DEEPFAKE DETECTION USING RESNEXT101 AND LSTM

A MAJOR PROJECT REPORT

Submitted by

SHRESTH GUPTA [RA2011026010091]
UTKARSH SRIVASTAVA [RA2011026010104]

Under the guidance of

Dr. Antony Sophia N

Assistant Professor, Department of Computational Intelligence

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

with specialization in Artificial Intelligence and Machine Learning

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR– 603 203

MAY 2024

Department of Computational Intelligence

SRM Institute of Science & Technology

Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **B.Tech in CSE with specialization in AI & ML**

Student Name : **Shresth Gupta, Utkarsh Srivastava**

Registration Number : **RA2011026010091, RA2011026010104**

Title of Work : **Deepfake Detection using ResNext101 and LSTM**

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section3 of UGC Act,1956)

BONAFIDE CERTIFICATE

Certified that 18CSP109L major project report titled "**Deepfake Detection using ResNext101 and LSTM**" is the bonafide work of "**SHRESTH GUPTA [RA2011026010091], UTKARSH SRIVASTAVA [RA2011026010104]"** who carried out the major project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. ANTONY SOPHIA N	**Dr. R. ANNIE UTHRA**
GUIDE	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor
Department of Computational Intelligence	Department of Computational Intelligence

EXAMINER I	**EXAMINER II**

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work. We are incredibly grateful to our Head of the Department, **Dr. R. Annie Uthra,** Professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Panel Head, **Dr. C. Amuthadevi**, Associate Professor and Panel Members, Department of Computational Intelligence, SRM Institute of Science and Technology, for their inputs during the project reviews and support. We register our immeasurable thanks to our Faculty Advisor, **Dr. Vimaladevi M**, Assistant Professor, Department of Computational Intelligence, SRM Institute of Science andTechnology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. Antony Sophia N**, Assistant Professor, Department of Computational Intelligence, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring. Finally, we would like to thank our parents, family members, and friends for their unconditionallove, constant support and encouragement.

SHRESTH GUPTA[RA2011026010091]

UTKARSH SRIVASTAVA[RA2011026010104]

ABSTRACT

Multimedia manipulation has been on the rise with the widespread use of AI, machine learning, and deep learning technologies which has produced both encouraging developments and unsettling concerns. Although there are many legal uses for these technologies, as in entertainment and education, their misuse has led to a number of problems, with deepfakes emerging as a particularly prominent concern. Deepfakes are powerful instruments for spreading propaganda, creating political unrest, and delivering false information. They are distinguished by realistic and high-quality video, picture, or audio modifications. These features are used by malicious individuals to produce content that looks legitimate and authentic, making it difficult to distinguish between fact and fiction. Researchers and engineers have investigated a number of strategies to deal with the problems caused by deepfakes, creating detection methods to recognize modified content is one approach. Deep neural networks and machine learning algorithms are used to examine patterns, discrepancies, and features in multimedia files in an effort to distinguish real from fake information. Furthermore, attempts have been made to develop digital forensics techniques that can track the authenticity and legitimacy of multimedia files. With the use of these forensic tools, investigators will be able to determine the legitimacy of content, which is important in the fight against deepfakes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
ResNext	Residual Networks with Aggregated Transformations
CNN	Convolutional Neural Network
ResNet	Residual Networks
UI	User Interface
AI	Artificial Intelligence
HOG	Histogram of Oriented Gradients

CHAPTER 1

INTRODUCTION

Over the past few years, deepfake technology has seen widespread use, which has resulted in significant worries over the ways in which it could be exploited and utilized to manipulate content in the media. The term "deepfakes" refers to artificial media that is generated using deep learning algorithms. These algorithms have the ability to convincingly alter the audio and visual components of digital video, and they also frequently have malicious intentions. Deepfake algorithms are becoming increasingly complex, which means that there is an urgent need for methods that are reliable and robust in order to detect and prevent the manipulate media. The purpose of this paper is to provide a comprehensive examination of deepfake detection approaches, with a specific focus on strategies that integrate deep learning techniques. Deep learning, a type of machine learning that makes use of artificial neural networks with multiple layers, has shown promising results in a variety of computer vision applications. These applications include object recognition, image categorization, and natural language processing, among others. Deep learning approaches have the ability to differentiate between real and fake media in the context of deepfake detection. This is accomplished by recognizing intricate patterns and characteristics that are present in the data. This work also tackles the open research problems and challenges in the field of deepfake detection. These include the demand for rigorous assessment criteria, large-scale datasets, and the capacity to apply to manipulation techniques that have not yet been encountered. In addition, ethical concerns that are associated with the development and implementation of deepfake detection systems are highlighted. The importance of openness, confidentiality, and the responsible deployment of these technologies is emphasized throughout the discussion.

1.1 MOTIVATION

The profound influence that deepfakes have had on internet users since their discovery in 2016 is what inspired the development of deepfake detection. These intentionally produced fake media are being utilized more often in cybercrimes to disseminate false information and cause harm. People were frequently the victims of deepfake content prior to improvements in detection technologies, which resulted in losses in terms of money, privacy violations, and reputational harm.

Reactive tactics were the main method used for a long time to detect deepfake content. Content was typically discovered to be deepfake only after it had already been distributed through mass media, which frequently led to public protest and the removal of the offending material. Moreover, many devices were not able to use these protection mechanisms because the detection techniques that were available had poor precision and were mainly available to systems with considerable processing power.

Leading internet corporations including Google, Facebook, and Microsoft worked together to address the critical need for more efficient deepfake detection techniques. Together, they introduced a cutting-edge tech-based project called the Deepfake Detection Challenge. The goal of this challenge was to use the combined knowledge and information of these major players in the market to stimulate the creation of effective and user-friendly techniques for identifying deepfake content on social media platforms. These businesses hoped to enable the larger tech community to develop more reliable and readily available methods for identifying and reducing the negative impacts of deepfakes by combining their resources and knowledge.

Initiatives for deepfake detection, like the Deepfake Detection Challenge, are driven by goals other than safeguarding individual users. It includes defending the integrity of online information, preserving public confidence in digital media, and battling the growing threat of false information and cybercrimes. The objective is to equip users with the necessary capabilities to recognize and reduce the hazards associated with deepfake content through cooperative efforts and creative solutions, ultimately promoting a more secure and safe online environment for everybody.

1.2 INNOVATION

Combining sophisticated models like ResNext101_64x4d with Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) architectures is a novel method for detecting deepfakes. This approach seeks to increase scalability by lowering computing complexity while simultaneously improving deepfake detection accuracy. The method can effectively learn complex patterns within deepfake movies by utilizing the ResNext101_64x4d model, which is well-known for its effectiveness in handling large-scale image recognition problems, in conjunction with the sequential learning capabilities of RNN and LSTM.

The algorithm will be trained and evaluated against a dataset of lower complexity in order to expedite the development process and guarantee the system's efficacy. This method enables the development of a straightforward but reliable classifier algorithm that is readily deployable on a number of different platforms. The algorithm's accessibility and usability are guaranteed, even on devices with less processing power, by its emphasis on efficiency and simplicity.

The ultimate objective is to include this cutting-edge deepfake detection technology into an online application. The user-friendly interface of this web application will make it simple for users to upload films for analysis and get timely detection results. The focus on developing a web-based solution guarantees broad accessibility, allowing users of different technical proficiency levels to make use of the deepfake detection tool. This method encourages broader acceptance and deployment of the algorithm for detecting deepfake content across various internet platforms in addition to improving accessibility.

1.3 PRODUCT VISION STATEMENT

The goal is to develop a state-of-the-art DeepFake Detection System that will protect people, institutions, and society as a whole from the negative impacts of misrepresented media. Modern artificial intelligence and machine learning algorithms will be used in creating a system to identify, evaluate, and stop the spread of DeepFake content with previously unheard-of accuracy and speed.

The primary goal is to rebuild integrity and confidence in digital media by offering a strong barrier against the spread of false and misleading information. This will enable viewers to clearly distinguish between real and fake media by utilizing cutting-edge computer vision and deep learning algorithms, therefore halting the spread of misleading information.

In the future, the DeepFake Detection System can easily be included into news outlets, social networking sites, and video-sharing platforms, acting as an essential first line of defense against the ever-present danger of false information. In addition to shielding users against malevolent DeepFakes, this solution will help journalists, decision-makers, and content producers preserve the legitimacy and authenticity of their work.

The mission is to become the market leader in DeepFake detection by constant innovation and partnership with leading industry experts, making the internet a safer and more reliable place for everyone.

CHAPTER 2

BACKLOG REFINEMENT

2.1 PRODUCT BACKLOG

Given that manipulated media can mislead and deceive in today's digital ecosystem, developing robust DeepFake detection algorithms is imperative. Nonetheless, in order to develop robust and trustworthy detection systems, a number of issues need to be resolved. The need for broad and varied datasets, the difficulties in managing unlabeled data that are common in law enforcement and journalism, the significance of taking temporal consistency into account in video analysis, and the susceptibility to adversarial attacks are all highlighted in this product backlog. By addressing these issues, we hope to improve DeepFake detection technologies' precision and robustness and keep them at the forefront of defense against digital manipulation and disinformation.

During the training process, the performance of a DeepFake detection model is dependent on the number and variety of huge datasets that are utilized. If the model is evaluated on downloaded media that contains an unknown form of manipulation, then it will be difficult to create the model in such a way that it can identify the unknown type of manipulation. Because of the widespread use of web-based applications, postprocessing procedures are conducted to DeepFake multimedia with the purpose of deceiving the DeepFake detector. These manipulations may include the elimination of temporal artifices, blurring, smoothing, cropping, and other techniques.

DeepFake detection models are typically trained with huge datasets the majority of the time. On the other hand, as in the case of journalism or law enforcement-based DeepFake identification, there may be just a limited dataset accessible in certain circumstances. Additionally, the labeling of the score that corresponds to the sort of forgery that was utilized requires additional effort since this kind of dataset is of this nature. As a consequence of this, additional research is necessary in order to comprehend situations of forgeries that include journalists or law enforcement. Because of the black-box nature of DeepFake detection models, the majority of them do not provide such an explanation. This is especially true for models that are built on deep learning methodologies. Consequently, the process of developing a DeepFake detection model with a limited dataset that is not labeled is a difficult one.

The DeepFake detection algorithms that are now in use make use of binary frame-level classification. This classification works by identifying whether each individual video frame is real or virtual. On the other hand, because these methods do not take into account the temporal consistency between frames, they could run into problems, such as displaying temporal anomalies and having actual and artificial frames occur in consecutive periods. Furthermore, in order to reach the final result, these methods require an additional step to be taken in order to compute the video integrity score. As a result, the score must be integrated for each frame.

