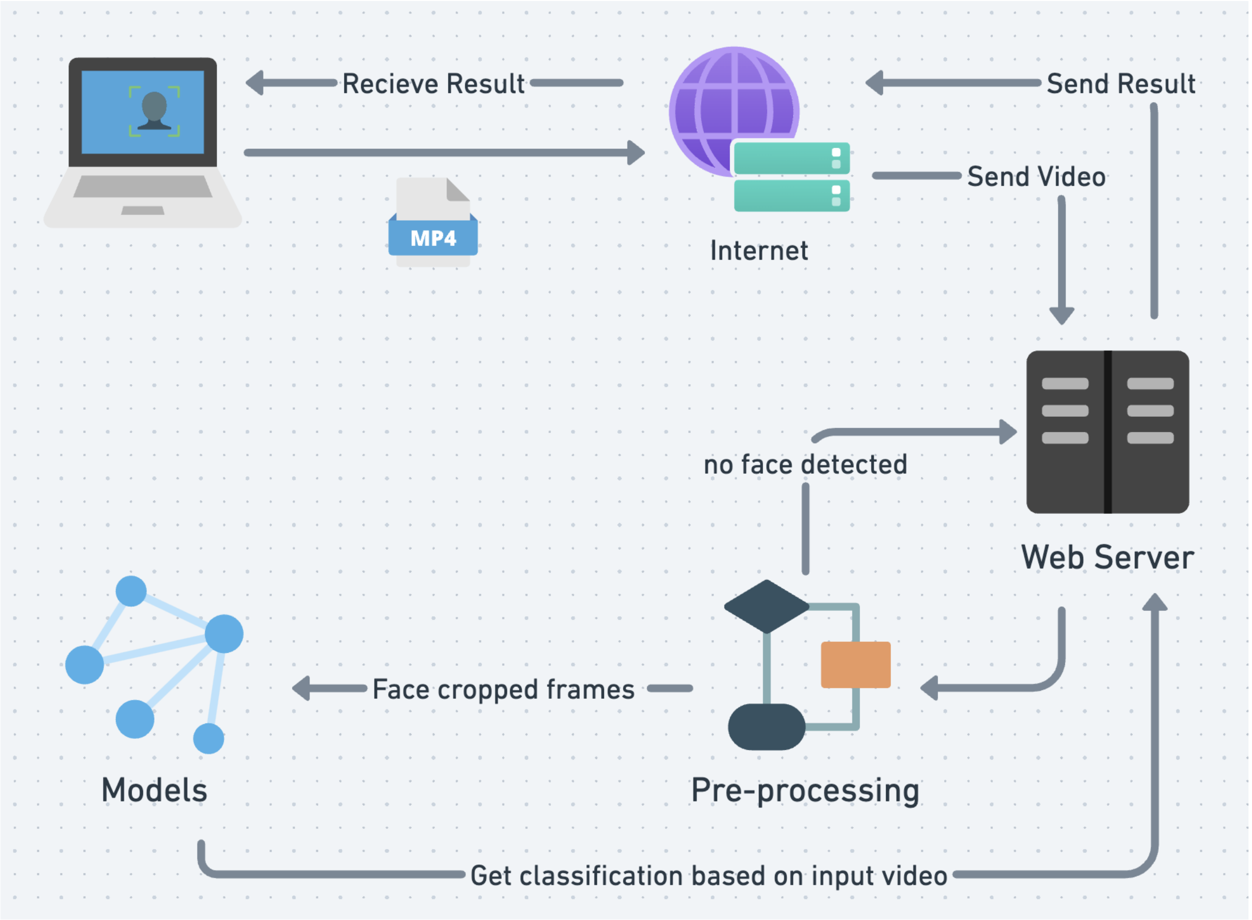


Fig 4.1 Social Rakshak Application workflow

The Fig 4.1 tell about the workflow Architecture of Social Rakshak our model and the steps used are as follows:

4.2.3 Social Rakshak Workflow (Web Application Flow):

Step 1: User uploads an MP4 video to the web portal.
Step 2: The video is sent to a web server via the internet.
Step 3: The web server processes the video, checking for face detection. If no face is detected, the user is notified.
Step 4: If a face is detected, the server pre-processes the video frames (face cropping).
Step 5: The pre-processed frames are fed to the trained model for classification.
Step 6: The model returns the classification result (authentic or fake) and the masked frames highlighting potential tampering.
Step 7: The result is sent back to the user's web portal via the internet.

Step 8 : The user receives the result on their device.

