A

MINI PROJECT REPORT

ON

"DEEPFAKE DETECTION USING RESNEXT AND LSTM"

Submitted to the

Dr. Babasaheb Ambedkar Technological University
Lonere, Raigad

in fulfilment of the requirements

for the award of the degree

BACHELORS OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING
2023-2024

BY

Vaishnavi Yadav	**2164191242007**
Paras Patil	**2164191242013**
Narendra Rathore	**2164191242019**
Prajyot Panmand	**2164191242008**

UNDER THE GUIDANCE OF

Prof. Dhammjyoti Dhawase

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PCET-NMVPM's

NUTAN COLLEGE OF ENGINEERING AND RESEARCH

TALEGAON, PUNE 410507

PCET-NMVPM's

NUTAN COLLEGE OF ENGINEERING & RESEARCH TALEGAON, PUNE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled **"DEEPFAKE DETECTION USING RESNEXT AND LSTM"**, which is being Submitted by, **Vaishnavi Yadav, Paras Patil, Narendra Rathore, Prajyot Panmand** as partial fulfillment for the **Degree Bachelor of Technology** (Computer Science and Engineering) of **DBATU, Lonere.** This is bonafide work carried under my supervision and guidance.

Place: Pune

Date: 23/06/2023

Prof. Dhammjyoti Dhawase
Project Guide

Dr. Sanjeevkumar Angadi
Head of Department

Dr. Aparna Pande
Principal

External Examiner [Name & Sign]

SEAL

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible. So, we acknowledge all those whose guidance and encouragement served as a beacon light and crowned our efforts with success.

We have immense pleasure in expressing thanks to the principal ***Dr. Aparna Pande*** for providing all the facilities for the successful completion of the project.

With due respect, we thank our Head of Department ***Dr. Sanjeevkumar Angadi, Computer Science and Engineering***, for his motivating support, keen interest which kept our spirits alive all through.

We would like to express thanks to our guide ***Prof. Dhammjyoti Dhawase.*****,** Department of ***Computer Science and Engineering*** who has guided us throughout the completion of this project.

Finally, we would like to thank ***all the teaching and non-teaching staff and all our friends*** who have rendered their support in the completion of this report.

Vaishnavi Govind Yadav _____

Paras Shreyons Patil _____

Narendra Kansingh Rathore _____

Prajyot Navnath Panmand _____

ABSTRACT

The growing computation power has made the deep learning algorithms so powerful that creating a indistinguishable human synthesized video popularly called as deep fakes have became very simple. Scenarios where these realistic face swapped deep fakes are used to create political distress, fake terrorism events, revenge porn, blackmail peoples are easily envisioned.

In this work, we describe a new deep learning-based method that can effectively distinguish AI-generated fake videos from real videos.Our method is capable of automatically detecting the replacement and reenactment deep fakes. We are trying to use Artificial Intelligence (AI) to fight Artificial Intelligence (AI). Our system uses a Res-Next Convolution neural network to extract the frame-level features and these features and further used to train the 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.

To emulate the real time scenarios and make the model perform better on real time data, we evaluate our method on large amount of balanced and mixed data-set prepared by mixing the various available data-set like Face-Forensic++[1], Deepfake detection challenge[2], and Celeb-DF[3]. We also show how our system can achieve competitive result using very simple and robust approach.

Keyword – *Res-Next Convolution neural network, Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), Computer Vision.*

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 INTRODUCTION

In the world of ever-growing social media platforms, Deepfakes are considered as the major threat of the AI. There are many Scenarios where these realistic face swapped deepfakes are used to create political distress, fake terrorism events, revenge porn, blackmail peoples are easily envisioned. Some of the examples are Brad Pitt, Angelina Jolie nude videos. It becomes very important to spot the difference between the deepfake and pristine video.

We are using AI to fight AI. Deepfakes are created using tools like FaceApp and Face Swap , which using pre-trained neural networks like GAN or Auto encoders for these deepfakes creation. Our method uses a LSTM based artificial neural network to process the sequential temporal analysis of the video frames and pre-trained Res-Next CNN to extract the frame level features. ResNext Convolution neural network extracts the frame-level features and these features are further used to train the Long Short-Term Memory based artificial Recurrent Neural Network to classify the video as Deepfake or real.

To emulate the real time scenarios and make the model perform better on real time data, we trained our method with large amount of balanced and combination of various available dataset like FaceForensic++, Deepfake detection challenge, and Celeb-DF. Further to make the ready to use for the customers, we have developed a front-end application where the user the user will upload the video. The video will be processed by the model and the output will be rendered back to the user with the classification of the video as deepfake or real and confidence of the model.

Advantages of Deepfake Detection

Deepfake detection using machine learning technology is a critical endeavor in the contemporary digital landscape, where the proliferation of manipulated media poses a significant threat to the authenticity of visual content. As the prevalence of deepfake techniques continues to rise, implementing robust detection mechanisms becomes imperative for maintaining trust and integrity in various domains. One key advantage of employing deepfake detection is the preservation of credibility in sensitive areas such as journalism, where misinformation can have profound societal consequences.

Beyond safeguarding against malicious intent, the integration of machine learning-based detection systems also offers several other advantages.

i. Preservation of Credibility

Deepfake detection through machine learning is vital for preserving the credibility of visual content, especially in sensitive sectors like journalism. Misinformation fueled by manipulated media can have far-reaching societal consequences. The ability to discern authentic content from deepfakes ensures that public trust remains intact, bolstering the reliability of information sources.

ii. Real-time Monitoring and Rapid Identification

Machine learning-based detection systems provide real-time monitoring capabilities, swiftly identifying and flagging deepfake content. This rapid identification is crucial for implementing timely responses, preventing the unchecked dissemination of misleading information. By staying ahead of the curve, organizations can effectively counteract the potential harm caused by deepfakes in various contexts.

iii. Adaptability to Emerging Threats

The adaptability of machine learning algorithms in deepfake detection is a key advantage. These algorithms can evolve to counter new and emerging deepfake techniques, providing a dynamic defense against evolving threats. This adaptability is essential in an environment where malicious actors are constantly refining their methods, ensuring that detection systems remain effective and resilient over time.

iv. Enhanced Cybersecurity

Implementing deepfake detection mechanisms contributes significantly to overall

cybersecurity. By identifying and neutralizing manipulated content before it gains widespread traction, these systems protect individuals and organizations from reputational damage and financial losses. The proactive nature of deepfake detection enhances the digital security posture, creating a robust defense against potential threats in the ever-evolving landscape of cybersecurity.

v. Promotion of Digital Literacy

Deepfake detection systems play a vital role in fostering public awareness about the existence and risks associated with deepfake technology. Educating the public on the nuances of synthetic media promotes digital literacy, empowering individuals to critically evaluate content authenticity. This awareness is essential for responsible media consumption, contributing to a more informed and discerning digital society.

vi. Scalability for Efficient Processing

The scalability of machine learning models allows for the efficient processing of large datasets, a crucial aspect of deepfake detection. This scalability enables the screening of vast amounts of multimedia content regularly, making it particularly advantageous for applications like social media platforms or online content-sharing sites where a high volume of content requires regular and thorough scrutiny.

1.2 NECESSITY

The growing necessity for deepfake detection arises from the escalating threat posed by manipulated media in various facets of society. In realms like politics, journalism, and entertainment, undetected deepfakes have the potential to erode trust and compromise the integrity of information dissemination. As deepfake technology becomes more accessible and sophisticated, the risk of misinformation amplifies, making it imperative to implement robust detection mechanisms. Failure to address this challenge can lead to severe consequences, including the erosion of public trust in media, the distortion of public discourse, and potential harm to individuals and organizations.

Moreover, national security concerns underscore the urgency of deepfake detection. Malicious actors could exploit deepfake technology to create convincing yet fabricated content for the purpose of deception or coercion. Detecting and preventing such instances is not only vital for safeguarding democratic processes but also for maintaining geopolitical stability. The necessity for deepfake detection thus extends beyond individual sectors to encompass broader societal well-being and security.

Additionally, the sheer volume and speed at which information circulates in the digital age heighten the urgency for real-time identification of deepfakes. Traditional methods of verification often lag behind the rapid spread of manipulated content, necessitating the deployment of advanced machine learning algorithms for swift and accurate detection. The necessity for deepfake detection is, therefore, a crucial component of a proactive strategy to mitigate the risks associated with the malicious use of synthetic media.