2.2 PRODUCT ROADMAP/RELEASE PLAN

Table 2.1 gives the product roadmap for the Deepfake Detection Project. The project basically includes of three major divisions: Data Collection and Preprocessing, Model Creation and Training and Prediction with Front-End Development

Table 2.1 Product Roadmap for Deepfake detection

YEAR	MILESTONES AND DEVELOPMENTS	DURATION
Dec 2023	• Gather diverse dataset containing real and fake videos • Ensure a balanced dataset of fake and real samples	2 Weeks
Jan 2024	• Implement preprocessing on the images • Apply face detection technique to create face cropped videos	3 Weeks
Feb 2024	• Set up ResNext101 architecture for feature extraction and LSTM for temporal analysis of video frames • Train initial models on subset of dataset and experimented with various hyperparameters and then train the model on entire dataset • Evaluate models based on accuracy, precision etc.	4 Weeks
Mar 2024	• Integrate trained model for deepfake detection and test prediction pipeline on unseen data • Design user interface for the deepfake detection system.	2 Weeks

2.3 HIGH-LEVEL ESTIMATION OF ALL EPICS

EPIC 1: Data Collection and Preprocessing

- Research existing deepfake datasets
- Gather labeled authentic and deepfake videos
- Preprocess collected data
- Create pipeline for data loading and preprocessing

EPIC 2: Model Creation and Training

- Research deep learning architectures
- Experiment with different models
- Train selected model on preprocessed data
- Evaluate model performance
- Fine-tune models for better accuracy

EPIC 3: Prediction and Front-End Development

- Develop prediction pipeline integrating trained model
- Implement front-end for user interaction
- Integrate prediction functionality with front-end
- Test end-to-end functionality

CHAPTER 3

SPRINT PLANNING

3.1 SPRINT 1: DATA COLLECTION AND PREPROCESSING

Sprint 1 marks the foundational phase of our deepfake detection project, focusing on Data Collection and Preprocessing. This sprint collects labeled authentic and deepfake videos, thoroughly examines deepfake datasets and tools already available, preprocesses the gathered data, and builds a reliable pipeline for preprocessing and data loading.

3.1.1 Capacity Plan for Sprint 1

Table 3.1 gives the Capacity Planning of Sprint 1 where we perform Data Collection and Data Preprocessing for the project.

Table 3.1 Capacity Planning of Sprint 1

		Capacity Plan for Sprint 1					
Name	**Role**	**Working Days**	**Planned Leaves (in Days)**	**Other Course work activities**	**Upskilling (in Days)**	**Design, Development, Testing, Documentation (in Days)**	**Estimated Hours**
Shresth Gupta	Data Scientist	20	10	3	3	15	30
Utkarsh Srivastava	Data Scientist	20	10	2	2	18	45
Shresth Gupta	Developer	20	10	2	3	18	45

During Sprint 1, a thorough capacity plan was developed to maximize team output. After accounting for planned leaves (2–3 days) and external coursework (2–3 days), this plan identifies team members (Shresth Gupta, Utkarsh Srivastava) and their roles (Data Scientist, Developer), as well as their available workdays (20). Upskilling activities are given priority in the plan, with 15–18 hours per team member set aside for continuous learning.

3.1.2 Detailed Estimation of User Stories

Table 3.2 gives the detailed estimation of user stories of Sprint 1 where we perform Data Collection and Data Preprocessing for the project.

Table 3.2 Detailed Estimation of User Stories of Sprint 1

ID		Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	**Sprint 1**	Data Collection	Data Collection and Preprocessing	As a user, I want to access a diverse dataset of labeled authentic and deepfake videos so that I can train a deepfake detection model.	Must	Ready	Dataset should have minimum 500 real and 500 fake videos, each video should be labeled clearly, videos should be of sufficient quality	1. Create scripts to automatically download and organize videos based on labels. 2. Provide a search or filtering mechanism for users to easily find videos based on labels.	1. Ensure the collected videos are stored securely to prevent unauthorized access. 2. Implement error handling and retry mechanisms for video downloads.	8 days	10 days
2		Data Preprocessing	Data Collection and Preprocessing	As a user, I want the collected videos dataset to be preprocessed for optimal training of the deepfake detection model.	Must	Ready	The system should the collected dataset, apply normalization techniques to standardize the data, convert the data into suitable format for training.	1.Convert video frames into arrays suitable for deep learning models (e.g., image tensors). 2.Save the preprocessed dataset in a specified format (e.g., CSV, HDF5).	Ensure efficient memory usage and processing speed during data loading and preprocessing.	20 days	15 Days

A critical component of project management, especially in the software development industry, is estimating user stories. Tasks can be broken down into specific user stories to help with understanding the scope of work involved, resource allocation, and realistic timescales when starting a project like developing a deepfake detection system.

Investigating currently available deepfake datasets and technologies is the first user story. To find pertinent statistics and tools that will form the project's foundation, this work entails searching the internet and scholarly sources. Recognizing what is already out there in the field is crucial to minimizing duplication and making the most of what is currently accessible. The user story of compiling identified real and deepfake videos comes next. The work at hand is searching actively for videos that the deepfake detection model will be trained and validated with. Ensuring a varied selection of genuine and deepfake videos is just as important as quantity when it comes to enhancing the model's accuracy and resilience.

Preprocessing the gathered data is the subject of the third user story. Data preparation, which includes sanitizing, organizing, and converting raw data into a format that can be used to train the model, is an essential stage in machine learning projects. The video will be divided into frames for this work, and then each frame will have its faces identified, cropped, and a video including the faces from each of the chosen frames will be produced.

The user story concludes with building a pipeline for preprocessing and data loading. Creating a methodical procedure or workflow to load the gathered data into the system, carry out the previously specified preparation actions, and get the data ready for model training is the main goal of this assignment.

3.1.3 Daily Scrum Activities

Table 3.3 gives the Daily Scrum Report for week 1 of Sprint 1 where we perform Data Collection and Data Preprocessing for the project. Table 3.4 gives the Daily Scrum Report for week 2 of Sprint 1 where we perform Data Collection and Data Preprocessing for the project. Table 3.5 gives the Daily Scrum Report for week 3 of Sprint 1 where we perform Data Collection and Data Preprocessing for the project. Table 3.6 gives the Daily Scrum Report for week 4 of Sprint 1 where we perform Data Collection and Data Preprocessing for the project. Table 3.7 gives the Daily Scrum Report for week 5 of Sprint 1 where we perform Data Collection and Data Preprocessing for the project.

Table 3.3 Daily Scrum Report for Week 1 of Sprint 1

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Read about deepfake detection	Read about deepfake detection	Read about deepfake detection	Explored various dataset for the project	Explored various dataset for the project
	What are doing today?	Gather information about the project	Gather relevant information about the project	Reading about the datasets for this project	Reading about the datasets for this project	Finalizing on the dataset that are to be used for the project
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Read about deepfake detection	Read about deepfake detection	Read about deepfake detection	Explored various dataset for the project	Explored various dataset for the project
	What are doing today?	Gather information about the project	Gather relevant information about the project	Reading about the datasets for this project	Reading about the datasets for this project	Finalizing on the dataset that are to be used for the project
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.4 Daily Scrum Report for Week 2 of Sprint 1

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Finalizing the dataset that are to be used	Finalizing the dataset that are to be used	Downloading the dataset for the project	Downloading the dataset for the project	Understanding the dataset and arranging the data
	What are doing today?	Finalizing the dataset that are to be used	Downloading the dataset for the project	Downloading the dataset for the project	Understanding the dataset and arranging the data	Understanding the dataset and arranging the data
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Finalizing the dataset that are to be used	Finalizing the dataset that are to be used	Downloading the dataset for the project	Downloading the dataset for the project	Understanding the dataset and arranging the data
	What are doing today?	Finalizing the dataset that are to be used	Downloading the dataset for the project	Downloading the dataset for the project	Understanding the dataset and arranging the data	Understanding the dataset and arranging the data
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.5 Daily Scrum Report for Week 3 of Sprint 1

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Understanding the dataset and arranging the data	Read about preprocessing of images	Read about preprocessing of images	Read about face detection algorithm and techniques	Reading about the demerits of already existing tools.
	What are doing today?	Read about preprocessing of images	Read about preprocessing of images	Read about face detection algorithm and techniques	Read about face detection algorithm and techniques	Read about face detection algorithm and techniques
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Understanding the dataset and arranging the data	Read about preprocessing of images	Read about preprocessing of images	Read about face detection algorithm and techniques	Read about face detection algorithm and techniques
	What are doing today?	Read about preprocessing of images	Read about preprocessing of images	Read about face detection algorithm and techniques	Read about face detection algorithm and techniques	Read about face detection algorithm and techniques
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.6 Daily Scrum Report for Week 4 of Sprint 1

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Read about face detection algorithm and techniques	Performed face detection and cropping on images	Learnt about HOG feature descriptor	Read about video preprocessing techniques	Performed preprocessing of videos by frame extraction
	What are doing today?	Performed face detection and cropping on images	Learnt about HOG feature descriptor	Read about video preprocessing techniques	Performed preprocessing of videos by frame extraction	Performed face detection and cropping on individual frames of the video
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Read about face detection algorithm and techniques	Performed face detection and cropping on images	Learnt about HOG feature descriptor	Read about video preprocessing techniques	Performed preprocessing of videos by frame extraction
	What are doing today?	Performed face detection and cropping on images	Learnt about HOG feature descriptor	Read about video preprocessing techniques	Performed preprocessing of videos by frame extraction	Performed face detection and cropping on individual frames of the video
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.7 Daily Scrum Report for Week 5 of Sprint 1

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Performed face detection and cropping on individual frames of the video	Combined the face cropped images of frames back into video	Combined the face cropped images of frames back into video	Performed the complete preprocessing of videos to the dataset	Performed the complete preprocessing of videos to the dataset
	What are doing today?	Combined the face cropped images of frames back into video	Combined the face cropped images of frames back into video	Performed the complete preprocessing of videos to the dataset	Performed the complete preprocessing of videos to the dataset	Performed the complete preprocessing of videos to the dataset
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Performed face detection and cropping on individual frames of the video	Combined the face cropped images of frames back into video	Combined the face cropped images of frames back into video	Performed the complete preprocessing of videos to the dataset	Performed the complete preprocessing of videos to the dataset
	What are doing today?	Combined the face cropped images of frames back into video	Combined the face cropped images of frames back into video	Performed the complete preprocessing of videos to the dataset	Performed the complete preprocessing of videos to the dataset	Performed the complete preprocessing of videos to the dataset
	Is there anything blocking you?	NO	NO	NO	NO	NO

The Daily Scrum report for week 1 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's initial stages. The two members of the team, continuously committed a substantial amount of time to reading and researching datasets related to the project's goals during the course of the week. Their commitment to producing high-quality results is demonstrated by their thorough understanding of the available data. They also put a lot of time and effort into learning about different deepfake detection methods so they could be prepared for the technical challenges of the project. The fact that there are no known blockers indicates that they are capable of handling their responsibilities and getting past potential roadblocks, even with the inherent difficulties in navigating datasets and investigating detection techniques. The team made significant progress during the week, and by the end of the week, they had almost finished the dataset selection process.

The Daily Scrum report for week 2 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's initial stages. Both teammates persisted in their dedication to the deepfake detection project, concentrating mainly on the acquisition and finalization of datasets. They were always working on improving the dataset selections, making sure that they were in line with the project goals, and downloading the datasets that they had found. Crucially, no roadblocks were reported during the course of the week, suggesting easy progress and skilful resolution of any difficulties that surfaced. The team made significant progress during the week, and by the end of the week, they had finished the acquisition of dataset.

The Daily Scrum report for week 3 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's initial stages. During the third week, the team members concentrated on improving their comprehension of face detection algorithms and image preprocessing methods by reading and researching extensively. They spent most of their days investigating different facets of image preprocessing techniques and face detection algorithms and methods in an effort to broaden their understanding and choose the best strategies for the demands of the project. Significantly, there were no blockers reported during the course of the week, highlighting the easy way in which their tasks progressed and their capacity to overcome obstacles.