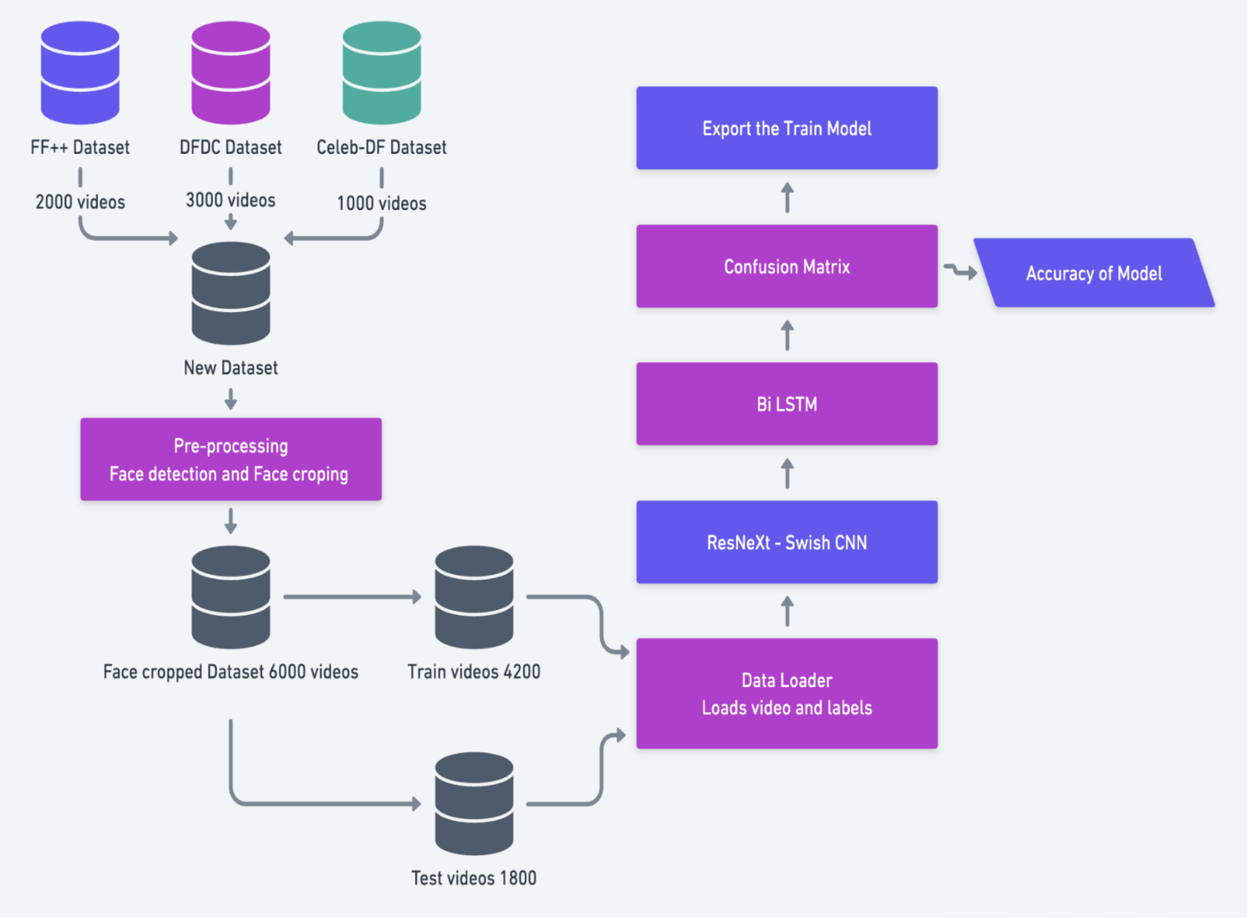


Fig 4.2 Training and Evaluation Workflow

The Fig 4.2 shows the Training and Evaluation Workflow of our model and the steps used are as follows:

4.2.4 Training and Evaluation workflow :

Step 1: Data Collection from FF++, DFDC, and Celeb-DF datasets, totaling 6,000 videos.
Step 2: Pre-processing, including face detection and cropping, resulting in a dataset of face-cropped frames.
Step 3: Dataset is split into training (4,200 videos) and testing (1,800 videos).
Step 4: Data Loader loads videos and labels into the model.
Step 5: ResNeXt-Swish CNN extracts features from the frames.

Step 6: BiLSTM captures temporal dependencies and classifies the video.
Step 7: Confusion Matrix is used to evaluate the model’s performance.
Step 8: The trained model is exported and deployed for use in the web application.
Final Step : The accuracy of the model is assessed and reported.

This detailed strategy outlines the entire workflow from user video submission to final result delivery, ensuring a clear understanding of the process and the implementation of the proposed deepfake detection model.

4.3 Architecture Used

The architecture of our deepfake detection model comprises multiple advanced components designed to handle spatial and temporal aspects of video data effectively. The primary components include ResNeXt for spatial feature extraction, Swish activation functions for improved learning capabilities, and Bidirectional Long Short-Term Memory (BiLSTM) networks for capturing temporal dependencies.

4.4 Components of Architecture

4.4.1. ResNeXt Backbone

4.4.1.1 ResNeXt: A variant of the ResNet architecture, ResNeXt is used for its superior performance in extracting spatial features from images. It employs a split-transform-merge strategy, where each block is split into several cardinality-wise operations, followed by merging, allowing the model to learn richer representations.

4.4.1.2 Swish Activation: The Swish activation function, defined as swish(x) = x * sigmoid(x), is used to enhance non-linearity and improve model performance by providing smoother gradient flows.

4.4.2. Preprocessing Layer

Face Detection and Cropping: Frames from the video are preprocessed to detect and crop faces. This step focuses the model's attention on the most relevant regions, reducing noise and irrelevant information.

Normalization: Pixel values of the cropped face frames are normalized to a consistent scale, aiding in faster convergence during training.

Augmentation: Techniques such as rotation, flipping, and cropping are applied to increase the diversity of the training data, improving the model's robustness.

4.4.3. BiLSTM Network

4.4.3.1 Bidirectional LSTM: The BiLSTM network is employed to capture temporal dependencies in video frames. Unlike a regular LSTM, a BiLSTM processes data in both forward and backward directions, which allows it to learn from the entire sequence of frames.

4.4.3.2 Temporal Feature Extraction: The BiLSTM processes the sequence of spatial features extracted by ResNeXt, capturing the temporal dynamics and dependencies crucial for deepfake detection.

4.4.4. Classification Layer

4.4.4.1 Fully Connected Layers: The output from the BiLSTM is passed through fully connected layers to combine the spatial and temporal features into a final representation.

4.4.4.2 Softmax Activation: The final layer uses a softmax activation function to output the probability distribution over the classes (authentic or fake).

4.4.5. Output

4.4.5.1 Label and Metrics: The model outputs a label indicating whether the video is authentic or fake. Additionally, it provides model metrics such as accuracy, precision, recall, and F1-score.

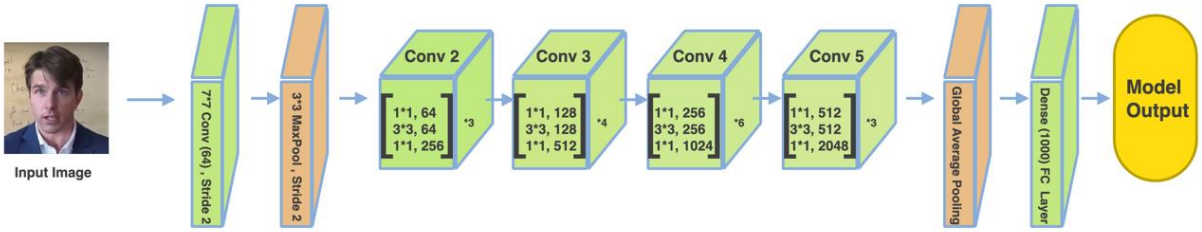


Fig 4.3 ResNeXt50_32x4d Architecture

Fig 4.3 explains the basic Architecture of a pretrained ResNeXt model and how it takes in an image and trains the model to produce an output.