How it is useful?

The utility of deepfake detection lies in its multifaceted impact on preserving the fabric of trustworthy information and securing digital landscapes. Firstly, it acts as a proactive defense mechanism, preventing the potential harm caused by the spread of manipulated content. By swiftly identifying deepfakes, detection systems allow for timely interventions, mitigating the impact on public perception, corporate reputation, and national security.

Furthermore, deepfake detection contributes to building resilience against evolving threats. The adaptability of machine learning algorithms enables these systems to stay ahead of emerging deepfake techniques, ensuring that detection mechanisms remain effective over time. This adaptability is crucial for maintaining the efficacy of detection systems in the face of constantly evolving tactics employed by those seeking to exploit synthetic media for malicious purposes.

In a broader societal context, the usefulness of deepfake detection is evident in its role as an educational tool. By raising awareness about the existence and potential risks of deepfake technology, these systems contribute to digital literacy. This awareness empowers individuals to critically evaluate media content, fostering responsible consumption habits and creating a more informed and discerning public. Overall, the utility of deepfake detection extends beyond immediate threat mitigation, actively contributing to the resilience, awareness, and integrity of digital ecosystems.

1.3 SYSTEM REQUIREMNETS

1.3.1 Hardware Requirements

A. Server Side

Operating System	Linux/Windows
Processor	Dual Core

B. Client Side

Operating System	Windows
RAM	16 Gigabyte
Hard Drive	100 GB Or Above
Internet Connection	1 MB Or more

1.3.2 Software Requirements

A. Server-Side

Operating System	Windows
Database	SQL
Programming Language	Html, CSS, Ajax, Python 3.0

B. Client-Side

Operating System	Linux/Windows
Web Browser	Mozilla Firefox,Chrome
Framework	PyTorch 1.4, Django 3.0
Cloud Platform	Google Cloud Platform
Libraries	OpenCV, Face-recognition

1.4 PROBLEM STATEMENT

Convincing manipulations of digital images and videos have been demonstrated for several decades through the use of 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 , 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.

1.5 OBJECTIVES

1. Our project aims at discovering the distorted truth of the deep fakes.
2. Our project will reduce the Abuses' and misleading of the common people on the world wide web.
3. Provide an easy-to-use system for used to upload the video and distinguish whether the video is real or fake.
4. Our project will distinguish and classify the video as deepfake or pristine

1.6 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. So-called "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.

2. LITERATURE SURVEY

Table 2.1 Literature survey

Sr. No	Paper Name,Year	Datasets	Methods/ Techniques	Accuracy	Advantage	Disadvantage	Future scope
1.	Deepfakes Detection Techniques Using Deep Learning: A Survey 2021	1) FFHQ; 2) 100K-Faces; 3) DFFD; 4) CASIA-WebFace; 5)VGGFace2 and (6) The eye-blinking dataset; 7) DeepfakeTIMIT	CNN, RNN, LSTM Biological Analysis	-	The model employs a Siamese network architecture for concurrent extraction of speech and face modalities.	The current deep learning methods need to improve as well to successfully identify fake videos and images	Current deep learning lacks clear methods for determining optimal layer count and selecting appropriate architecture in deepfake detection.
2.	Using Deep Learning to Detecting Deepfakes 2022	Celeb-DF v2, DeepfakeTIMIT, FaceForensics++, Celeb-DF	CNN. ResNet 18, Temporal Feature Deepfake Detection Method	92%	Provides a good benchmark to compare models to each other, as results can be heavily dependent on the dataset applied	One issue that plagues many deepfake models is their lack of transferability	Developing models that are able to perform well on multiple different databases in order to prepare this technology to be applied in a real-world context and aid people online around the world.

3.	Deepfake Video Detection 2020	Faceforensics++ and MesoNet, DFDC	RNN, DeepVision, Meso4	89%	There Are Benefits to Correlations and Regression Analysis	Accuracy is potentially not as high as it is claimed to be	This project highlights the importance of continuously testing previous detection methods to assess their ongoing effectiveness in identifying deepfake content.
4.	A Machine Learning Based Approach for Deepfake Detection in Social Media Through Key Video Frame Extraction 2021	FF++, DFDC	CNN, GAN, DNN	90%	An algorithm which cuts down the computational burden significantly has been proposed.	We avoided training with enormous amounts of data even though we accommodated a large number of videos.	Reducing the run-time memory and the model size would be a great effort as the future work.
5.	Deepfake Detection: A Systematic Literature Review 2022	FF, FF++, DFD, DFDC, FD-TIMIT, SMFW	RNN, CNN, GANs	-	It can be stated that, in general, the deep learning models outperform the non-deep learning models.	Construct validity, internal validity	This SLR provides a valuable resource for the research community in developing effective detection methods and

							countermeasures.
6.	A survey on deepfake video detection datasets 2022	DeepFakeDetection, Faceforensics++, DeeperForensics-1.0	Celeb-DF-v2, GAN (FSGAN), FOMM, ATFHP, and Wav2Lip	-	Universal detection model since fake video creation technology has improved a lot.	The results show detection scores of the datasets are high for some models and also have lower performance scores	It is very expected that further competitive deepfake detection models will comes up in future articles with most accurate results
7.	Deepfake Detection: A Comprehensive Study from the Reliability Perspective 2022	UADFV, DeepfakeTIMIT, FF++	GAN	91%	Interpretability, transferability	Robustness challenges is not yet accomplished in the current research domain	Tracing original sources of synthetic contents and recovering synthetic operation sequences
8.	Deepfake Detection: A Comparative Analysis 2022	FakeAVCeleb, CelebDFV2, DFDC, and FaceForensics++	CNN, Xception	-	Achieve better overall performance in intra-dataset comparison	They do not provide the models with any generalisation capability	Self-supervised training strategies to train models, as well as try to incorporate knowledge distillation
9.	A GAN-Based Model of Deepfake Detection in Social Media 2022	DDFD; 100K-Faces; FFHQ ; Deepfake TIMIT; VGGFace2; the eye-blinking dataset; CASIA-WebFac	GAN	91%	Discusses the performance of deep convolution GAN for 10 successive iterations.	Losses of discriminator get decrease while of generator increases with each successive iteration	To improve the handling of small datasets and overcome GAN constraints

10.	Deepfake Video Detection Using Convolutional Neural Network 2022	Celeb-DF	CNN, RNN	93%	The proposed model works well and is able to successfully gather features required for further processing to test for deepfakes.	It is observed that the accuracy of the proposed model decreases with low quality images and with medium quality video	Research in the field of image & video forgery and digital media forensics.
11.	Short And Low Resolution Deepfake Video Detection Using CNN 2022	DFDC, FF++	CNN	92%	Convolutional Neural Network (CNN) can accurately predict whether or not a video has been manipulated	It is so hard to discover deepfake recordings where the manipulation is just present in a little part of the video	How we may improve our system's robustness against altered videos by using unknown techniques during training.
12.	A Review of Deep Learning-based Approaches for Deepfake Content Detection 2022	Celeb-DF, and DeepfakeDetection (DFD), FF++	Face2Face, FaceSwap, SVM	91%	Complexity and realism of deepfake methods are actively refined as a consequence of the advances in deep learning techniques	Time-consuming and laborious manual annotation of massive amounts of new data	Dynamic procedure or even the combination of supervised and unsupervised learning to rapidly identifying and actively tracking the patterns
13.	Deepfake Attacks: Generation, Detection, Datasets, Challenges, and Research Directions 2022	FF++, DFDC, Celeb-DF, WildDeepfake	StyleGAN, AttGAN	91%	Forgery detection without capturing the intricacies and subtleties of advanced deepfakes	The quality of the deepfake datasets is yet another prominent challenge in deepfake detection.	Focus on finding a balance between utilizing deepfakes' beneficial potential and reducing

							their negative effects
14.	Deepfake Detection with Deep Learning: Convolutional Neural Networks versus Transformers 2022	FF++, DFDC, Celeb-DF	CNN	92%	Models performed very well consistently when trained	Models trained on the Deeper Forensics dataset fared poorly when tested on the other datasets.	Future works along with newer and more sophisticated deepfakes datasets.
15.	Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods 2022	FFHQ	CNN,DNN, XAI	90%	Correctly identify real images to make the model trustworthy	XAI is limited to images, but in the future, video samples could be validated by XAI	Researchers may have to look at medical images to implement and get the best outcome from deepfake detection for further medical analysis
16.	ResViT: A Framework for Deepfake Videos Detection2022	DFDC2, Celeb-DF	ResVIT	92%	We use a couple of libraries like BlazeFace and MTCNN for face extraction, known as the rapid processing of large amounts of images	We present the training and validation losses and accuracy on three sample datasets for the proposed ResViT	Performance of the ResViT under massive datasets with more baseline models
17.	DEEPFAKE DETECTION THROUGH DEEP LEARNING USING RESNEXT CNN AND LSTM 1 2022	FF++, DFDC, Celeb-DF	CNN , RNN	93%	It can accurately predict the output by analyzing just one second of video	Integrating it into a larger sof	Processing the temporal sequence and identifying changes between consecutive frames.