The Daily Scrum report for week 4 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's initial stages. Both members of the team committed time to learning face detection methods and algorithms, putting them into practice by cropping and detecting faces in photos. They also studied feature descriptors known as Histogram of Oriented

Gradients (HOG), which are essential for identifying features in images. In order to prepare videos for analysis, they also delved deeper into the field of video preprocessing techniques, with a particular emphasis on frame extraction

The Daily Scrum report for week 5 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's initial stages. Both team members meticulously performed face detection and cropping on individual frames of the videos, ensuring that only relevant facial regions were extracted. The preprocessing pipeline was then finished by smoothly recombining the cropped face images of the frames into videos. Over the course of the week, they both worked to improve their preprocessing methods and make sure the dataset was ready for further analysis and model training. Interestingly, there were no reported roadblocks, pointing to easy progress and productive teamwork. With the completion of Week 5, the dataset has been fully preprocessed, allowing them to confidently move forward into the next stage of model development and evaluation.

3.1.4 Functional Documents

a. Introduction

Deepfake Detection using ResNext101 and LSTM is a software solution aimed at identifying manipulated or synthetic media content, commonly known as deepfakes. Leveraging advanced machine learning techniques, specifically ResNext101 convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) networks, this product offers robust detection capabilities to combat the proliferation of fake media in various contexts.

b. Product Goal

The product goal is to develop a robust and accurate deepfake detection system using deep learning techniques, specifically leveraging the ResNext architecture. The system aims to assist various stakeholders, including social media platforms, news agencies, and law enforcement agencies, in identifying and combating the proliferation of deepfake videos. By detecting deepfake content early, the system contributes to preserving the integrity of digital media and preventing misinformation.

c. Demography (Users, Location)

The target users of the product include social media platform moderators, content verification teams, news editors, journalists, and law enforcement agencies involved in combating digital manipulation and misinformation. The product will be applicable globally, catering to regions where the spread of deepfake content poses significant threats to societal trust and stability.

d. Business Processes

The business processes in Sprint 1 focus on data collection, preprocessing, and initial model experimentation. This includes sourcing relevant datasets for training the ResNext101 and LSTM models and preprocessing the data to prepare it for training,

e. Features

- Feature 1: Data Collection
  - Description: The Data Collection aggregates sources from which the real and deepfake films will be sourced, including publicly accessible datasets, web resources, and sources that have been specifically found through study.
  - User Story: In User Story, the user wants a diverse dataset of labeled authentic and deepfake videos to train a deepfake detection model.
- Feature 2: Data Preprocessing
  - Description: The Data Preprocessing offers a thorough rundown of the procedures necessary to get the gathered data ready for model training.
  - User Story: In User Story, the user wants the collected videos dataset to be preprocessed for optimal training of the deepfake detection model.

f. Authorization Matrix

Table 3.8 gives the authorization matrix for Sprint 1 in Data Collection and Preprocessing.

Table 3.8 Authorization Matrix for Sprint 1

Role	Permission
Administrator	Administrators have full access to all system functionalities.
User	Users can use the deepfake detection system
Analyst	Analysts have access to both detection features and advanced analytics tools.

g. Assumptions

- The availability of a diverse and well-labelled dataset for training the ResNext model on deepfake detection.
- Sufficient computational resources for Data Preprocessing

3.1.5 Architecture Diagram

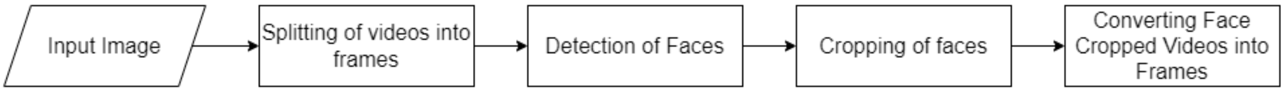

Input Image
Splitting of videos into frames
Detection of Faces
Cropping of faces
Converting Face Cropped Videos into Frames

Fig 3.1 Architecture Diagram for Preprocessing of Videos

Fig 3.1 shows the architecture diagram of the complete preprocessing of videos starting from splitting of frames, detection of faces, cropping of faces and then converting then back to videos.

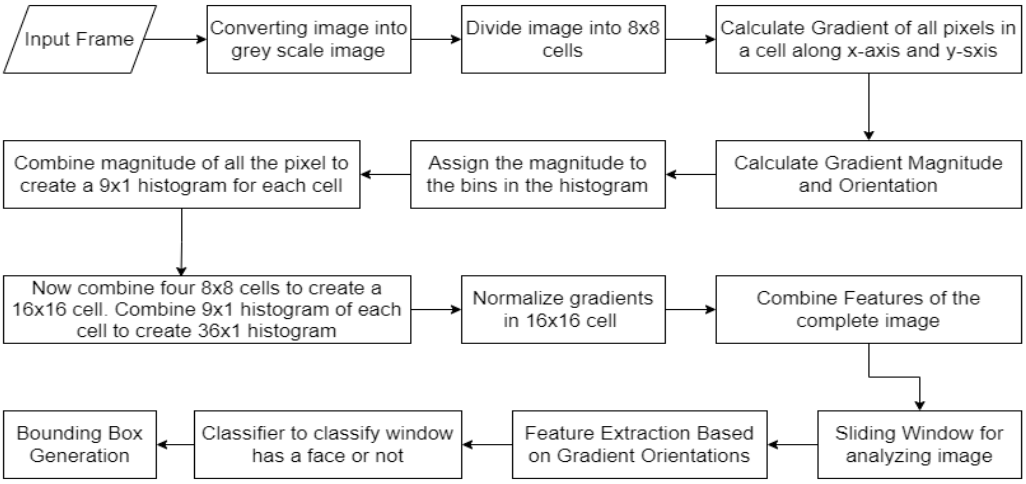

Input Frame
Converting image into grey scale image
Divide image into 8x8 cells
Calculate Gradient of all pixels in a cell along x-axis and y-sxis
Calculate Gradient Magnitude and Orientation
Assign the magnitude to the bins in the histogram
Combine magnitude of all the pixel to create a 9x1 histogram for each cell
Now combine four 8x8 cells to create a 16x16 cell. Combine 9x1 histogram of each cell to create 36x1 histogram
Normalize gradients in 16x16 cell
Combine Features of the complete image
Sliding Window for analyzing image
Feature Extraction Based on Gradient Orientations
Classifier to classify window has a face or not
Bounding Box Generation

Fig 3.2 Architecture Diagram for Face Detection

Fig 3.2 shows the Architecture Diagram of a face detection algorithm using HOG. HOG, or Histogram of Oriented Gradients, is a feature descriptor that is often used to extract features from image data. It is widely used in computer vision tasks for object detection.

3.1.6 Architecture Document

An elaborate preprocessing pipeline has been constructed to get the videos ready for examination of deepfake detection. The purpose of this pipeline is to extract facial regions of interest (ROIs) from unprocessed video footage. This is an essential step in making sure that the deepfake detection algorithms that follow are accurate and efficient. The OpenCV library is used to extract frames from the input video at the beginning of the procedure. The face_recognition library's face detection technique is then applied to each frame. Frames are processed in batches of four consecutive frames, with the goal of maximizing computational efficiency.

In order to precisely identify faces inside an image, a sequence of processes called Histogram of

Oriented Gradients (HOG) face detection is used. To make processing easier, the input frame is first transformed into a greyscale image. Next, the image is separated into tiny cells, each measuring 8 by 8 pixels. The gradient of every pixel in each of these cells is determined along the x and y axes. These gradients represent the change in pixel intensities in the horizontal and vertical direction. This gradient information is essential for highlighting the image's contours and edges.

$$\boldsymbol{G_x} = \boldsymbol{I(x+1,y)} – \boldsymbol{I(x-1,y)} \tag{3.1}$$

$$\boldsymbol{G_y} = \boldsymbol{I(x,y+1)} – \boldsymbol{I(x,y-1)} \tag{3.2}$$

Eq 3.1 and 3.2 show the calculation of gradient along the x and y axis. Next, for every pixel in the cell, the gradient orientations and magnitudes are calculated.

$$\boldsymbol{G} = \sqrt{\boldsymbol{G_x}^2 + \boldsymbol{G_y}^2} \tag{3.3}$$

$$\boldsymbol{\Phi} = \tan^{-1}(\boldsymbol{G_y}/\boldsymbol{G_x}) \tag{3.4}$$

Eq 3.3 and 3.4 shows the calculation of Gradient Magnitude and Orientation. Based on their orientations, these magnitudes are subsequently allocated to bins in a histogram. In essence, this procedure records the gradient distribution in various directions inside the cell. As a result, each 8x8 cell produces a 9-bin histogram, which shows the gradient orientation. Next, a larger 16x16 cell is created by combining four adjacent 8x8 cells. For the 16x16 cell, the histograms of these four cells are combined to form a single 36x1 histogram. After that, these gradients inside the 16x16 cell are normalized. It is imperative to do this normalizing phase to make sure that the image's lighting variations have no appreciable impact on the detecting process.

After computing and normalizing the gradients for the full image, a sliding window method is used in the procedure. Using the previously computed histograms, a feature vector based on gradient orientations is recovered for each window position while a 64x128 pixel window glides across the image. The essential details regarding the window's boundaries and textures are contained in this feature vector.

A classifier that has been trained to discern between windows that have faces and those that do not is then fed these feature vectors. Neural networks or support vector machines (SVMs) are common classifiers for this kind of task. Each feature vector is examined by the classifier, which then determines whether or not a face is likely to be present in the relevant window.

Lastly, a bounding box is created around a window that has been identified as having a face in order to indicate where the face was found inside the picture. For tasks like facial recognition or tracking, this bounding box provides important information about the size and location of the recognized face. To reliably identify faces in photos, the HOG-based face recognition method integrates gradient analysis, histogram computing, feature extraction, and classification.

This method guarantees the correctness and integrity of the final dataset, which is essential for trustworthy training of deepfake detection models. The project intends to standardize and optimize the input data for ensuing deepfake detection algorithms by putting this preprocessing methodology into practice. The prepared preprocessing method is a keystone of the deepfake detection system, enabling efficient detection and reduction of artificial intelligence (AI)-generated alterations in visual content.

3.1.7 Functional Test Case Document

Table 3.9 gives the functional test case document of Sprint 1 where we perform Data Collection and Data Preprocessing for the project.

Table 3.9 Functional Test Case Table of Sprint 1

Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status
Data Collection	Gather real and fake videos from the dataset	1. Read about various types of datasets available 2. Download the dataset from the web 3. Examine the videos in the dataset and understand the data	Dataset downloaded successfully	Dataset downloaded successfully but some of the videos are corrupted	Passed
Data Preprocessing	To preprocess the videos in the dataset for the model	1. Research about preprocessing of videos 2. Extraction of frames from the videos 3. Detection of faces from the videos and cropping these faces 4. Converting face cropped frames back into videos	Faces are detected successfully and face cropped frames are converted back to videos.	1. Only a single face is detected and cropped from a video with multiple faces 2. Videos is created successfully of face cropped frames	Passed

The Functional Test Case Report for Sprint 1 outlines two key test cases: Data Collection and Data Preprocessing.