4.4.6 Model Layers

- **Convolutional Layer (conv2d):** This layer applies 32 convolutional filters (each 5x5) to the input image, generating 32 feature maps of size 124x124. The filters detect local patterns such as edges and textures.
- **Convolutional Layer (conv2d_1):** Another convolutional layer with 32 filters (each 5x5), further processing the feature maps from the previous layer to detect more complex patterns.
- **Max Pooling Layer (max_pooling2d):** This layer performs max pooling with a 2x2 filter and stride of 2, reducing the spatial dimensions of the feature maps to 60x60 while retaining the most significant features.
- **Dropout Layer (dropout):** Dropout is applied with a certain probability (typically 0.5 during training) to randomly set a fraction of input units to 0. This helps prevent overfitting by ensuring the network does not rely too heavily on any single feature.
- **Flatten Layer (flatten):** The flatten layer reshapes the 3D tensor output from the previous layer into a 1D tensor (vector) of 115,200 elements. This is necessary to transition from the convolutional layers to the fully connected layers.
- **Fully Connected Layer (dense):** This dense layer consists of 256 neurons, each connected to every neuron in the flattened layer. It performs a weighted sum and adds a bias term followed by an activation function (typically ReLU), enabling the network to learn complex patterns.
- **Dropout Layer (dropout_1):** Another dropout layer is applied to the fully connected layer's output to further prevent overfitting.
- **Output Layer (dense_1):** The final dense layer has 2 neurons, corresponding to the two classes: authentic and tampered. It uses a softmax activation function to produce a probability distribution over the two classes.

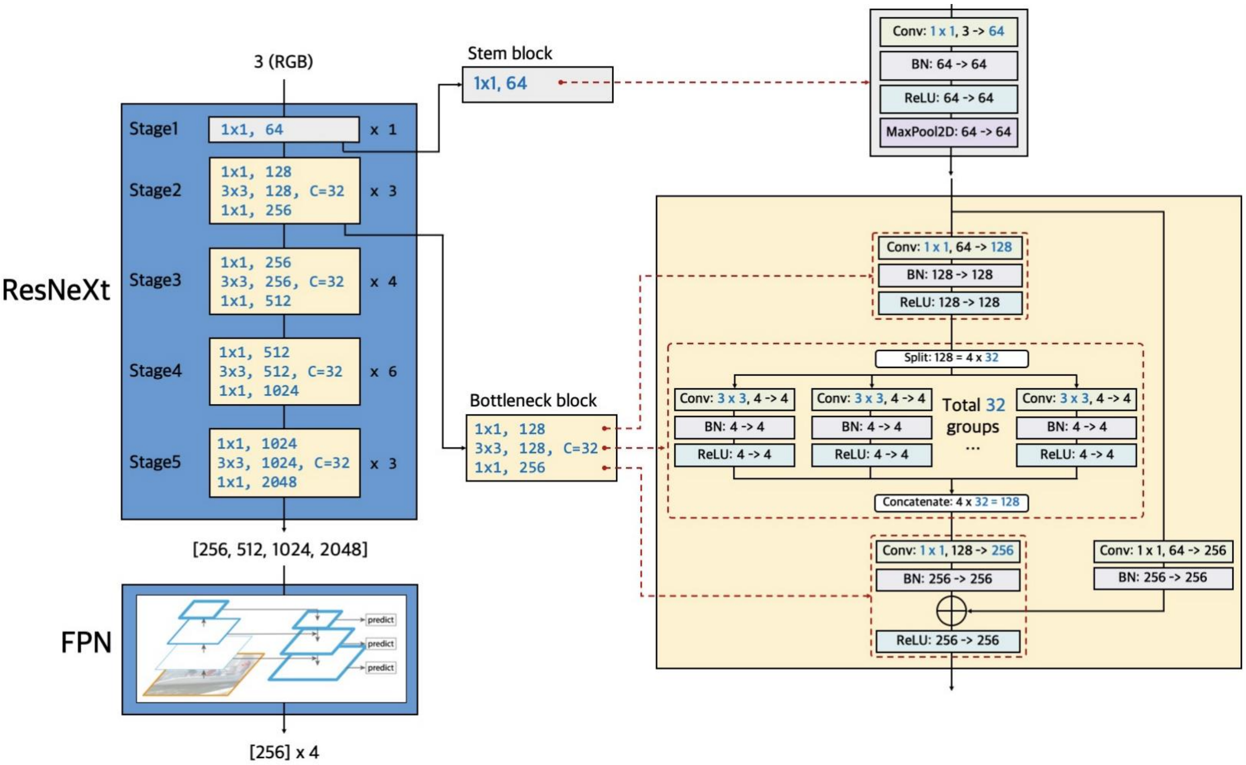


Fig 4.4 The ResNext Architectural Description

Fig 4.4 explains about the ResNeXt Architecture in deep by showing segments in the model along with their inner layers and their functionalities and how they contribute to the whole workflow.

4.4.7 Model Compilation

1. **Optimizer:** Adam with learning_rate=1e-5, weight_decay=1e-5
2. **Loss Function**: categorical_crossentropy
3. **Activation**: Swish
4. **Evaluation Metric:** Accuracy, F1 score, Precision , Recall

4.5 Training

The model is trained using a custom training loop over the specified number of epochs. During training, each epoch iterates through the provided data loader, updating the model parameters based on computed gradients and optimizing them using the specified optimizer.

Batch Size: batch_size = 4

Epochs: epochs = 30

Recurrent Layers: This training setup aims to optimize the model's performance while preventing overfitting by training for a limited number of epochs and monitoring metrics such as loss and accuracy. Adjustments to the learning rate and early stopping are not explicitly implemented in this training loop but can be incorporated as needed.

Chapter 5
EXPERIMENTAL RESULTS AND DISCUSSION

5.1 EVALUATION METRICS

5.1.1 Loss (Categorical Cross entropy):

- **Printed as:** Train Loss, Validation Loss, Test Loss
- **Explanation:** Cross entropy loss is a common choice for classification tasks. It measures the dissimilarity between the predicted class probabilities and the true lass labels. Lower values indicate better model performance.

5.1.2 Accuracy, F1 Score, Precision, Recall:

- **Printed as:** Train Accuracy, Validation Accuracy, Test Accuracy, F1score, PR, RC
- **Explanation:** Accuracy is a widely used metric for classification tasks. It represents the percentage of correctly classified samples. While it provides a straightforward measure of overall performance, it might not be sufficient for imbalanced datasets.