18.	Deepfake Detection on Social Media: Leveraging Deep Learning and fasttext Embeddings for Identifying Machine-Generated Tweets 2016	DFDC	CNN, LSTM, SGC	87%	The adoption of a CNN model structure in this Study shows its superiority in terms of simplicity, computational efficiency	No higher accuracy	The quantum NLP and other cutting-edge methodologies will be applied for more sophisticated and efficient detection systems

3. SYSTEM DEVELOPMENT

3.1 Technologies We Used

3.1.1 Planning

1. OpenProject : OpenProject serves as a collaborative project management tool that facilitates efficient planning, tracking, and coordination of tasks. Its features include Gantt charts, task boards, and a centralized platform for team communication. OpenProject aids in organizing project timelines, assigning responsibilities, and maintaining a transparent overview of project progress, ensuring a structured and streamlined development process.

3.1.2 UML Tools

1. draw.io : As a web-based diagramming tool, draw.io provides a versatile platform for creating Unified Modeling Language (UML) diagrams. It supports the visualization of system architecture, data models, and various UML diagram types. draw.io's user-friendly interface and extensive library of shapes make it an invaluable tool for designing and communicating system structures, enhancing the clarity of project documentation.

3.1.3 Programming Languages

1. Python3 : Python3, known for its simplicity and readability, serves as the primary programming language for the deepfake detection project. Its extensive libraries, such as PyTorch and NumPy, facilitate machine learning tasks and data manipulation, making it a powerful language for developing sophisticated algorithms.

2. JavaScript : JavaScript is utilized for web-based components and interactivity in the project. Its versatility and compatibility with web browsers make it a valuable asset for enhancing the user interface and overall user experience.

3.1.4 Programming Frameworks

1. PyTorch : PyTorch is a popular open-source machine learning framework, chosen for its flexibility and dynamic computation graph. It is well-suited for building and training deep neural networks, making it a cornerstone for the deepfake detection model development.

2. Django : Django, a high-level Python web framework, is employed for developing the backend of the application. Its "batteries-included" philosophy streamlines the development process, offering features like ORM (Object-Relational Mapping) and authentication, ensuring robust and scalable web application development.

3.1.5 IDE

1. Google Colab : Google Colab, a cloud-based Jupyter notebook environment, provides a collaborative platform for writing and executing Python code. Its integration with Google Drive allows for seamless sharing and version control, enhancing collaboration among team members during model development and experimentation.

2. Jupyter Notebook : Jupyter Notebook is another interactive computing environment used for creating and sharing documents that contain live code, equations, visualizations, and narrative text. It facilitates a literate programming approach, making it easy to document and present code workflows and results.

3. Visual Studio Code : Visual Studio Code (VS Code) is a versatile code editor that supports various programming languages. Its rich set of extensions, integrated Git support, and debugging capabilities enhance the development experience, especially for code editing and collaboration.

3.1.6 Versioning Control

1. Git : Git is a distributed version control system that allows multiple developers to collaborate seamlessly on code projects. It enables tracking changes, managing different versions, and maintaining a cohesive codebase, ensuring efficient collaboration and easy rollbacks if needed.

3.1.7 Cloud Services

1. Google Cloud Platform : Google Cloud Platform (GCP) provides a suite of cloud services that includes computing power, storage, and machine learning capabilities. Leveraging GCP allows for scalable and reliable deployment of the deepfake detection application, with services such as Google Cloud Engine supporting application and web servers.

3.1.8 Application and web servers:

1. Google Cloud Engine : Google Cloud Engine is used as an application and web server to host and deploy the deepfake detection application. Its scalability, reliability, and integration with other GCP services make it a suitable choice for serving the application to end-users.

3.1.9 Libraries

1. torch and torchvision
2. os, numpy, cv2, matplotlib
3. face_recognition
4. json, pandas, copy, glob, random, sklearn

3.2 SYSTEM OVERVIEW

3.2.1 System Architecture

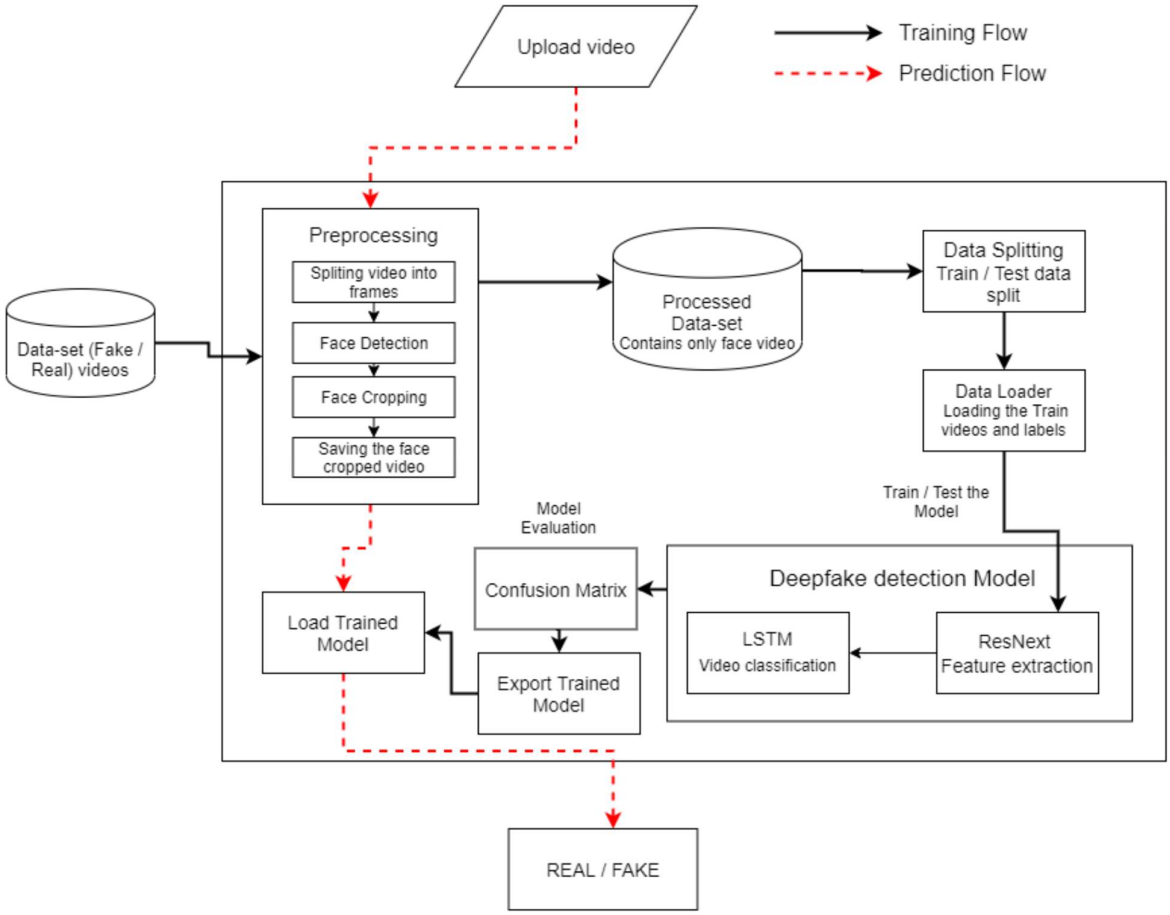

Figure 3.1 Deepfake generation

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

Creating deepfake videos

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

networks models are very realistic that it is almost impossible to spot the difference by the naked eyes. But in reality, the deepfakes creation tools leaves some of the traces or artifacts in the video which may not be noticeable by the naked eyes. The motive of this paper to identify these unnoticeable traces and distinguishable artifacts of these videos and classified it as deepfake or real video.