- The test case for data collection entails obtaining both real and fake videos from the dataset through a series of procedures that include looking up different kinds of datasets, downloading the dataset, and watching the videos. Although the dataset was successfully downloaded, which was the expected result, some videos were found to be corrupted.
- The Data Preprocessing test case concentrates on actions such as investigating preprocessing techniques, obtaining frames, identifying and resizing faces, and repurposing frames as videos. Even though face detection and video creation worked as planned, there were a few problems, like the generated videos having fewer frames than expected and only one face being recognized from videos with multiple faces.

Although both test cases passed overall, certain areas needed to be improved upon and optimized in upcoming sprints.

3.1.8 Sprint Retrospective

Table 3.10 gives the Sprint Retrospective of Sprint 1 where we perform Data Collection and Data Preprocessing for the project.

Table 3.10 Sprint Retrospective of Sprint 1

Liked	Learned	Lacked	Longed For
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
Efficient identification of various researches made into deepfake detection and exploring the web in search of deepfake detection datasets and the various technologies existing to identify deepfakes.	One of the key lessons learned was the significant effort and time required to collect authentic and deepfake videos. It taught us the importance of meticulous planning and resource allocation when dealing with complex datasets.	Access to comprehensive and labeled datasets specifically designed for deepfake detection proved to be challenging.	The team longed for additional resources to explore more datasets available and more balanced datasets to work on.

Liked	Learned	Lacked	Longed For
The implementation of video processing techniques like frame extraction, face detection and cropping	The preprocessing step revealed the necessity of implementing robust data cleaning and normalization techniques to ensure the quality and consistency of collected data.	The team lacked sufficient resources and expertise to develop data augmentation techniques limiting our ability to enhance diversity of the dataset	There was a longing for smoother integration of data loading and preprocessing pipelins to streamline workflows and increase efficiency.

During the Sprint 1 retrospective, a number of factors were identified as having positively impacted our advancement. The proficient identification of diverse research and technologies associated with deepfake detection, in conjunction with the examination of datasets, exemplified our team's proactive methodology and unwavering dedication to remaining up-to-date with industry developments. Furthermore, the effective application of video processing methods like face detection and frame extraction demonstrated our technological prowess and willingness to take on challenging assignments. But this sprint also taught us some important lessons. We discovered how crucial it is to plan ahead and allocate resources carefully, especially when gathering real and deepfake videos. This underscores the need for better techniques to handle complicated datasets. In addition, difficulties were found, mainly with regard to gaining access to extensive and labeled datasets created especially for the purpose of detecting deepfakes. Resource limitations also hindered the development of data augmentation methods. In order to improve workflow efficiency, the team expressed a wish for more seamless integration of pipelines for preprocessing and data loading in the future. Overall, as we move forward with our deepfake detection project, the retrospective from Sprint 1 has given us important insights for improving our technical capabilities, streamlining our processes, and resolving resource constraints.

3.2 SPRINT 2: MODEL CREATION AND TRAINING

In Sprint 2, our focus shifts towards Model Creation, a pivotal phase in our deepfake detection project. This phase explores the creation and testing of deep learning architectures specifically designed for deepfake detection. This sprint's objectives include investigating deep learning architectures that meet our requirements, training a selection of models on preprocessed data, assessing their effectiveness, and optimizing their accuracy. The goal of this sprint is to develop a strong and effective deepfake detection model by utilizing the capabilities of LSTM and ResNext101 architectures.

3.2.1 Capacity Plan for Sprint 2

Table 3.11 gives the Capacity Planning of Sprint 2 where we perform Model Creation and Training for the project.

Table 3.11 Capacity Planning of Sprint 2

		Capacity Plan for Sprint 2					
Name	**Role**	**Working Days**	**Planned Leaves (in Days)**	**Other Course work activities**	**Upskilling (in Days)**	**Design, Development, Testing, Documentation (in Days)**	**Estimated Hours**
Shresth Gupta	Data Scientist	20	10	3	2	15	45
Utkarsh Srivastava	AI Engineer	20	10	3	2	15	45
Shresth Gupta	Developer	20	10	3	3	18	54

During Sprint 2, a thorough capacity plan was developed to maximize team output. After accounting for planned leaves (2–3 days) and external coursework (2–3 days), this plan identifies team members (Shresth Gupta, Utkarsh Srivastava) and their roles (Data Scientist, Developer, AI Engineer), as well as their available workdays (20). Upskilling activities are given priority in the plan, with 15–18 hours per team member set aside for continuous learning. Both of the team members are estimated to work for around 45 to 54 hours. This capacity plan reflects a balanced allocation of time and resources to ensure that the team can effectively manage their workload.

3.2.2 Detailed Estimation of User Stories

Table 3.12 gives the detailed estimation of user stories of Sprint 2 where we perform Model Creation and Training for the project.

Table 3.12 Detailed Estimation of User Stories of Sprint 2 (continued)

ID		Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	**Sprint 2**	Model Training	Model Creation and Training	As a user, I want to train deep learning models for detecting deepfake videos.	Must	Ready	Experiment various deep learning architectures and evaluate their performance	1. Implement deep learning architectures such as ResNext101_64x4d, RNN, LSTM, etc. 2. Train models using appropriate loss functions and optimization algorithms.	1. Ensure efficient model training with reasonable time and resource consumption. 2. Aim for high accuracy in deepfake detection to minimize false positives and negatives.	10 days	12 days
2		Model Fine-Tuning	Model Creation and Training	As a user, I want to fine-tune the selected model to further improve its performance in deepfake detection.	Must	Ready	Fine-tune the model by implementing hyperparameter optimization and save the fine-tuned model	1. Implement hyperparameter optimization techniques (e.g., grid search, random search). 2. Fine-tune the selected model using the validation dataset. 3. Save the fine-tuned model and its optimized hyperparameters.	1. Ensure the fine-tuned model shows improvement in detection accuracy. 2. Optimize hyperparameters for faster inference without sacrificing accuracy.	3 days	5 Days

Table 3.12 Detailed Estimation of User Stories of Sprint 2

ID		Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
3	**Sprint 2**	Model Evaluation and Selection	Model Creation and Training	As a user, I want to evaluate trained models and select the best one for deepfake detection.	Must	Ready	Evaluate the model on a validation dataset, compare model performance using accuracy, precision.	1. Load the trained models and the validation dataset. 2. Evaluate models using appropriate evaluation metrics. 3. Compare performance metrics across different models. 4. Select the model with the highest overall performance.	1. Ensure the selected model has high accuracy in detecting deepfake videos. 2. Select a model that balances performance with computational resources.	3 days	3 days

In Sprint 2, which is centered on Model Creation for the deepfake detection system, segmenting the work into comprehensive user stories facilitates comprehension of the task scope, effective resource allocation, and establishment of reasonable completion deadlines.

Investigating deep learning architectures for deepfake detection is the initial user narrative. Finding and comprehending the different neural network topologies that function well for identifying deepfake movies depends on completing this challenge. To guide the next steps of model experimentation and selection, research on architectures such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) customized for this particular purpose is crucial.

The user story about testing out several models comes next. This assignment uses the preprocessed data from Sprint 1 to create and train multiple deep learning models. The most efficient model for deepfake detection will be determined by the data scientists through experimentation with various settings, hyperparameters, and architectures. The purpose of this testing step is to determine which model works best with the given dataset.

The next step in the user journey is to train the chosen model using the preprocessed data. The group will set aside time to train these models on the preprocessed dataset after experimenting to determine which model is best. In order to make sure the model is learning and performing better, this stage entails feeding the data into the chosen models, modifying the parameters, and keeping an eye on the training process.

The assessment of model performance is the fourth user story. It's crucial to carefully assess the models' performance using measures like accuracy, precision, recall, and F1-score once they've been trained. This assessment phase aids in determining the models' accuracy in differentiating between real and deepfake films. It also sheds light on potential areas for model enhancement or adjustment.

3.2.3 Daily Scrum Activities

Table 3.13 gives the Daily Scrum Report for week 1 of Sprint 2 where we perform Model Creation and Training for the project. Table 3.14 gives the Daily Scrum Report for week 2 of Sprint 2 where we perform Model Creation and Training for the project. Table 3.15 gives the Daily Scrum Report for week 3 of Sprint 2 where we perform Model Creation and Training for the project. Table 3.16 gives the Daily Scrum Report for week 4 of Sprint 2 where we perform Model Creation and Training for the project.

Table 3.13 Daily Scrum Report for Week 1 of Sprint 2

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Research about various Deep Learning architectures	Research about various Deep Learning architectures	Research about various Deep Learning architectures	Research about ResNext101	Research about ResNext101 and LSTM
	What are doing today?	Research about various Deep Learning architectures	Research about various Deep Learning architectures	Research about ResNext101	Research about ResNext101 and LSTM	Research about ResNext101 and LSTM
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Research about various Deep Learning architectures	Research about various Deep Learning architectures	Research about various Deep Learning architectures	Research about ResNext101	Research about ResNext101 and LSTM
	What are doing today?	Research about various Deep Learning architectures	Research about various Deep Learning architectures	Research about ResNext101	Research about ResNext101 and LSTM	Research about ResNext101 and LSTM
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.14 Daily Scrum Report for Week 2 of Sprint 2

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Research about ResNext101 and LSTM	Splitting the data and loading into batches	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM
	What are doing today?	Splitting the data and loading into batches	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Research about ResNext101 and LSTM	Splitting the data and loading into batches	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM
	What are doing today?	Splitting the data and loading into batches	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM	Implementation of ResNext101 and LSTM
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.15 Daily Scrum Report for Week 3 of Sprint 2

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Implementation of ResNext101 and LSTM	Training of ResNext101 and LSTM model on a small dataset	Evaluated the training on a small dataset	Changed the hyperparameters of the model and retrained the model	Changed the hyperparameters of the model and retrained the model
	What are doing today?	Training of ResNext101 and LSTM model on a small dataset	Evaluated the training on a small dataset	Changed the hyperparameters of the model and retrained the model	Changed the hyperparameters of the model and retrained the model	Finalized the hyperparameters of the model and retrained the model
	Is there anything blocking you?	Made use of Google Colab for training	Made use of Google Colab for training	Made use of Google Colab for training	Made use of Google Colab for training	Made use of Google Colab for training
Utkarsh Srivastava	**What did you do yesterday?**	Implementation of ResNext101 and LSTM	Training of ResNext101 and LSTM model on a small dataset	Evaluated the training on a small dataset	Changed the hyperparameters of the model and retrained the model	Changed the hyperparameters of the model and retrained the model
	What are doing today?	Training of ResNext101 and LSTM model on a small dataset	Evaluated the training on a small dataset	Changed the hyperparameters of the model and retrained the model	Changed the hyperparameters of the model and retrained the model	Finalized the hyperparameters of the model and retrained the model
	Is there anything blocking you?	Made use of Google Colab for training	Made use of Google Colab for training	Made use of Google Colab for training	Made use of Google Colab for training	Made use of Google Colab for training

Table 3.16 Daily Scrum Report for Week 4 of Sprint 2

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Finalized the hyperparameters of the model and retrained the model	Started training the model on entire dataset available	Started training the model on entire dataset available	Evaluation of the model based on accuracy, confusion matrix	Complete documentation of the entire project
	What are doing today?	Started training the model on entire dataset available	Started training the model on entire dataset available	Evaluation of the model based on accuracy, confusion matrix	Complete documentation of the entire project	Complete documentation of the entire project
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Finalized the hyperparameters of the model and retrained the model	Started training the model on entire dataset available	Started training the model on entire dataset available	Evaluation of the model based on accuracy, confusion matrix	Complete documentation of the entire project
	What are doing today?	Started training the model on entire dataset available	Started training the model on entire dataset available	Evaluation of the model based on accuracy, confusion matrix	Complete documentation of the entire project	Complete documentation of the entire project
	Is there anything blocking you?	NO	NO	NO	NO	NO

The Daily Scrum report for week 1 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's development stages. The daily activities of both teammates during the first week of Sprint 2 focused on delving into the complexities of these architectures and comprehending their advantages, disadvantages, and possible uses within the framework of the deepfake detection project. Notably, both team members painstakingly followed through on their research assignments without running into any obstacles, suggesting easy progress and productive teamwork.