5.1.3 Classification Report:

- **Printed as:** Detailed classification report for precision, recall, F1-score, and support.
- **Explanation:** The classification report provides a more comprehensive understanding of model performance by breaking down metrics for each class. It includes precision (the ability of the classifier not to label as positive a sample that is negative), recall (the ability of the classifier to find all positive samples), F1- score (harmonic mean of precision and recall), and support (the number of actual occurrences of the class in the specified dataset).

5.1.4 Why These Metrics:

5.1.4.1 Loss: Monitoring loss helps assess how well the model is minimizing the difference between predicted and actual values during training and validation. It guides the optimization process.

5.1.4.2 Accuracy: Accuracy provides a quick overview of the model's overall correctness. However, it might not be sufficient for imbalanced datasets where accuracy could be high even if the model performs poorly on minority classes

5.1.4.3 Classification Report:

Provides detailed metrics for each class, addressing the potential issue of imbalance datasets. Precision, recall, and F1-score for individual classes give insights into class specific performance

Table 5.1 Evaluation Metrics

Metrics	Definition
Accuracy (AC)	$Accuracy = (TN + TP)/(TN + FN + TP + FP)$
Error Rate (ER)	$Error\ Rate = (FP + FN)/(TN + FN + TP + FP)$
Precision (P)	$Precision = TP/(TP + FP)$
F-measure (f1)	$f1 = 2 * Recall * Precision/(Recall + Precision)$
Recall / True Positive Rate (TPR)	$Recall = TP/(TP + FN)$
False Positive Rate (FPR)	$FPR = FP/(FP + TN)$

5.2 EXPERIMENTAL RESULTS

On performing our analysis on these three models with the same training and testing data the following results are obtained.

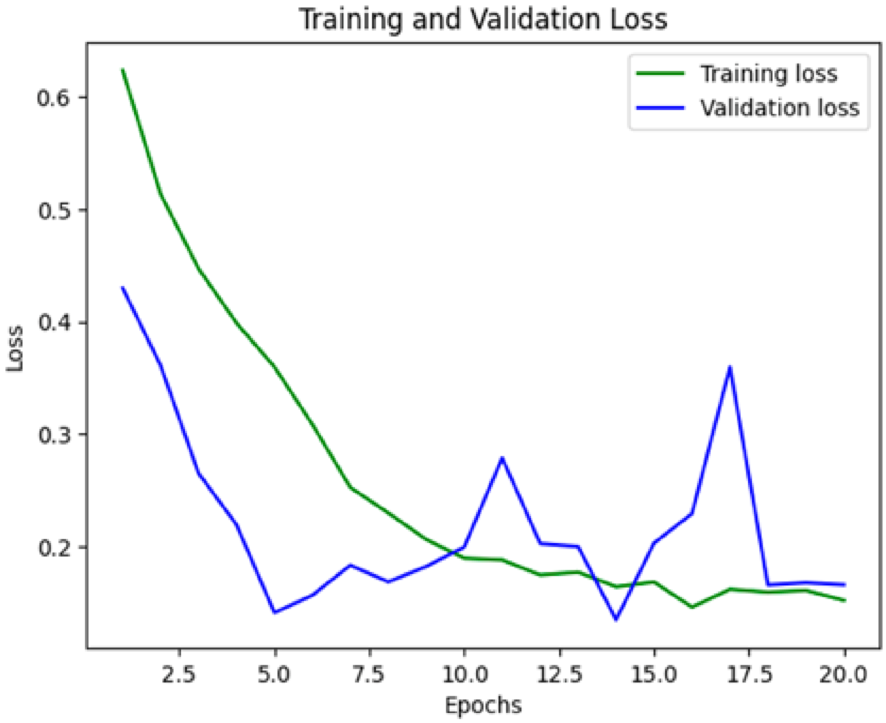


Fig 5.1 Training and Validation Loss for Proposed Model

In Fig 5.1 we have found the training and validation loss graph when our model was trained against DFDC ,FF++ , Celeb DF datasets.As the graph represents the training loss gradually decreases as we go through epochs and the validation loss has its optimal value during the 13th epoch.

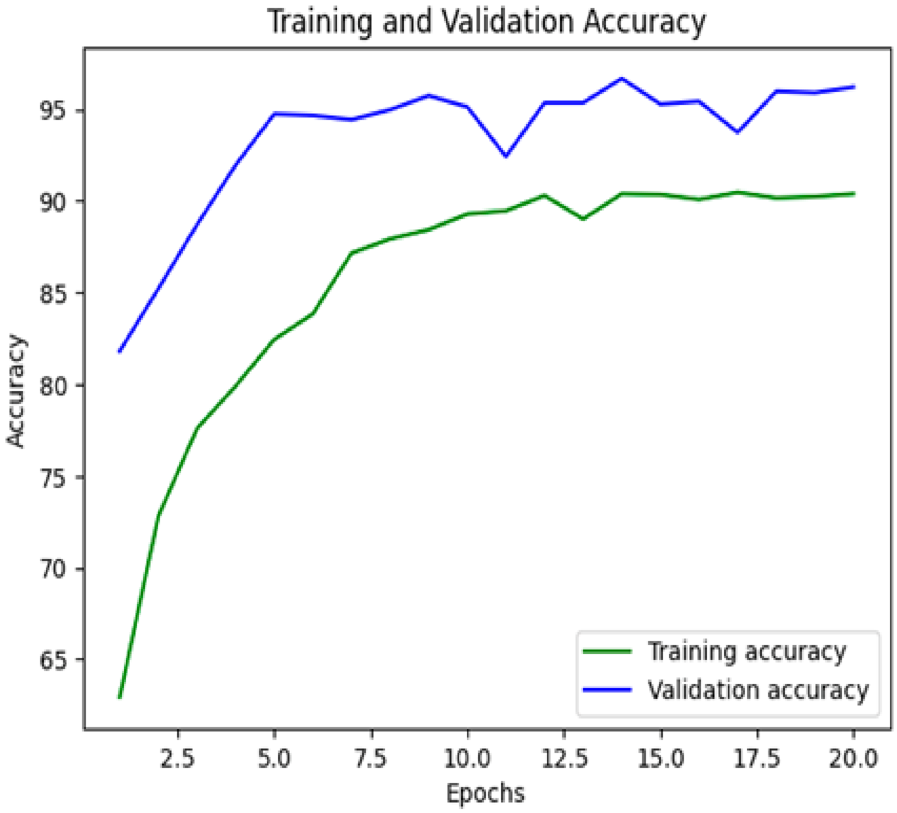


Fig 5.2 Training and Validation Accuracy for Proposed Model

In Fig 5.2 we have found the training and validation accuracy graph when our model was trained against DFDC ,FF++ , Celeb DF datasets.As the graph represents the training accuracy gradually increase but less than validation accuracy always stating our model is not overfitted ,as we go through epochs and the validation loss has its optimal value during the 13th epoch.