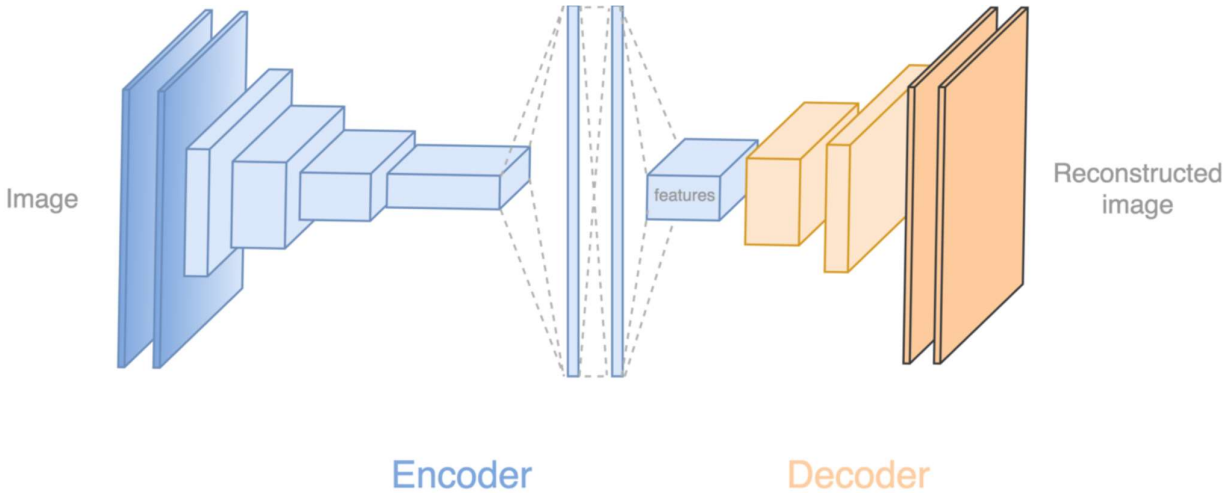

Figure 3.2 Deepfake generation

Figure 3.3 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

3.2.2 Architectural Design

Module 1: Data-set Gathering

When accessing healthcare services, users present their QR code to authenticate themselves. This can be done by scanning the QR code using a dedicated mobile app or a QR code reader. For making the model efficient for real time prediction. We have gathered the data from different available data-sets like FaceForensic++(FF), Deepfake detection challenge(DFDC), and Celeb-DF. Futher we have mixed the dataset the collected datasets and created our own new dataset, to accurate and real time detection on different kind of videos. To avoid the training bias of the model we have considered 50% Real and 50% fake videos. Deep fake detection challenge (DFDC) dataset consist of certain audio alerted video, as audio deepfake are out of scope for this paper. We preprocessed the DFDC dataset and removed the audio altered videos from the dataset by running a python script. After preprocessing of the DFDC dataset, we have taken 1500 Real and 1500 Fake videos from the DFDC dataset. 1000 Real and 1000 Fake videos from the FaceForensic++(FF) dataset and 500 Real and 500 Fake videos from the Celeb-DF dataset. Which makes our total dataset consisting 3000 Real, 3000 fake videos and 6000 videos in total. Figure 2 depicts the distribution of the data-sets.

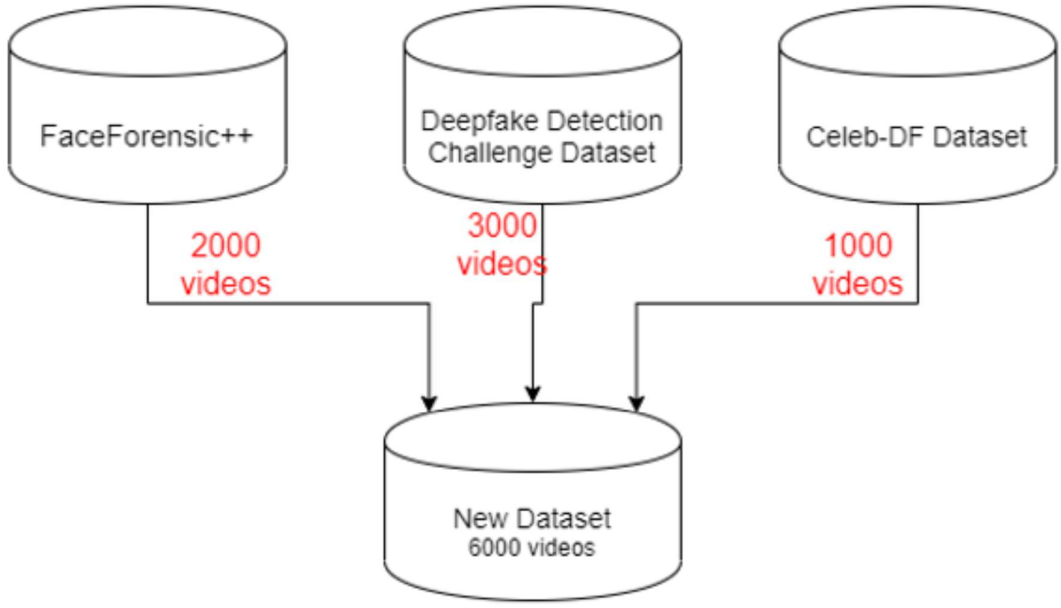

Figure 3.4 Dataset

Module 1: Pre-processing

In this step, the videos are preprocessed and all the unrequired and noise is removed from videos. Only the required portion of the video i.e face is detected and cropped. The first steps in the preprocessing of the video is to split the video into frames. After splitting the video into frames the face is detected in each of the frame and the frame is cropped along the face. Later the cropped frame is again converted to a new video by combining each frame of the video. The process is followed for each video which leads to creation of processed dataset containing face only videos. The frame that does not contain the face is ignored while preprocessing. To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each video. Another reason for selecting a threshold value is limited computation power. As a video of 10 second at 30 frames per second(fps) will have total 300 frames and it is computationally very difficult to process the 300 frames at a single time in the experimental environment. So, based on our Graphic Processing Unit (GPU) computational power in experimental environment we have selected 150 frames as the threshold value. While saving the frames to the new dataset we have only saved the first 150 frames of the video to the new video. To demonstrate the proper use of Long Short-Term Memory (LSTM) we have considered the frames in the sequential manner i.e. first 150 frames and not randomly. The newly created video is saved at frame rate of 30 fps and resolution of 112 x 112.

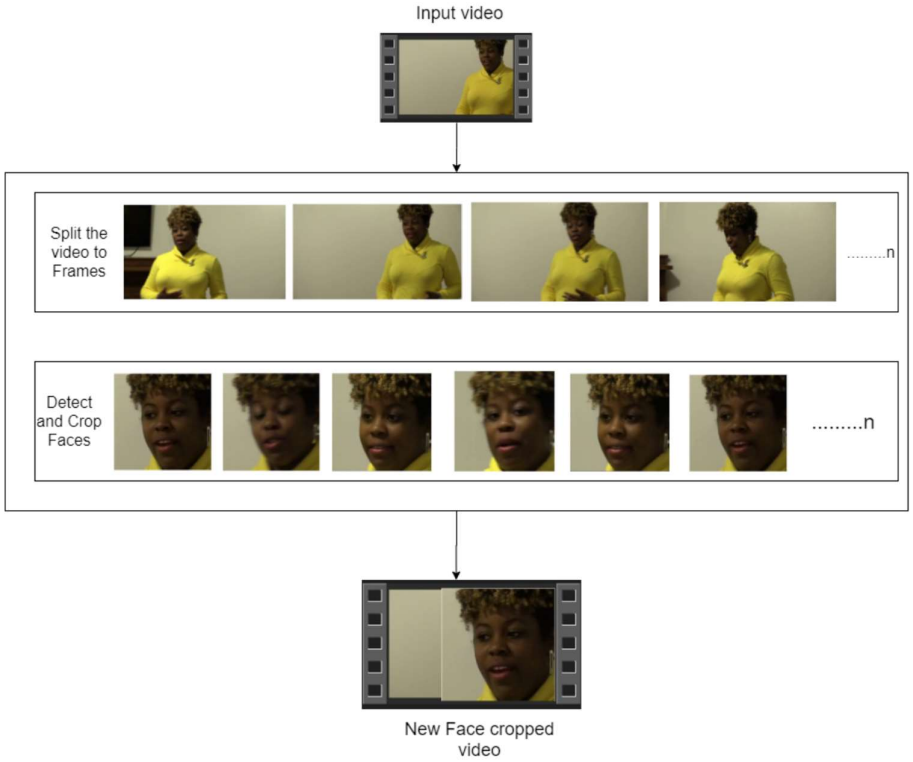

Figure 3.5 Pre-processing of video

Module 3: Data-set split

The dataset is split into train and test dataset with a ratio of 70% train videos (4,200) and 30% (1,800) test videos. The train and test split is a balanced split i.e 50% of the real and 50% of fake videos in each split.