The Daily Scrum report for week 2 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's development stages. The two team members started the second week of Sprint 2 by dividing the data into training and testing sets and loading the data into batches with the appropriate classifications. They then committed the rest of the week to putting ResNext101 and LSTM architectures into practice. Notwithstanding the intricacy of the assignment, neither team member faced any obstacles, suggesting seamless advancement and efficient cooperation.

The Daily Scrum report for week 3 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's development stages. During the third week of Sprint 2, both team members persisted in concentrating on putting ResNext101 and LSTM models for deepfake detection into practice and training them. They trained these models on a small dataset at the start of the week, and then they assessed the training outcomes. To increase performance, they retrained the models and iteratively changed their hyperparameters during the course of the week. They had trouble conducting training sessions on their local systems, but they managed to use Google Colab to conduct training sessions successfully, guaranteeing continuous progress. Both team members completed the last round of model training and adjusted the hyperparameters towards the end of the week. Despite the early difficulties, their tenacity and flexibility helped them get over problems and make great strides toward the advancement of the deepfake detection project. The team is now ready to proceed with the evaluation and optimization phases, bringing them closer to accomplishing project objectives, with the models trained and hyperparameters finalized.

The Daily Scrum report for week 4 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's development stages. During the last week of Sprint 2, the team members concentrated on completing the documentation for the deepfake detection project and wrapping up model training and evaluation. They started the week by completing the model's hyperparameters and starting training on the complete dataset. They kept training the models on the entire dataset during the course of the week, and they evaluated the models using confusion matrices and accuracy metrics to get a thorough understanding of their performance. They also took the time to carefully record the project, making sure that all procedures, approaches, and results were well-recorded for later use and distribution. Notwithstanding the heavy workload, neither team member ran into any obstacles, facilitating easy progress and efficient task completion. By the end of the week, both of their efforts had paid off, and the deepfake detection project had successfully undergone training, evaluation, and documentation, representing a critical turning point in its evolution.

3.2.4 Functional Documents

a. Introduction

Deepfake Detection using ResNext101 and LSTM is a software solution aimed at identifying manipulated or synthetic media content, commonly known as deepfakes. Leveraging advanced machine learning techniques, specifically ResNext101 convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) networks, this product offers robust detection capabilities to combat the proliferation of fake media in various contexts.

b. Product Goal

The product goal is to develop a robust and accurate deepfake detection system using deep learning techniques, specifically leveraging the ResNext architecture. The system aims to assist various stakeholders, including social media platforms, news agencies, and law enforcement agencies, in identifying and combating the proliferation of deepfake videos. By detecting deepfake content early, the system contributes to preserving the integrity of digital media and preventing misinformation.

c. Demography (Users, Location)

The target users of the product include social media platform moderators, content verification teams, news editors, journalists, and law enforcement agencies involved in combating digital manipulation and misinformation. The product will be applicable globally, catering to regions where the spread of deepfake content poses significant threats to societal trust and stability.

d. Business Processes

The business processes in Sprint 2 focus on model creation, model training and evaluation. This includes creating a ResNext101 and LSTM model with finetuned hyperparameters, and training the model on the available dataset and evaluating the model performance.

e. Features

- Feature 1: Model Development
  - Description: The Model Development involves the process of using ResNext101 and LSTM model to create a model that can be used to train on real and fake videos by learning intricate patterns from these videos.
  - User Story: In User Story, the user wants a deepfake detection model that is able to classify between real and fake videos.
- Feature 2: Hyperparameter Tuning
  - Description: In Hyperparameter Tuning, the model is fine-tuned by choosing hyperparameter values that help in increasing the model performance.
  - User Story: In User Story, the user wants a deepfake detection model that is able to classify between real and fake videos.
- Feature 3: Model Evaluation
  - Description: The Model Evaluation involves selecting a correct metric for evaluating the model training and also how much accurate is our model in predicting the videos.
  - User Story: In User Story, the user wants a deepfake detection model that is able to classify between real and fake videos

f. Authorization Matrix

Table 3.17 gives the authorization matrix for Sprint 2 in Model Creation and Training.

Table 3.17 Authorization Matrix for Sprint 2

Role	Permission
Administrator	Administrators have full access to all system functionalities.
User	Users can use the deepfake detection system
Analyst	Analysts have access to both detection features and advanced analytics tools.

g. Assumptions

- The availability of a diverse and well-labelled dataset for training the ResNext model on deepfake detection.
- Sufficient computational resources for Model Training and Evaluation

3.2.5 Architecture Diagram

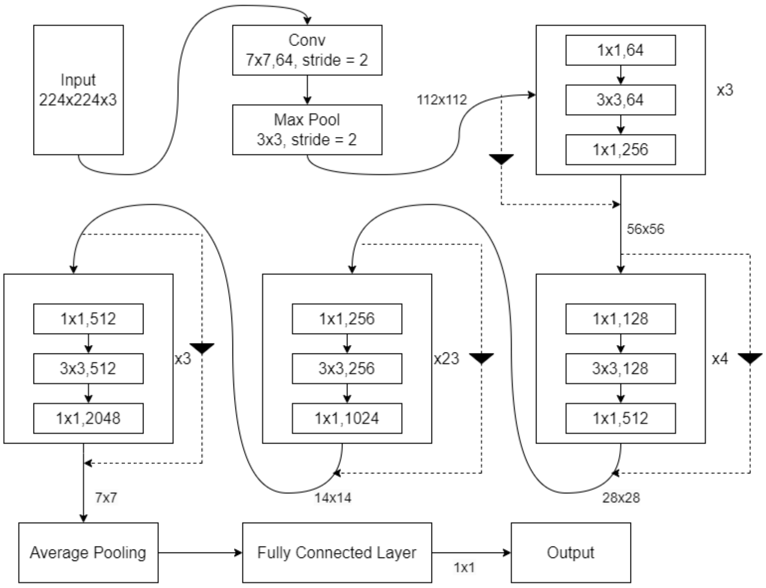


Fig 3.3 Architecture of ResNext101 Model

Fig 3.3 shows the architecture diagram of the ResNext101 model. Convolutional neural network (CNN) architecture ResNeXt101-64*4d is a member of the ResNeXt (Residual Next) family of models, which was first presented by Facebook AI Research (FAIR) researchers. The widely used ResNet (Residual Network) architecture is expanded upon by it. ResNeXt models use a "cardinality" parameter in addition to depth and width to enhance the scalability and performance of deep neural networks.

101: This refers to the total number of layers in the network, including convolutional layers, pooling layers, and fully connected layers. Specifically, it has 101 layers.

64: This represents the "width" of the network. In ResNeXt models, the concept of "cardinality" is introduced, which can be thought of as the number of paths or "groups" within a layer. In this case, the cardinality is 64, meaning that the network has 64 groups of paths within each block.

4d: The "d" here refers to the bottleneck design of each residual block. The bottleneck design is a feature of ResNet-style architectures where each block consists of three convolutional layers: 1x1, 3x3, and 1x1, where the 1x1 layers are used to reduce and then increase (restore) dimensions, and the 3x3 layer maintains the feature map size. The "4d" suggests that this design is repeated four times within each block.

The input image that the model uses is usually 224 by 224 pixels in size, which is the standard size for ImageNet benchmarks. Basic operations including activation functions, batch normalization, and convolution are typically found in the first layers of a network. The low-level features in the input image are extracted by these layers. Residual blocks comprise the fundamental building blocks of ResNeXt101-64*4d. Three convolutional layers—1x1, 3x3, and 1x1—are stacked in each residual block; the 1x1 layers are utilized to both enhance and decrease dimensionality. The 3x3 convolutional layer is divided into 64 groups according to the cardinality parameter, which in this case is set at 64. The convolutional operation individually learns several features within each group. The 4d indicates that each residual block contains four repetitions of this block structure. The vanishing gradient issue is lessened by the skip connection, also known as the identity shortcut connection, which permits the gradient to flow more freely during training. Because ResNeXt uses a widening factor of 64, the number of feature maps, or channels, may expand as we move across the network. The use of periodic pooling layers, like max pooling, lowers spatial dimensions and manages overfitting. There are fully linked layers and a softmax layer for classification jobs at the end of the network. These layers map the high-level features that were discovered by the preceding layers to the classes present in the dataset.

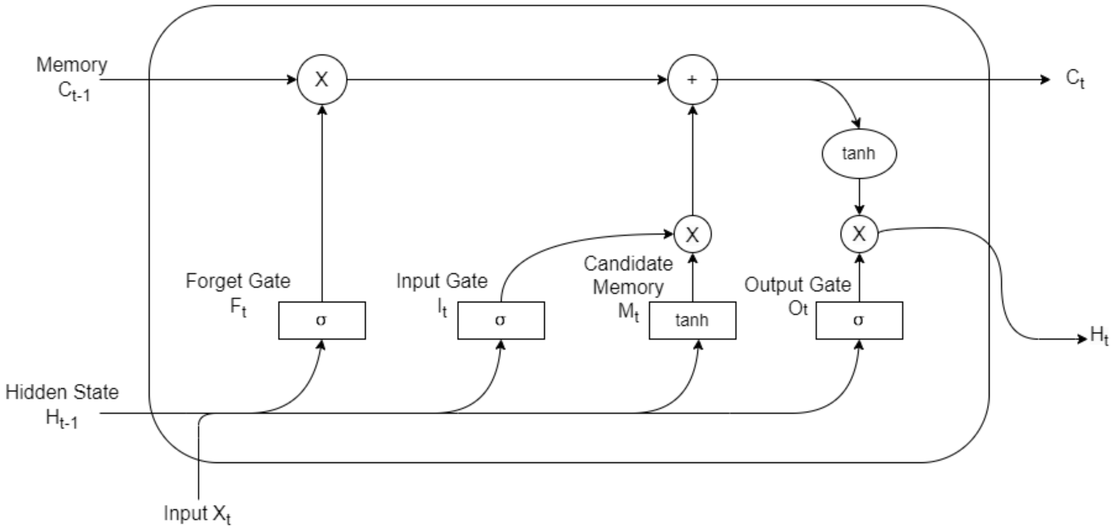


Fig 3.4 Architecture Diagram of LSTM

Fig 3.4 shows the architecture diagram of the LSTM. Recurrent neural network (RNN) architecture known as Long Short-Term Memory (LSTM) was created to solve the vanishing and exploding gradient issues that might arise during regular RNN training. When it comes to sequence prediction tasks where context and long-range dependency memory are essential, such language modeling, speech recognition, translation, and other applications, LSTMs are especially useful.