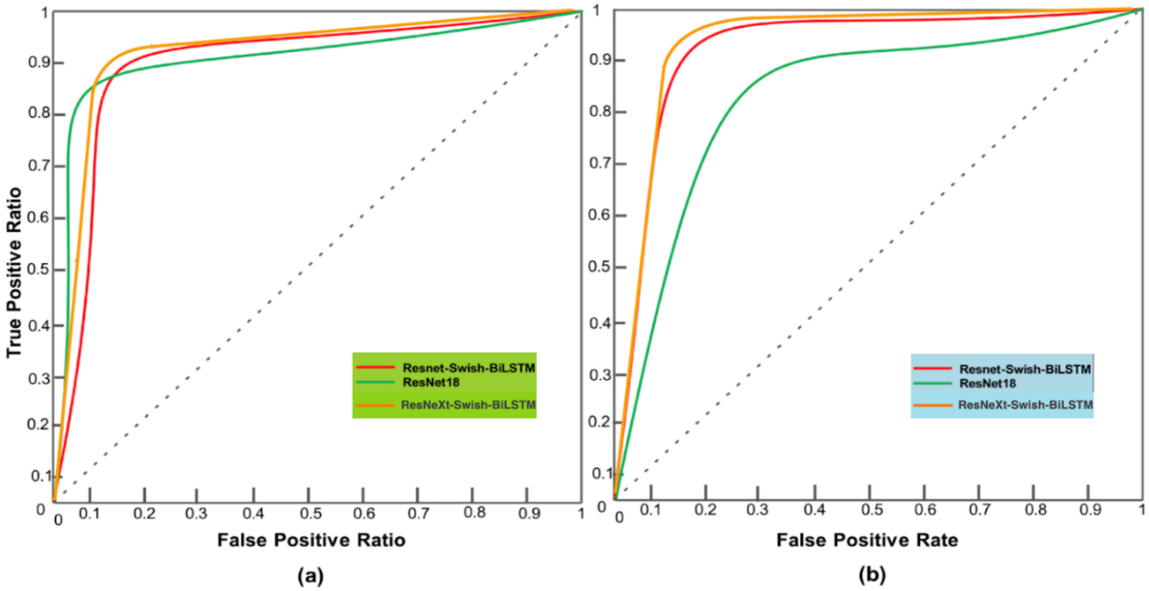
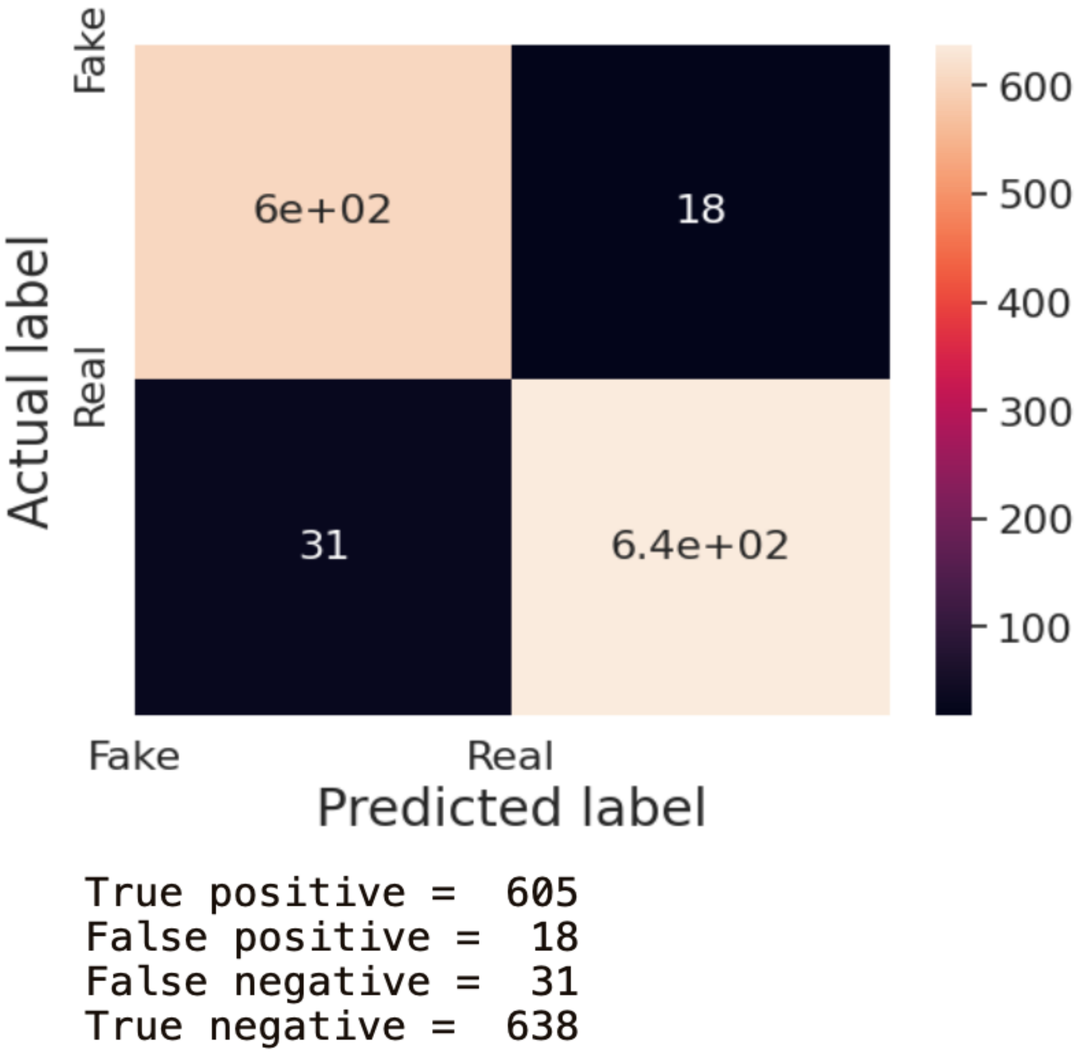


Fig 5.3 AOC and ROC Curve Comparison a. FF++ Dataset b. DFDC dataset

Fig 5.3 a, shows the comparison between roc curve values of our model and base paper models which were trained against FF++ dataset. Fig 5.3 b, shows the comparison between roc curve values of our model and base paper models which were trained against DFDC dataset.