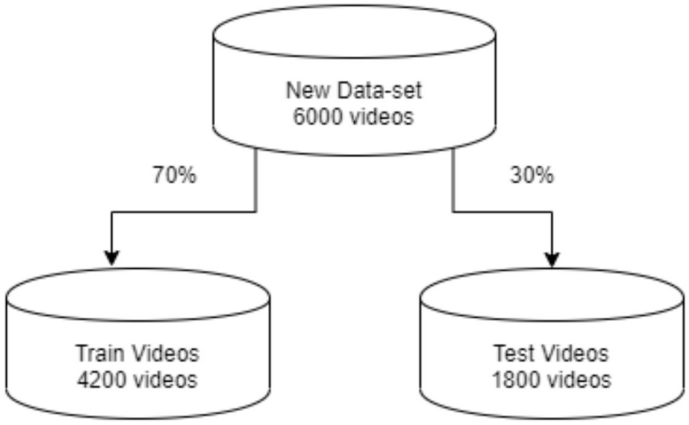

Figure 3.6 Train test split

Module 4: Model Architecture

Our model is a combination of CNN and RNN. We have used the Pre- trained ResNext CNN model to extract the features at frame level and based on the extracted features a LSTM network is trained to classify the video as deepfake or pristine. Using the Data Loader on training split of videos the labels of the videos are loaded and fitted into the model for training.

ResNext:

Instead of writing the code from scratch, we used the pre-trained model of ResNext for feature extraction. ResNext is Residual CNN network optimized for high performance on deeper neural networks. For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and 32 x 4 dimensions. Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimensional feature vectors after the last pooling layers of ResNext is used as the sequential LSTM input.

LSTM for Sequence Processing:

2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable

to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at 't' second with the frame of 't-n' seconds. Where n can be any number of frames before t. The model also consists of Leaky Relu activation function. A linear layer of 2048 input features and 2 output features are used to make the model capable of learning the average rate of correlation between eh input and output. An adaptive average polling layer with the output parameter 1 is used in the model. Which gives the the target output size of the image of the form H x W. For sequential processing of the frames a Sequential Layer is used. The batch size of 4 is used to perform the batch training. A SoftMax layer is used to get the confidence of the model during predication.

Figure 3.7 Overview of our model

Module 5: Hyper-parameter tuning

It is the process of choosing the perfect hyper-parameters for achieving the maximum accuracy. After reiterating many times on the model. The best hyper-parameters for our dataset are chosen. To enable the adaptive learning rate Adam optimizer with the model parameters is used. The learning rate is tuned to 1e-5 (0.00001) to achieve a better global minimum of gradient descent. The weight decay used is 1e-3. As this is a classification problem so to calculate the loss cross entropy approach is used.To use the available computation power properly the batch training is used. The batch size is taken of 4. Batch size of 4 is tested to be ideal size for training in our development environment. The User Interface for the application is developed using Django framework. Django is used to enable the scalability of the application in the future. The first page of the User interface i.e index.html contains a tab to browse and upload the video. The uploaded video is then passed to the model and prediction is made by the model. The model returns the output whether the video is real or fake along with the confidence of the model. The output is rendered in the predict.html on the face of the playing video.

3.3 PROCESS

3.3.1 Processing Details

- Using glob we imported all the videos in the directory in a python list.
- cv2.VideoCapture is used to read the videos and get the mean number of frames in each video.
- To maintain uniformity, based on mean a value 150 is selected as idea value for creating the new dataset.
- The video is split into frames and the frames are cropped on face location.
- The face cropped frames are again written to new video using VideoWriter.
- The new video is written at 30 frames per second and with the resolution of 112 x 112 pixels in the mp4 format.
- Instead of selecting the random videos, to make the proper use of LSTM for temporal sequence analysis the first 150 frames are written to the new video.

3.3.2 Model Details

The model consists of following layers:

- **ResNext CNN** : The pre-trained model of Residual Convolution Neural Network is used. The model name is resnext50 (32x4d). This model consists of 50 layers and 32 x 4 dimensions. Figure shows the detailed implementation of model.

stage	output	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 [1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax
# params.		**25.0**×10^6

Figure 3.8 ResNext Architecture

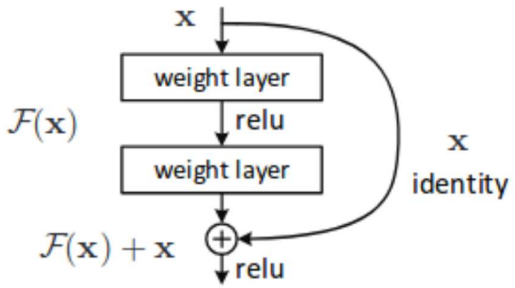

Figure 3.9 ResNext Working

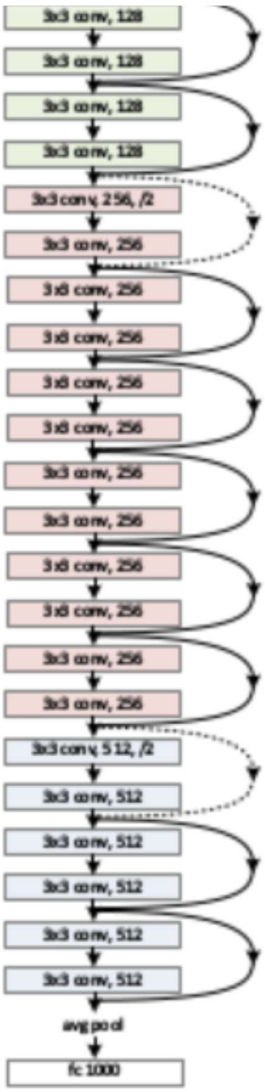

Figure 3.10 Overview of ResNext Architecture

- **Sequential Layer:** Sequential is a container of Modules that can be stacked together and run at the same time. Sequential layer is used to store feature vector returned by the ResNext model in a ordered way. So that it can be passed to the LSTM sequentially.

- **LSTM Layer:** LSTM is used for sequence processing and spot the temporal change between the frames.2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at 't' second with the frame of 't-n' seconds. Where n can be any number of frames before t.

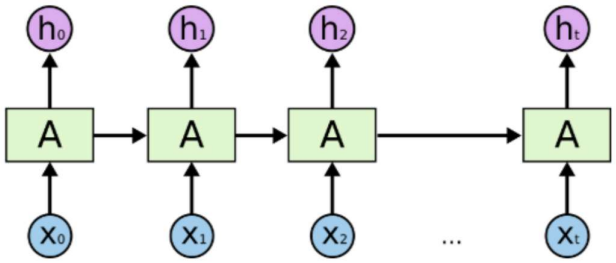

Figure 3.11 Overview of LSTM Architecture

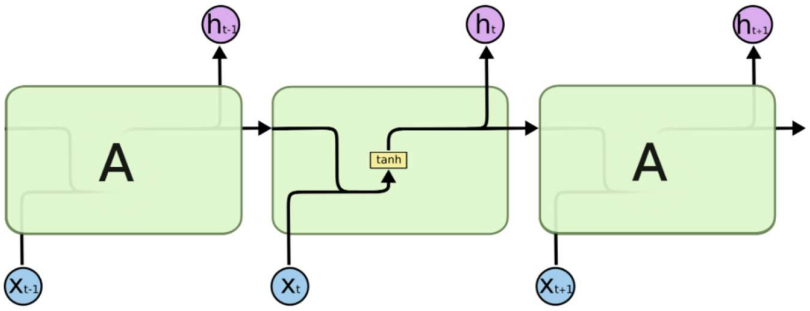

Figure 3.12 Internal LSTM Architecture

- **ReLU:** A Rectified Linear Unit is activation function that has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work. ReLU is non-linear and has the advantage of not having any backpropagation errors unlike the sigmoid function, also for larger Neural Networks, the speed of building models based off on ReLU is very fast.