Components of LSTM:

Cell State (Ct): This is the core idea behind LSTM. The cell state runs straight down the entire chain, with only some minor linear interactions. It acts as a conveyor belt, carrying relevant information across timesteps.

Hidden State (ht): Similar to the hidden state in traditional RNNs, the LSTM's hidden state also carries information from previous timesteps. However, due to the gating mechanisms, it has more control over what to keep and what to discard.

Gates: LSTMs have three gates to control the flow of information:

- Forget Gate: Decides what information to throw away from the cell state. It looks at the previous hidden state h_{t-1} and the current input x_t, outputs a number between 0 and 1 for each number in the cell state C_{t-1}, where 0 means "completely forget" and 1 means "completely keep".
- Input Gate: Decides what new information to store in the cell state. It consists of two parts:
    - A sigmoid layer called the "input gate layer" that decides which values to update.
    - A tanh layer that creates a vector of new candidate values, M_t, to add to the cell state.
- Output Gate: Decides what to output based on the cell state. It controls the extent to which the cell state should be exposed to the network. It has two parts:
    - A sigmoid layer that determines which parts of the cell state are going to be output.
    - The cell state is passed through a tanh function and then multiplied by the output of the sigmoid gate to produce the hidden state h_t .

LSTM Operation:

1. Forget Gate Layer:

$$\boldsymbol{f_t = \sigma\,(W_f\,.\,[h_t - 1, x_t]\, +\, b_f)} \tag{3.5}$$

Eq 3.5 shows the calculation of output of forget gate at time step t where σ is the sigmoid function, W_f are the weights for the forget gate, h_{t-1} is the previous hidden state, x_t is the current input and b_f is the bias.

2. Input Gate Layer:

$$\boldsymbol{i_t = \sigma\,(W_i\,.\,[h_{t-1}, x_t]\, +\, b_i)} \tag{3.6}$$

$$\boldsymbol{M_t = tanh(W_c\,.\,[h_{t-1}, x_t]\, +\, b_c)} \tag{3.7}$$

Eq 3.6 and 3.7 shows the calculation of input gate activation and new candidate cell state where i_t is the input gate, W_i are the weights for the input gate, M_t is the candidate cell state, W_C are the weights for the candidate cell state, and b_i and b_C are biases.

3. Update Cell State:

$$\boldsymbol{C_t = f_t * C_{t-1} + i_t * M_t} \tag{3.8}$$

Eq 3.8 shows the calculation for updating the cell state. The forget gate decides how much of the previous cell state to retain, and the input gate decides how much of the candidate values to add to the new cell state.

4. Output Gate Layer:

$$\boldsymbol{o_t = \sigma\,(W_o\,.\,[h_{t-1}, x_t]\, +\, b_o)} \tag{3.9}$$

$$\boldsymbol{h_t = o_t * tanh(C_t)} \tag{3.10}$$

Eq 3.9 and Eq 3.10 shows the output gate activation at time step t and the final hidden state output at time t where o_t is the output gate, W_o are the weights for the output gate, and b_o is the bias and h_t is the final hidden state output.

3.2.6 Architecture Document

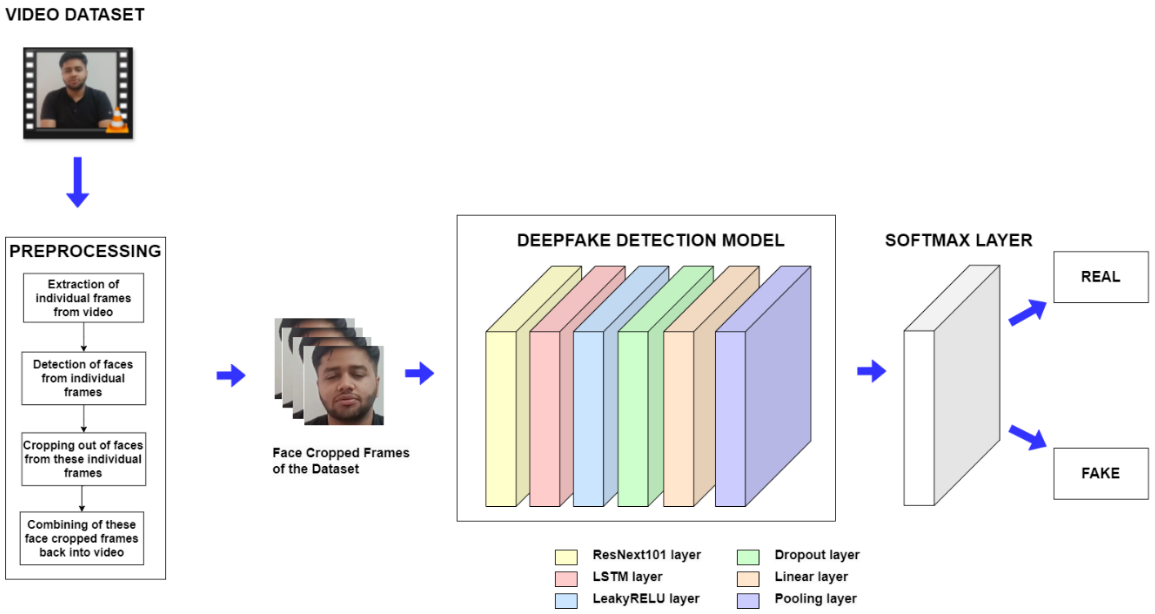

VIDEO DATASET
PREPROCESSING
Extraction of individual frames from video
Detection of faces from individual frames
Cropping out of faces from these individual frames
Combining of these face cropped frames back into video
Face Cropped Frames of the Dataset
DEEPFAKE DETECTION MODEL
SOFTMAX LAYER
REAL
FAKE
ResNext101 layer
LSTM layer
LeakyRELU layer
Dropout layer
Linear layer
Pooling layer

Fig 3.5 Architecture Diagram of Proposed Model

Fig 3.5 shows the proposed architecture diagram of deepfake detection model. It is a sophisticated architecture designed to effectively discern between genuine and manipulated visual content. Leveraging a powerful Residual Network Convolutional Neural Network (CNN) backbone, specifically the ResNeXt-101 64x4d model pretrained on large-scale image datasets, the model exhibits a robust capacity to extract high-level features from input videos. This ResNeXt-101 backbone is employed to capture intricate patterns and details within the frames, facilitating an in-depth understanding of the visual content. Following the ResNeXt-101 backbone, the architecture incorporates a Long Short-Term Memory (LSTM) layer, a pivotal component for temporal modeling. With the ability to process sequential data, the LSTM layer is crucial for capturing the temporal dependencies and nuances present in video sequences. This temporal modeling aspect enables the model to discern subtle changes and inconsistencies across frames, a critical capability for detecting the nuanced manipulations characteristic of deepfake videos. The architecture includes a dropout layer with a dropout rate of 0.4 and LeakyReLU activation functions for non-linearity toimprove the model's learning ability and avoid overfitting. By using regularization approaches, the risk of learning the training samples by heart is reduced and the model is guaranteed to perform well when applied to new data. A feature map and a classification output are the two parts of the model's output. The ResNeXt-101's extracted features are represented by the feature map. A linear layer and a dropout layer that processes the features and generates a classification score for every video also contribute to the classification output. The possibility that the input video is afake or authentic one is indicated by the final classification score.

3.2.7 Functional Test Case Document

Table 3.18 gives the functional test case document of Sprint 2 where we perform Model Creation and Training for the project.

Table 3.18 Functional Test Case Table of Sprint 2

Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status
ResNext101 and LSTM Model Development	Model Initialization	Load the pre-trained ResNext model Train the model on the dataset and evaluate its performance	Successful initialization of ResNext101 and LSTM Model. High accuracy	Model trained successfully with high accuracy	Passed
Hyperparameter Tuning	Optimal hyperparameter to be chosen for the model	Change the values of the hyperparameter and observe the changes in its performance and choose the optimal values	Correct values of hyperparameters are chosen for which the loss in minimum and accuracy is high	Model trained successfully with high accuracy	Passed
Model Evaluation	Evaluate Model Performance	Choose the required evaluation metric for the model and use it to check for the performance of the model	Accurate evaluation metrics reflecting the model's performance in distinguishing between real and deepfake images.	Evaluation metrics obtained accurately, reflecting the model's performance.	Passed

The Functional Test Case Report for Sprint 2 outlines three key test cases: Model Initialization, Hyperparameter Tuning and Model Evaluation.

- The successful initialization of the LSTM model and ResNext101 is the main focus of this test case. The pre-trained ResNext model must be loaded, the model's architecture and parameters must be defined, the model must be trained on both a small and a large dataset, and its performance must then be assessed using test data. A successful initialization with high accuracy is the desired result. The actual output, which achieved an accuracy of 88.06% on the test data, validates that the model was successfully trained with high accuracy. This test case was successful, proving that the model had been initialized correctly.

- Selecting the model's ideal hyperparameters is the goal of this test case. It entails a multi-step procedure whereby, following each training iteration, the hyperparameters are changed, the accuracy and loss for each value are noted, and values that minimize loss and maximize accuracy are chosen. The selection of appropriate hyperparameter values that produce low loss and high accuracy is the anticipated result. The real result validates that the right hyperparameter values were selected, resulting in a high-accuracy and successful model training process. The test case passed despite the difficult hyperparameter tuning process, indicating the efficacy of the selected hyperparameters.
- The purpose of this test case is to assess the trained model's performance. In order to understand the performance of the model, it entails choosing the proper evaluation metric for the machine learning project, experimenting with and settling on the evaluation metric, and examining the confusion matrix and prediction visualization. Accurate evaluation metrics that demonstrate the model's capacity to discriminate between genuine and deepfake images are anticipated, in addition to perceptive analysis findings. The analysis yielded insightful information, and the actual output attests to the accuracy with which the evaluation metrics were obtained, reflecting the model's performance. The fact that this test case was successful indicates that the trained model is reliable in practical situations.

Together, these test cases serve as essential benchmarks for validating the reliability and effectiveness of the ResNext101 and LSTM model in real-world deepfake detection applications, ensuring its readiness for deployment and usage.

3.2.8 Sprint Retrospective

Table 3.19 gives the Sprint Retrospective of Sprint 2 where we perform Model Creation and Training for the project.

Table 3.19 Sprint Retrospective of Sprint 2

Liked	Learned	Lacked	Longed For
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
Comprehensive exploration of various deep learning architecture for model selection. Experimenting with different model provide valuable insights about their strength and weakness	Research in deep learning architectures revealed the importance of selecting suitable models that balance computational complexity and performance.	Comprehensive documentation and resources for deep learning architectures posed a challenge, requiring additional time and effort to gather relevant information and insights	The team longed for more computational power in order to experiment more complex models which may have helped us create a better model
The ResNext101 and LSTM model that was chosen to create the deepfake detection model proved to effective in capturing both temporal and spatial features, essential for distinguishing real and fake videos	We gained insights into the intricacies of deep learning model development, including the importance of fine-tuning hyperparameters and optimizing the architecture for improved performance.	Comprehensive documentation of the model development and training process. While we successfully trained the deepfake detection model, the lack of detailed documentation hindered knowledge sharing and future reproducibility of results	The team longed for more computational power in order to increase the complexity of the model

During the Sprint 2 retrospective, several noteworthy aspects emerged that contributed to the overall progress and learning experience of the team. It was especially successful to thoroughly examine different deep learning architectures in order to choose a model; this gave important information about the advantages and disadvantages of each architecture. This procedure emphasized how crucial it is to choose models that are appropriate and strike a balance between computational complexity and performance, as deep learning architecture research has shown. The absence of thorough documentation and resources, however, presented a difficulty and required more time and work to compile pertinent data.. Nevertheless, the sprint taught us important lessons about developing deep learning models, such as how crucial it is to optimize architecture and fine-tune hyperparameters for better performance.