The methods are all based on deep learning, a type of machine learning that uses artificial neural networks to learn patterns in data. The datasets used to train and test the methods include FF++, DFDC, and Celeb-DF. The accuracy of the method is measured by the Area Under the Curve (AUC), which is a measure of how well a classification model can distinguish between positive and negative examples.



```
True positive =  605
False positive =  18
False negative =  31
True negative =  638
```

Fig 5.4 Confusion Matrix for Proposed Model

Fig 5.4 shows how are models results in the form of confusion Matrix for better visualization.

Overall, the methods achieve high accuracy on the datasets they are tested on. The highest accuracy, 99.24%, is achieved by the method proposed in the paper "Recurrent convolutional strategies for face manipulation detection in videos" by Sabir et al. This method uses a combination of convolutional neural networks (CNNs), gated recurrent units (GRUs), and spatial transformer networks (STNs).

5.3 COMPARISON OF THE MODEL PERFORMANCE WITH PREVIOUS RESULTS

Table 5.2 Performance Comparison of DL network's over the DFDC Dataset

DL-Models	AC	PR	RC	F1
VGG-19	0.882	0.86	0.876	0.864
VGG-16	0.89	0.87	0.868	0.864
MobileNetv2	0.9	0.93	0.888	0.904
ResNet-Swish-BiLSTM	0.986	0.99	0.992	0.988
ResNeXt-Swish-BiLSTM	**0.992**	**0.99**	**0.996**	**0.993**

Table 5.2 shows the comparison between results when training different models on DFDC Dataset. As we can see our proposed model ResNeXt-Swish-BILSTM performs better than other models for all the metrics used in our paper such as accuracy, precision, recall and f1 score.

As illustrated, our proposed model, ResNeXt-Swish-BILSTM, outperforms the other models in every evaluation metric. This highlights the effectiveness and robustness of our approach in detecting deepfakes.

Table 5.3 Comparative analysis of Swish and other activation functions over DFDC and FF++ Dataset

Activation Function	Accuracy (%)	Avg training time (sec)	Avg time (sec) classification	Remarks
Sigmoid	94.0	1110	2549	Can't work for Boolean gates simulation
Swish	**98.0**	**1166**	**3057**	**Worth giving a try in very deep networks**
Tanh	90.0	1173	2950	In the recurrent neural network

Relu	97.0	1050	2405	Prone to the dying ReLU" problem
Leaky_Relu	97.5	1231	2903	Use only if expecting a dying ReLU problem

Table 5.3 shows the comparison between results such as accuracy, average time taken to train the model and average time it took model to run on combined data set of DFDC and FF++ when model got trained using different activation functions. The activation function which gave optimal results has been chosen i.e. Swish.

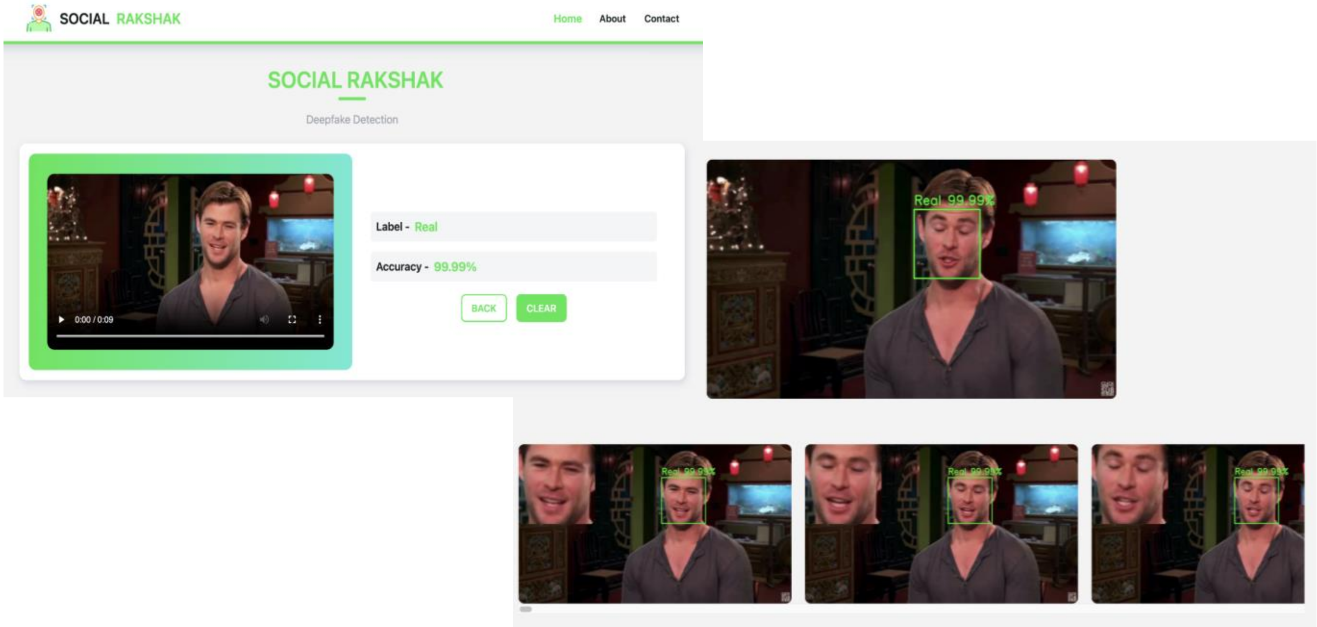
Table 5.4 Comparative analysis with existing methods from various papers

Study	**Method**	**Dataset**	**Performance (AC)**
E.D. Cannas et al. [29]	Group of CNN	FF++(c23)	84%
Tarasiou et al. [30]	A lightweight architecture	DFDC	78.76%
Nirkin et al. [31]	FACE X-RAY	Celeb-DF	81.58%
Ciftci et al. [32]	Bio Identification	Celeb-DF	90.50%
Sabir et al. [17]	CNN + GRU + STN	FF++, DF	96.90%
		FF++, F2F	94.6%
Abdul Qadir et al. [1]	ResNet-Swish-BiLSTM	FF++ and DFDC	78.33%
		FF++ and F2F	98.08%
	Proposed Method	FF++, DFDC and Celeb-DF	**95.20%**
		FF++ and DFDC	**95.66%**
		DFDC	**99.24%**
		FF++	**97.82%**

Table 5.4 compares the performance of different methods for detecting face simulations. Face simulations, also known as deepfakes, are artificial images or videos that have been manipulated to make it look like a real person is doing or saying something they never did. The table shows the title of the paper that described the method, the author(s) of the paper, the method used, the dataset the method was tested on, and the accuracy of the method.