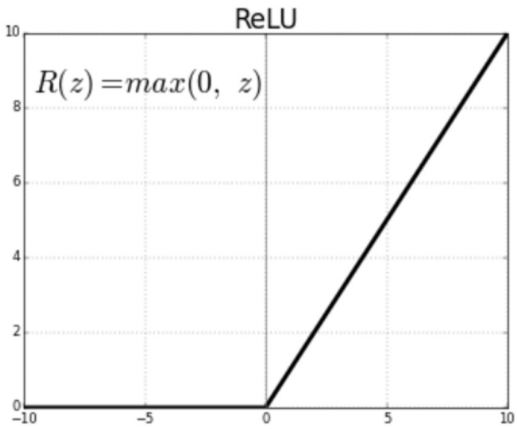

Figure 3.13 Relu Activation function

3.3.2 Model Prediction Details

- The model is loaded in the application
- The new video for prediction is preprocessed and passed to the loaded model for prediction
- The trained model performs the prediction and return if the video is a real or fake along with the confidence of the prediction.

3.4 BLOCK DIAGRAMS

Use Case View diagram

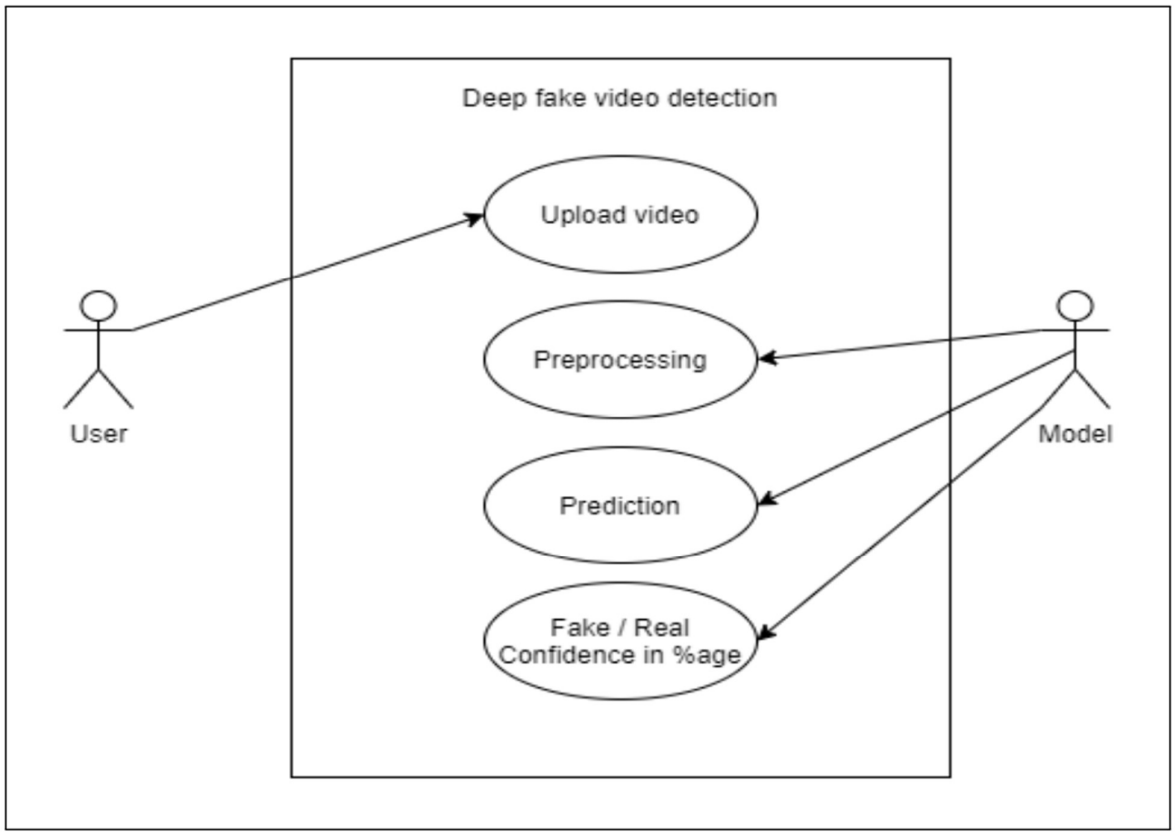

Figure 3.14 Use Case diagram

DFD Level -0

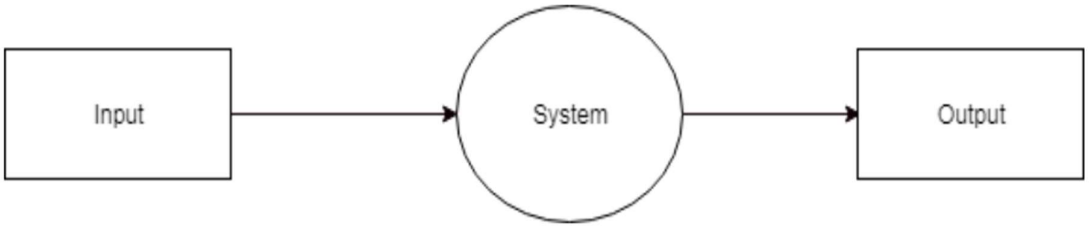

Figure 3.15 DFD Level 0

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

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

Hence, the data flow diagram indicates the visualization of system with its input and output flow.

DFD Level -1

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

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

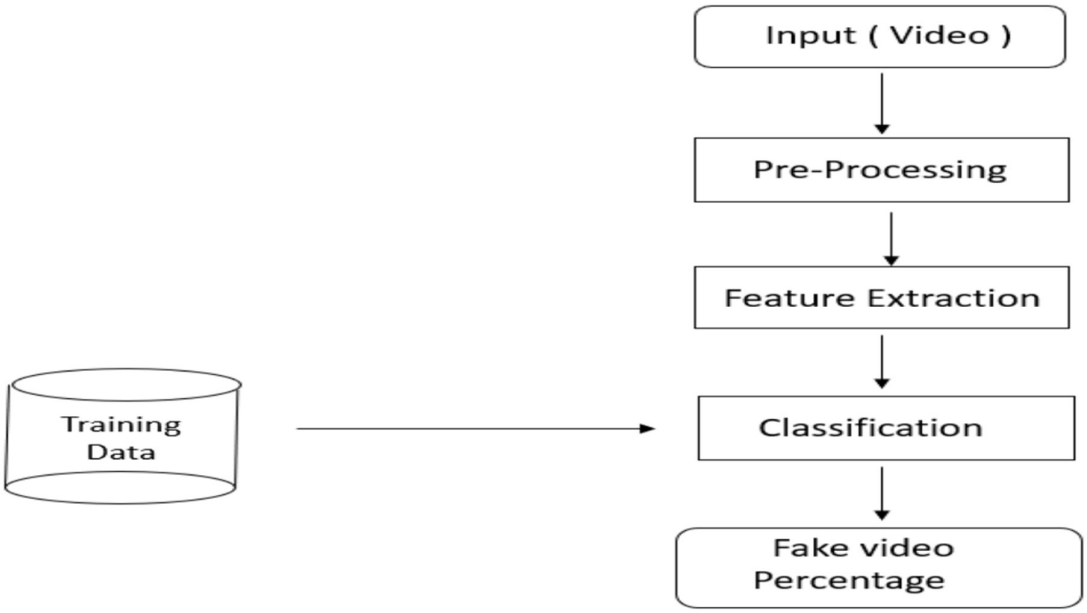

Figure 3.16 DFD Level 1

DFD Level -2

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

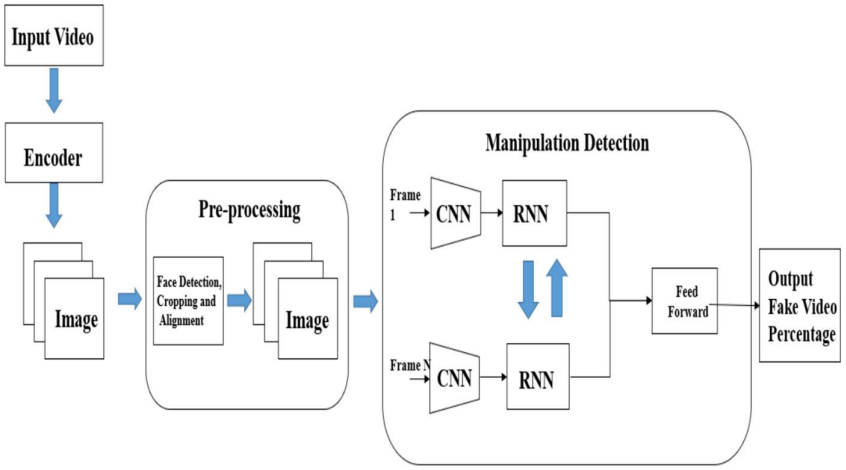

Figure 3.17 DFD Level 2

Activity Diagram

Training Workflow:

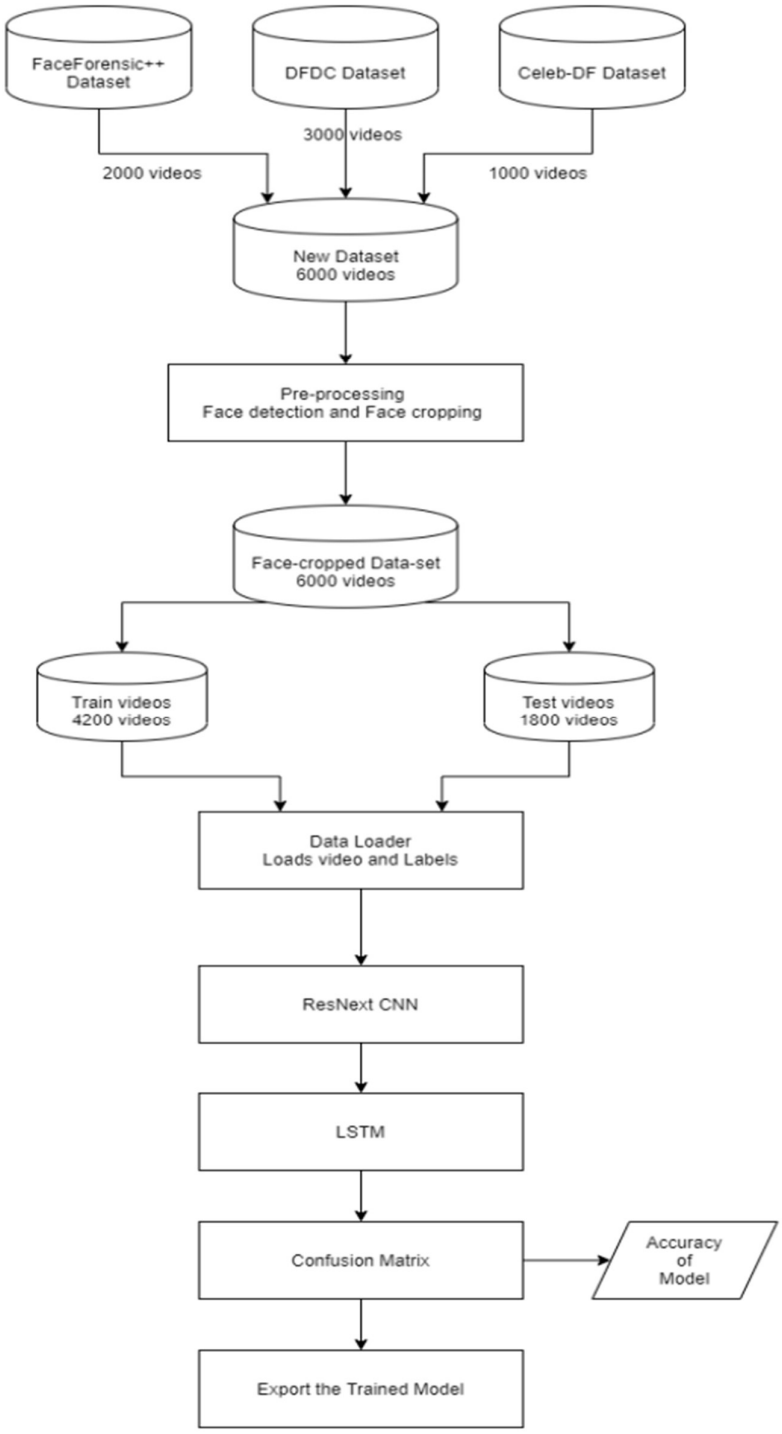

Figure 3.18 Training Workflow

Testing Workflow:

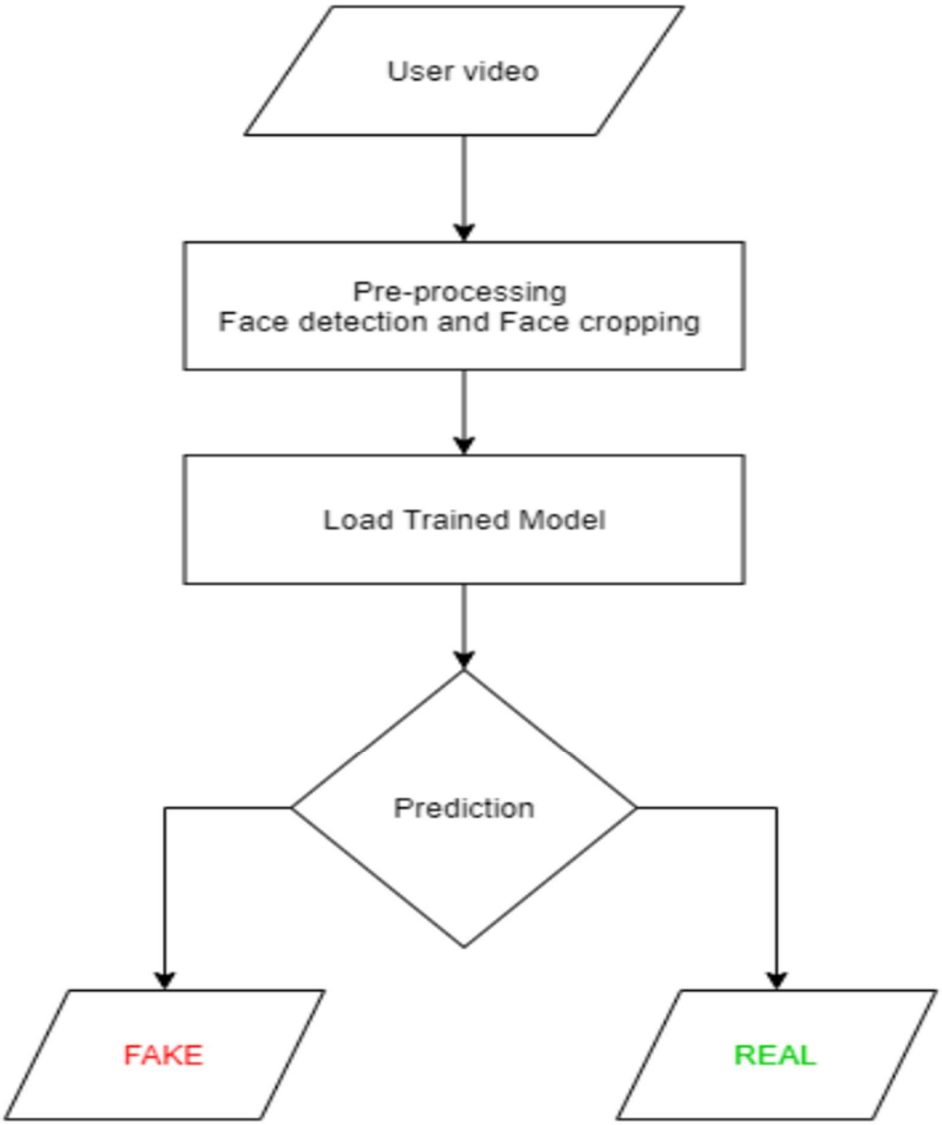

Figure 3.19 Testing Workflow

Sequence Diagram:

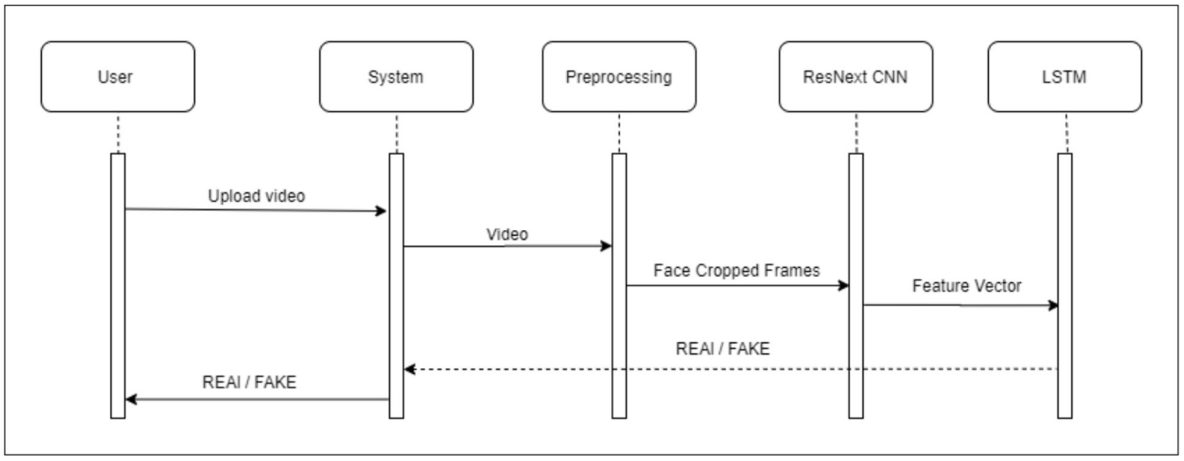

Figure 3.20 Sequence Diagram

4. PERFORMANCE ANALYSIS

4.1 TESTING

A. TYPE OF TESTING USED

Functional Testing

1. Unit Testing
2. Integration Testing
3. System Testing
4. Interface Testing

Non-functional Testing

1. Performance Testing
2. Load Testing
3. Compatibility Testing

B. TEST CASES AND TEST RESULT

We perform all the testing related website following Table 4.1 shows all the testing measures.

Table 4.1 Test Case Report

Case id	Test Case Description	Expected Result	Actual Results	Stat us
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

5.2 Tentative Project Timeline

A tentative time schedule of research work planned is given below**:**

Sr. No.	**Research Activity**	**Schedule**
1	Collecting Databases and Design of Tentative Module	
2	**Mini Project-I Review 1 – Presentation**	
3	Implementation and Design of Algorithms and Modules 20% To 30%	
4	**Mini Project-I Review 2 – Presentation**	
5	Results Comparisons and Testing of Whole Projects	
6	Report Write-Up, Revision and Correction	
7	**Mini Project-II Review 1 – Presentation**	
8	Module 2 execution [60% work done]	
9	Full Project completion [100% work done]	
10	**Mini Project-II Review 2 - Presentation**	
11	Submission of Report	
12	Submission of Published Paper	

6. CONCLUSION

We presented a neural network-based approach to classify the video as deep fake or real, along with the confidence of proposed model. Our method is capable of predicting the output by processing 1 second of video (10 frames per second) with a good accuracy. We implemented the model by using pre-trained ResNext CNN model to extract the frame level features and LSTM for temporal sequence processing to spot the changes between the t and t-1 frame. Our model can process the video in the frame sequence of 10,20,40,60,80,100.

6.1 APPLICATIONS

- **Media Integrity and Trustworthiness:**

Our deepfake detection system contributes to ensuring the integrity of media content, helping to maintain trust in the authenticity of visual information.

- **Social Media and Online Platforms:**

The prevalence of deepfakes on social media platforms poses a significant threat to the spread of misinformation. Our project's application in detecting deepfakes can aid in controlling the dissemination of false information online.

- **Law Enforcement and Security:**

Deepfake technology can be misused for creating fraudulent videos that may be used for malicious purposes. Implementing our deepfake detection system can support law enforcement in verifying the authenticity of visual evidence in criminal investigations.

- **Election Security:**

As elections and political events are vulnerable to misinformation campaigns, our deepfake detection system can play a crucial role in ensuring the authenticity of videos related to political figures, speeches, and events.

- **Online Identity Protection:**

Individuals may become victims of identity theft through the creation of deepfake content. Our system can help protect individuals by detecting and preventing the malicious use of their images

or videos for fraudulent purposes.

- **Corporate Security:**

Businesses and organizations may face threats from deepfakes that can be used to manipulate financial information, corporate communications, or public perception. Our project's application in deepfake detection contributes to safeguarding corporate integrity.

- **Journalism and Media Production:**

In the field of journalism and media production, ensuring the authenticity of visual content is paramount. Our deepfake detection system can be integrated into media production workflows to verify the legitimacy of media assets.

- **Personal Privacy:**

With the rising concern of deepfakes invading personal privacy, our project can provide individuals with a tool to identify and mitigate potential threats to their digital identity and reputation.

- **Education and Awareness:**

Our project can be used as an educational tool to raise awareness about the existence and risks associated with deepfakes. Educating the public about deepfake technology and the importance of detection systems can contribute to a more informed society.

6.2 FUTURE SCOPE

There is always a scope for enhancements in any developed system, especially when the project build using latest trending technology and has a good scope in future.

• Web based platform can be upscaled to a browser plugin for ease of access to the user.

• Currently only Face Deep Fakes are being detected by the algorithm, but the algorithm can be enhanced in detecting full body deep fakes.

REFERENCES

REFERENCE

1. Elise van Belle, (2020) "Feasibility and early effectiveness of the Tell-us Card communication tool to increase in-hospital patient participation: a cluster randomized controlled pilot study doi: 10.1111/scs.12909
2. Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies,Matthias Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images" in arXiv:1901.08971.
3. Deepfake detection challenge dataset : https://www.kaggle.com/c/deepfake-detection-challenge
4. 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
5. Deepfake Video of Mark Zuckerberg Goes Viral on Eve of House A.I. Hearing : https://fortune.com/2019/06/12/deepfake-mark-zuckerberg
6. 10 deepfake examples that terrified and amused the internet : https://www.creativebloq.com/features/deepfake-examples
7. TensorFlow: https://www.tensorflow.org/
8. Keras: https://keras.io/
9. PyTorch : https://pytorch.org/
10. G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. arXiv:1702.01983, Feb. 2017
11. J. Thies et al. Face2Face: Real-time face capture and reenactment of rgb videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2395, June 2016. Las Vegas, NV.
12. Face app: https://www.faceapp.com/
13. Face Swap : https://faceswaponline.com/
14. Deepfakes, Revenge Porn, And The Impact On Women :
15. https://www.forbes.com/sites/chenxiwang/2019/11/01/deepfakes-revenge-porn-andthe-impact-on-women/
16. F. Zubaydi, A. Saleh, F. Aloul, and A. Sagahyroon, (2015) "Security of mobile health (mHealth) systems," in IEEE 15th International Conference on Bioinformatics and Bioengineering, 19(1), 47-56 BIBE 2015
17. Yuezun Li, Siwei Lyu, "ExposingDF Videos By Detecting Face Warping Artifacts," in arXiv:1811.00656v3.

18. Yuezun Li, Ming-Ching Chang and Siwei Lyu “Exposing AI Created Fake Videos by Detecting Eye Blinking” in arXiv:1806.02877v2.
19. Huy H. Nguyen , Junichi Yamagishi, and Isao Echizen “ Using capsule networks to detect forged images and videos ” in arXiv:1810.11215.
20. D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.
21. I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, June 2008. Anchorage, AK
22. Umur Aybars Ciftci, ˙Ilke Demir, Lijun Yin “Detection of Synthetic Portrait Videos using Biological Signals” in arXiv:1901.02212v2
23. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, Dec. 2014.
24. ResNext Model : https://pytorch.org/hub/pytorch_vision_resnext/
25. https://www.geeksforgeeks.org/software-engineering-cocomo-model/
26. Deepfake Video Detection using Neural Networks http://www.ijsrd.com/articles/IJSRDV8I10860.pdf
27. International Journal for Scientific Research and Development http://ijsrd.com/