3.3 SPRINT 3: PREDICTION AND FRONT-END DEVELOPMENT

Sprint 3 heralds the phase of Prediction and Front-End Development, marking a crucial step towards the completion of our deepfake detection project. Creating an intuitive front end for user interaction, integrating prediction functionality seamlessly with the front end, testing the system's end-to-end functionality thoroughly, and developing the prediction pipeline to integrate the trained ResNext101 and LSTM model are the main goals of this sprint

3.3.1 Capacity Plan for Sprint 3

Table 3.20 gives the Capacity Planning of Sprint 3 where we perform Prediction and Front-End Development for the project.

Table 3.20 Capacity Planning of Sprint 3

		Capacity Plan for Sprint 3					
Name	**Role**	**Working Days**	**Planned Leaves (in Days)**	**Other Course work activities**	**Upskilling (in Days)**	**Design, Development, Testing, Documentation (in Days)**	**Estimated Hours**
Shresth Gupta	Data Scientist	20	10	3	2	15	45
Utkarsh Srivastava	UI/UX Designer	20	10	3	2	15	45
Shresth Gupta	Developer	20	10	3	3	18	54
Utkarsh Srivastava	QA Engineer	20	10	3	2	15	45

During Sprint 3, a thorough capacity plan was developed to maximize team output. After accounting for planned leaves (2–3 days) and external coursework (2–3 days), this plan identifies team members (Shresth Gupta, Utkarsh Srivastava) and their roles (Data Scientist, UI/UX Designer, Developer, QA Engineer), as well as their available workdays (20). Upskilling activities are given priority in the plan, with 15–18 hours per team member set aside for continuous learning. Both of the team members are estimated to work for around 45 to 54 hours. Overall, the capacity plan aims to optimize team productivity while accommodating individual roles and responsibilities effectively.

3.3.2 Detailed Estimation of User Stories

Table 3.21 gives the detailed estimation of user stories of Sprint 3 where we perform Prediction and Front-End Development for the project.

Table 3.21 Detailed Estimation of User Stories of Sprint 3

ID		Title	Epic	User Story	Priority (MoSCoW)	Status	Acceptance Criteria	Functional Requirements	Non-Functional Requirements	Original Estimate	Actual Effort (In days)
1	**Sprint 3**	Model Prediction	Prediction and Front-End Development	As a user, I want to use the trained deep learning model to predict whether a video is a deepfake or authentic.	Must	Ready	Load the trained model, implement a prediction pipeline using the model and display the results	1. Load the trained deep learning model and its weights. 2. Develop a prediction pipeline to preprocess input videos. 3. Predict authenticity of uploaded videos.	1. Ensure fast and efficient model inference for real-time predictions. 2. Aim for high accuracy in predicting deepfake videos to minimize false positives and negatives.	5 days	5 days
2		Integration with Front-End	Prediction and Front-End Development	As a user, I want to interact with the deepfake detection system through a user-friendly front-end interface.	Must	Ready	Integrate the prediction pipeline with front-end interface, Ensure communication between front-end and backend.	1. Develop front-end components for video upload and prediction result display. 2. Integrate prediction pipeline with the front-end interface.	1. Design an intuitive and visually appealing interface for user interaction. 2. Ensure responsive front-end interactions with minimal latency.	5 days	5 days

In Sprint 3, the team's responsibilities are broken down into specific user stories and were centered around prediction and front-end development for the deepfake detection system.

Creating a prediction pipeline that integrates the trained model is the first user narrative. In order to implement the trained deepfake detection model in a production setting, this task is essential. The infrastructure will be set up by the data scientists, who will also make sure the model can accurately receive input data, make predictions, and produce outputs. The efficacy of the system as a whole depends on this pipeline, which forms the core of the prediction functionality.

The user story for building the front end for user interaction comes next. The front-end developer has been given this assignment; they will be in charge of creating the deepfake detection system's user interface (UI). In order to enable user interaction with the system, screens, buttons, forms, and other UI elements must be created. In order to make it simple for customers to upload films for analysis and view the results, the front-end developer will concentrate on creating a clear, user-friendly interface.

Subsequently, the user story entails the front end and prediction functionality integration. To guarantee that the prediction pipeline created previously is smoothly incorporated into the user interface, cooperation between the front-end developer and data scientists is necessary for this task. Creating a seamless experience for users to upload videos, initiate the prediction of the deepfake detection model, and obtain the findings in an intuitive way is the aim.

Testing the system's end-to-end functionality is the fourth user story. In order to make sure that the front-end interface, prediction pipeline, and their integration all function as planned, this process entails extensive testing. Functional testing will verify the features of the system, while usability testing will evaluate the user experience. In order to find and address any defects or issues prior to the system being deployed, this stage is essential.

3.3.3 Daily Scrum Activities

Table 3.22 gives the Daily Scrum Report for week 1 of Sprint 3 where we perform Prediction and Front-End Development for the project. Table 3.23 gives the Daily Scrum Report for week 2 of Sprint 3 where we perform Prediction and Front-End Development for the project

Table 3.22 Daily Scrum Report for Week 1 of Sprint 3

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Saved the trained model into a checkpoint file	Developing a prediction pipeline by taking an input video	Converting the input video into a face cropped video of the input	Giving the preprocessed input to the trained model and receiving an output	Integrating the complete prediction pipeline
	What are doing today?	Developing a prediction pipeline by taking an input video	Converting the input video into a face cropped video of the input	Giving the preprocessed input to the trained model and receiving an output	Integrating the complete prediction pipeline	Integrating the complete prediction pipeline
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Saved the trained model into a checkpoint file	Developing a prediction pipeline by taking an input video	Converting the input video into a face cropped video of the input	Giving the preprocessed input to the trained model and receiving an output	Integrating the complete prediction pipeline
	What are doing today?	Developing a prediction pipeline by taking an input video	Converting the input video into a face cropped video of the input	Giving the preprocessed input to the trained model and receiving an output	Integrating the complete prediction pipeline	Integrating the complete prediction pipeline
	Is there anything blocking you?	NO	NO	NO	NO	NO

Table 3.23 Daily Scrum Report for Week 2 of Sprint 3

Team member	Question	Monday	Tuesday	Wednesday	Thursday	Friday
Shresth Gupta	**What did you do yesterday?**	Integrating the complete prediction pipeline	Developing the front end for user interaction	Developing the front end for user interaction	Integrating the prediction functionality with front end	Conducting end-to-end functionality testing
	What are doing today?	Developing the front end for user interaction	Developing the front end for user interaction	Integrating the prediction functionality with front end	Conducting end-to-end functionality testing	Conducting end-to-end functionality testing
	Is there anything blocking you?	NO	NO	NO	NO	NO
Utkarsh Srivastava	**What did you do yesterday?**	Integrating the complete prediction pipeline	Developing the front end for user interaction	Developing the front end for user interaction	Integrating the prediction functionality with front end	Conducting end-to-end functionality testing
	What are doing today?	Developing the front end for user interaction	Developing the front end for user interaction	Integrating the prediction functionality with front end	Conducting end-to-end functionality testing	Conducting end-to-end functionality testing
	Is there anything blocking you?	NO	NO	NO	NO	NO

The Daily Scrum report for week 1 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's development stages. Both team members spent the first week of Sprint 3 working on creating and integrating the project's prediction pipeline. The trained model was first saved into a checkpoint file, and after that, they concentrated on creating the prediction pipeline. This entails taking an input video, face-cropping it, preprocessing it, and then feeding it through the trained model to produce an output. There haven't been any reported roadblocks this week, suggesting that the development and integration tasks are moving along without any issues. The team is getting closer to accomplishing the project's goals and providing a workable solution by continuously improving and integrating the prediction pipeline.

The Daily Scrum report for week 2 provides a comprehensive overview of the team's daily progress and the challenges encountered during the project's development stages. Both team members have been actively involved in developing the front end for user interaction and integrating the entire prediction pipeline during the last week of Sprint 3. This has involved putting together user interfaces and integrating front-end prediction features, among other things. To guarantee the system runs smoothly, they have also carried out extensive end-to-end functionality testing. There haven't been any reported roadblocks this week, which suggests that the team is working well together and making progress. The team is getting closer to meeting the project objectives and providing the end users with a reliable solution by concentrating on these crucial tasks.

3.3.4 Functional Documents

a. Introduction

Deepfake Detection using ResNext101 and LSTM is a software solution aimed at identifying manipulated or synthetic media content, commonly known as deepfakes. Leveraging advanced machine learning techniques, specifically ResNext101 convolutional neural networks (CNNs) and Long Short-Term Memory (LSTM) networks, this product offers robust detection capabilities to combat the proliferation of fake media in various contexts.

b. Product Goal

The product goal is to develop a robust and accurate deepfake detection system using deep learning techniques, specifically leveraging the ResNext architecture. The system aims to assist various stakeholders, including social media platforms, news agencies, and law enforcement agencies, in identifying and combating the proliferation of deepfake videos. By detecting deepfake content early, the system contributes to preserving the integrity of digital media and preventing misinformation.

c. Demography (Users, Location)

The target users of the product include social media platform moderators, content verification teams, news editors, journalists, and law enforcement agencies involved in combating digital manipulation and misinformation. The product will be applicable globally, catering to regions where the spread of deepfake content poses significant threats to societal trust and stability.

d. Business Processes

The business processes involved in Sprint 3 focusses on creating a prediction pipeline for a given input video and development of a front-end application for the prediction process. The prediction pipeline will take an input video and give the output whether the video is real or deepfake. The front-end application provides a user interface for the users to do the same and get the output along with the confidence of the model.

e. Features

- Feature 1: Video Prediction Pipelines
  - Description: The Video Prediction Pipeline feature take an input video, performs preprocessing on the video and gives the preprocessed video to the trained deepfake detection model as an input and obtain the result whether the video is real or deepfake
  - User Story: In User Story, the user wants the model to classify correctly whether the given input video for prediction is real or deepfake.

- Feature 2: User Interface
  - Description: The User Interface offers the user with a web application for giving the input video and to get the output of the same video whether it is real or deepfake.
  - User Story: In User Story, the user wants a web application that can be used to check for real or fake videos in optimized time.

f. Authorization Matrix

Table 3.24 gives the authorization matrix for Sprint 3 in Prediction and Front-End Development

Table 3.24 Authorization Matrix for Sprint 1

Role	Permission
Administrator	Administrators have full access to all system functionalities.
User	Users can use the deepfake detection system
Analyst	Analysts have access to both detection features and advanced analytics tools.

g. Assumptions

- The availability of a diverse and well-labelled dataset for training the ResNext model on deepfake detection.
- Sufficient computational resources for Data Preprocessing

3.3.5 UI Design

A crucial component of Sprint 3's Prediction and Front-End Development for the deepfake detection system is UI Design. The UI Design challenge for this sprint include developing a simple and easy-to-use interface that allows users to upload videos or images for deepfake detection and to view the predictions made.