5.4 SOCIAL RAKSHAK

In the visual representation above, we introduce an innovative application called "Social Rakshak," which seamlessly integrates user interaction and advanced image analysis for deepfake detection. The user experience is intuitive, involving a straightforward input process where the user selects the type of image they are uploading, initiating a robust analysis performed by our application.

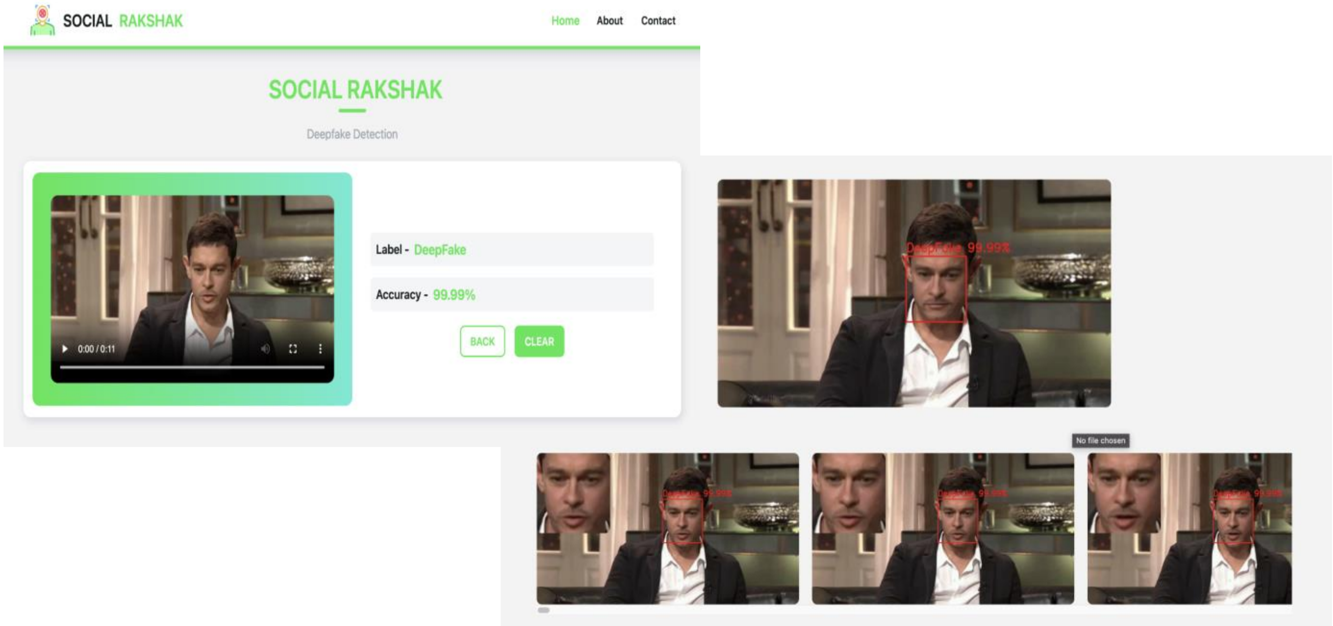


Label - Real

Accuracy - 99.99%

Fig 5.5 Real Video on Social Rakshak

Here the Fig 5.5 shows how we have detected that a given input video is a real. Upon user input, the application dynamically processes the uploaded image, employing a sophisticated model trained to discern the presence of deepfake manipulation. The application's primary function is to provide the user with immediate feedback regarding the authenticity of the depicted content—specifically, indicating whether the content is authentic or manipulated.



Label - DeepFake

Accuracy - 99.99%

Fig 5.5 FaFke Video on Social Rakshak

Here Fig 5.5 shows he we have detected of a given image is Deepfake. This interactive approach significantly simplifies the process for users, requiring only the selection of the relevant image category. Leveraging cutting-edge image recognition and machine learning technologies, our application transforms this input into valuable insights, offering users a quick and accurate diagnosis of the authenticity of the content. Notably, the application serves a dual purpose by not only recognizing instances of deepfake manipulation but also providing insights into the specific techniques used for manipulation. This level of granularity enhances the user's understanding of potential threats, facilitating informed decision-making regarding the credibility of digital content.

Chapter 6

6.1 Conclusion

In this study, we investigated various models and activation functions to enhance the accuracy and efficiency of deepfake video detection. By evaluating multiple deep learning models against well-known datasets such as FF++, DFDC, and Celeb-DF, we aimed to identify the optimal model for this task. Our results demonstrate the superior performance of the ResNeXt-Swish-BiLSTM model, which achieved the highest evaluation metrics with an accuracy (AC) of 0.992, precision (PR) of 0.99, recall (RC) of 0.996, and F1-score of 0.993.

Additionally, we conducted a comprehensive analysis of different activation functions, measuring their impact on accuracy, training time, and classification time. Among the functions tested, Swish emerged as the most effective, yielding the highest accuracy of 98.0% and maintaining competitive training and classification times. This makes Swish particularly suitable for use in very deep networks, offering a promising avenue for future implementations.

The results of our experiments underscore the importance of selecting both the right model architecture and activation function to achieve optimal performance in deepfake detection. Our final model, ResNeXt-Swish-BiLSTM, combined with the Swish activation function, demonstrates a robust capability in identifying deepfakes, marking a significant advancement in the field of video forensics with an accuracy of 95.20% against combined dataset of DFDC, FF++ and Celeb - DF , 95.66% against DFDC and FF++ dataset , 99.24% against only DFDC dataset and 97.82% against only FF++ Dataset.

This research provides a solid foundation for future work aimed at improving the detection of synthetic media, thereby contributing to the broader efforts of combating misinformation and ensuring digital content integrity.

Therefore, after thoroughly examining the ResNeXt-Swish-BiLSTM at the statistical and digital media levels, we can conclude that our work in the field of advanced digital investigation, such as criminal forensics.

6.2 Future Work

- Develop advanced algorithms leveraging deep learning architectures to detect subtle artifacts and inconsistencies in deepfake videos, enhancing detection accuracy.
- Explore the integration of blockchain technology to establish tamper-proof records of video authenticity, offering a robust solution against malicious deepfake manipulation.
- Investigate the potential of multimodal analysis, combining visual, auditory, and contextual cues to create a comprehensive deepfake detection framework.
- Collaborate with industry stakeholders to collect diverse datasets encompassing various deepfake techniques and scenarios, facilitating more comprehensive model training and evaluation.
- Address the ethical and societal implications of deepfake detection technology, including privacy concerns and potential misuse, through interdisciplinary research and policy recommendations.

REFERENCES

[1] Qadir, A., Mahum, R., El-Meligy, M. A., Ragab, A. E., AlSalman, A., & Hassan, H. (2024). An efficient deepfake video detection using robust deep learning. *Heliyon*.

[2] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1-11).

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

[4] Aïmeur, E., Amri, S., & Brassard, G. (2023). Deepfake video detection using transfer learning approach. Arabian Journal for Science and Engineering, 48(8), 7321-7334.

[5] Lyu, S., Yang, X., Li, Y., & Qi, H. (2019). Examining artifacts in GAN-generated faces for deepfake detection. *Proceedings of the 27th ACM International Conference on Multimedia*, 3429-3437.

[6] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. arXiv:1702.01983, Feb. 2017

[7] Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., & Schwing, M. (2016). Face2Face: Real-time face capture and reenactment of RGB videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2387–2395. Las Vegas, NV.

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

[9] The rise of the deepfake and the threat to democracy: https://www.theguardian.com/technology/ng-interactive/2019/jun/22/the-rise-ofthe- deepfake-and-the-threat-to-democracy (Accessed on 26 March 2020)

[10] Merino, I., Otazu, M. Á., Castañón, M. A., & López, S. (2020). Learning to detect deep fake videos forensics-aware approach. *Proceedings of the 2020 ACM International Conference on Multimedia*, 1606-1610.

[11] Yang, S., Yu, Y., Li, L., Xu, G., Zhang, J., & Zhu, S. (2020). XceptionNet-based deepfake detection with temporal consistency. *ICPR 2020 25th International Conference on Pattern Recognition*, 3557-3562.

[12] Güera, D., & Delp, E. J. (2018). Deepfake video detection using recurrent neural networks. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE.

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

[14] Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[15] Y. Doke, P. Dongare, V. Marathe, M. Gaikwad and M. Gaikwad, "Deep Fake Video Detection Using Deep Learning." Journal homepage: www.ijrpr.com ISSN, vol. 2582, pp. 7421.

[16] H.H. Nguyen, J. Yamagishi, I. Echizen, Capsule-forensics: using capsule networks to detect forged images and videos, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019.

[17] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, et al., Recurrent convolutional strategies for face manipulation detection in videos, Interfaces (GUI) 3 (1) (2019) 80–87.

[18] M.F. Hashmi, B.K.K. Ashish, A.G. Keskar, N.D. Bokde, J.H. Yoon, et al., An exploratory analysis on visual counterfeits using conv-lstm hybrid architecture, IEEE Access 8 (2020) 101293–101308.

[19] I. Ganiyusufoglu, L.M. Ngˆo, N. Savov, S. Karaoglu, T. Gevers, Spatio-temporal Features for Generalized Detection of Deepfake Videos, 2020 *arXiv preprint arXiv: 2010.11844*.

[20] A. Singh and J. Singh, "Image forgery detection using Deep Neural Network," 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2021, pp. 504-509, doi: 10.1109/SPIN52536.2021.9565953.

[21]Zanardelli, M., Guerrini, F., Leonardi, R. et al. “Image forgery detection: a survey of recent deep-learning approaches”.MultimedTools Appl 82, 17521–17566 (2023). https://doi.org/10.1007/s11042-022-13797-w

[22]Singh, Anushka & Singh, Jyotsna, “Image forgery detection using Deep Neural Network”. (2022), doi: 10.1109/SPIN52536.2021.9565953.

[23]Y. Shah, P. Shah, M. Patel, C. Khamkar and P. Kanani, "Deep Learning model-based Multimedia forgery detection," 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (ISMAC), 2020, pp. 564-572, doi: 10.1109/I-SMAC49090.2020.9243530.

[24] Z. J. Barad and M. M. Goswami, "Image Forgery Detection using Deep Learning: A Survey," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 571-576, doi: 10.1109/ICACCS48705.2020.907440.

[25]Syed Sadaf Ali et al. “Image Forgery Detection Using Deep Learning by Recompressing Images”, (2022) , https://doi.org/10.3390/electronics11030403

[26]Shukla, S., & Goyal, S. Deep Learning-Based Image Forgery Detection Using CNN. 2021 6th International Conference on Advanced Co

[27] A Kuznetsov, “Digital image forgery detection using deep learning approach”, 2019 J. Phys.: Conf. Ser. 1368 032028, doi10.1088/1742-6596/1368/3/032028

[28] Doegar, Amit, Maitreyee Dutta, and Kumar Gaurav. ”Cnn based image forgery detection using pre-trained alexnet model.” International Journal of Computational Intelligence IoT 2, no. 1 (2019)

[29] Bonettini, N., Cannas, E. D., Mandelli, S., Bondi, L., Bestagini, P., & Tubaro, S. (2021, January). Video face manipulation detection through ensemble of cnns.

In *2020 25th international conference on pattern recognition (ICPR)* (pp. 5012-5019). IEEE.

[30] Zhang, W., Zhao, C., & Li, Y. (2020). A novel counterfeit feature extraction technique for exposing face-swap images based on deep learning and error level analysis. *Entropy*, *22*(2), 249.

[31] Nirkin, Y., Wolf, L., Keller, Y., & Hassner, T. (2021). Deepfake detection based on discrepancies between faces and their context. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(10), 6111-6121

[32] Ciftci, U. A., Demir, I., & Yin, L. (2020). Fakecatcher: Detection of synthetic portrait videos using biological signals. IEEE transactions on pattern analysis and machine intelligence.