This sprint's main objective for UI Design is to provide a fluid and approachable user interface that makes it simple for users to interact with the deepfake detection system. The Front-End Developer and the UI Design team work together to create the user interface for the two primary functions of uploading movies and showing prediction results. To improve the user experience, the design needs to be visually appealing, responsive, and intuitive.

The UI Design team develops a user interface element for the "upload" functionality that enables users to quickly choose and upload photos or videos for deepfake detection. This entails creating a clear, straightforward file upload system that works with a variety of frequently used video file formats. The UI Design team concentrates on the "display prediction results" functionality when the user submits the video. This entails creating an area of the user interface where users may see the results of the trained model's deepfake detection. The results, which indicate whether the uploaded media is categorized as legitimate or a deepfake, should be presented in an easily comprehensible style by the design.

3.3.6 Product Framework

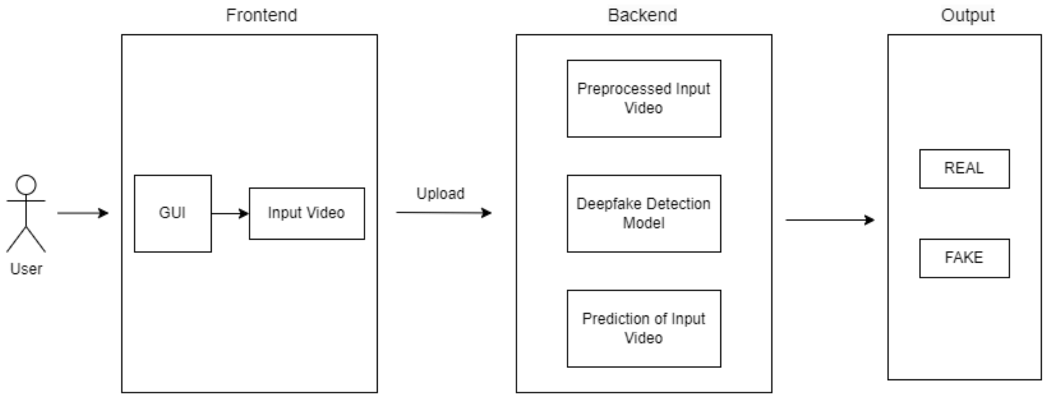


Fig 3.6 Product Framework for Web Application

Fig 3.6 shows the product framework for our web application for a Deepfake Detection Application.

The application is divided into three main components: Frontend, Backend, and Output.

- Frontend: Represents the user interface where the user interacts with a graphical user interface (GUI) to upload an input video.
- Backend: The input video is then uploaded to the backend where it undergoes preprocessing. After preprocessing, the video is passed through a deepfake detection model, which predicts whether the input video is real or fake.
- Output: The final prediction is then presented to the user, with the output clearly labeled as either 'REAL' or 'FAKE'.

3.3.7 Functional Test Case Document

Table 3.25 gives the functional test case document of Sprint 3 where we perform Prediction and Front-End Development for the project.

Table 3.25 Functional Test Case Table of Sprint 3

Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status
Prediction Pipeline	Creating a prediction pipeline	Take an input video from user, perform preprocessing on it and then predict the type of video using the model along with the confidence of the model	Model correctly predicts the given input video and also displays the confidence	Model is able to predict the output and displays the confidence of the model as well	Passed
Web application interface	User Interface Test Case	Launch web application interface, navigate to deepfake detection page, input alleged deepfake image, View predicted video result	Application interface loads successfully Page loads without errors Video upload functionality works as expected	Deepfake detection displayed accurately Features accessible without errors	Passed

The Functional Test Case Report for Sprint 3 outlines two key test cases: Creating a prediction pipeline and user interface.

- The purpose of the Prediction Pipeline test case was to verify the pipeline's ability to predict the authenticity and degree of confidence in a given video. Preprocessing, running the video through the model, and displaying the prediction and confidence were the test steps. The model successfully predicted the authenticity of the video and showed the expected confidence levels, indicating that the test was successful.
- The goal of the User Interface Test Case was to make sure the web application interface operated without a hitch. The procedure involved opening the interface, going to the page for deepfake detection, entering a purportedly deepfake image, and seeing the anticipated outcomes. The test verified that the video upload feature worked properly, the interface loaded successfully, and the results of the deepfake detection were accurately displayed with confidence scores.

Overall, the tests demonstrated the robustness and reliability of both the prediction pipeline and the web application interface, ensuring a high-quality user experience.

3.3.8 Sprint Retrospective

Table 3.26 gives the Sprint Retrospective of Sprint 3 where we perform Prediction and Front-End Development for the project.

Table 3.26 Sprint Retrospective of Sprint 3

Liked	Learned	Lacked	Longed For
Share aspects of the sprint that you enjoyed or found particularly effective.	*Discuss lessons learned, whether they are related to processes, technical aspects, or teamwork.*	*Identify areas where the team felt a lack of resources, support, or information.*	*Discuss any desires or expectations that the team had but were not met during the sprint.*
During the prediction and front-end development sprint, the team found the integration of the trained model with the prediction pipeline to be smooth and effective.	Lessons were learned regarding the importance of error handling in the prediction pipeline.	We felt a lack of access to a wider range of video formats for testing would have improved the robustness of the prediction pipeline.	We had hoped for a smoother integration process with the front-end components and more extensive user testing and feedback.
Developing a user-friendly front-end interface received positive feedback during testing, and effective communication between the front-end and backend teams was vital.	Lessons were learned on how to integrate the deep learning model into the front-end for the user to take advantage of the application	We felt a lack of support for optimizing front-end performance. Expectations for a smoother integration process with the front-end components and more extensive user testing and feedback were not entirely met.	Limited resources impacted our ability to optimize front-end performance and access a wider range of video formats for testing.

During the Sprint 3 retrospective, the team reflected on several positive aspects and valuable lessons learned throughout the development process. The seamless and successful integration of the trained model with the prediction pipeline, which produced an intuitive front-end interface, was one noteworthy highlight. Positive feedback was obtained during testing for this interface, demonstrating how well it facilitates user interaction. The retrospective also underscored how important it was for the front-end and back-end teams to communicate effectively with one another, as this was essential to facilitating collaboration and seamless integration. But there were also chances for growth and learning during the sprint. During the testing phases, it became clear how important it was to have robust error handling within the prediction pipeline. In addition, the team learned a lot about how to improve user experience and accessibility by incorporating deep learning models into the front end. Notwithstanding these achievements, difficulties persisted, such as what seemed to be a lack of funding and assistance for improving front-end performance. Furthermore, there were limitations on the prediction pipeline's robustness due to restricted availability of a wide variety of video formats for testing. Additionally, there were unmet expectations for more thorough user testing and feedback as well as a more seamless integration process with the front-end components. With an eye toward the future, the team is still dedicated to tackling these obstacles, using the knowledge gained to improve their strategy and raise user satisfaction and application performance.

CHAPTER 4

RESULTS AND DISCUSSION

Each sprint's results have aided in the creation of a strong deepfake detection system. With careful data gathering, preparation, and pipeline design, Sprint 1 set the foundation. Sprint 3 built a user-friendly interface for the model and pushed it into production, while Sprint 2 concentrated on its creation and optimization. These results represent a major advancement toward the project's objectives of improving web security and thwarting the spread of deepfake content.

4.1 OUTCOME OF SPRINT 1

Investigations of deepfake datasets and tools that are now available have yielded important insights regarding the state of deepfake data. The basis for training and testing the deepfake detection model was collected from multiple sources, comprising labeled authentic and deepfake videos. The gathered data underwent rigorous preprocessing to guarantee that it was in the best possible format for model training. In order to improve the dataset's diversity and support the deepfake detection model's resilience and generalizability, data enrichment techniques were devised. It was feasible to construct a pipeline for preprocessing and data loading, which streamlined the procedure for upcoming data manipulation jobs.

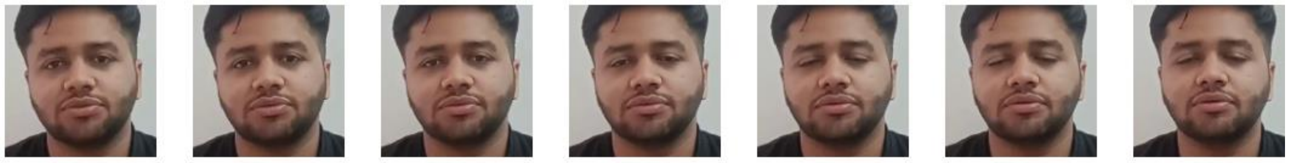


Fig 4.1 Preprocessed face cropped frames of video

Fig 4.1 shows the preprocessed face cropped frames of the input video. These face cropped frames are then converted into a video format for the ResNext101 model.

4.2 OUTCOME OF SPRINT 2

Deep learning architectures appropriate for deepfake detection have been thoroughly studied, with an emphasis on accuracy and efficiency. A number of models, such as ResNext101_64x4d, were tested in order to identify the best architecture for the job. The preprocessed data was used to train the chosen models, improving their capacity to identify deepfake content. In order to evaluate the efficacy of the trained models, measures like accuracy, precision, and recall were examined. In order to increase the model's robustness and accuracy and enable more accurate identification of deepfake movies, fine-tuning approaches were used.

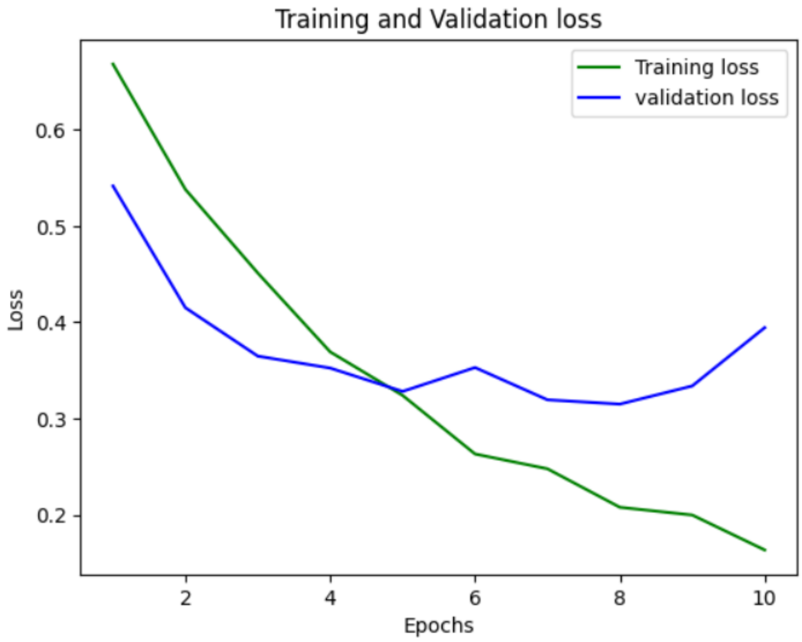


Fig 4.2 Training and Validation Loss

Fig 4.2 shows the training and validation loss of the ResNext101 model on our training and testing data. The training loss calculated for the model is 0.163536 and the validation loss is 0.394417.

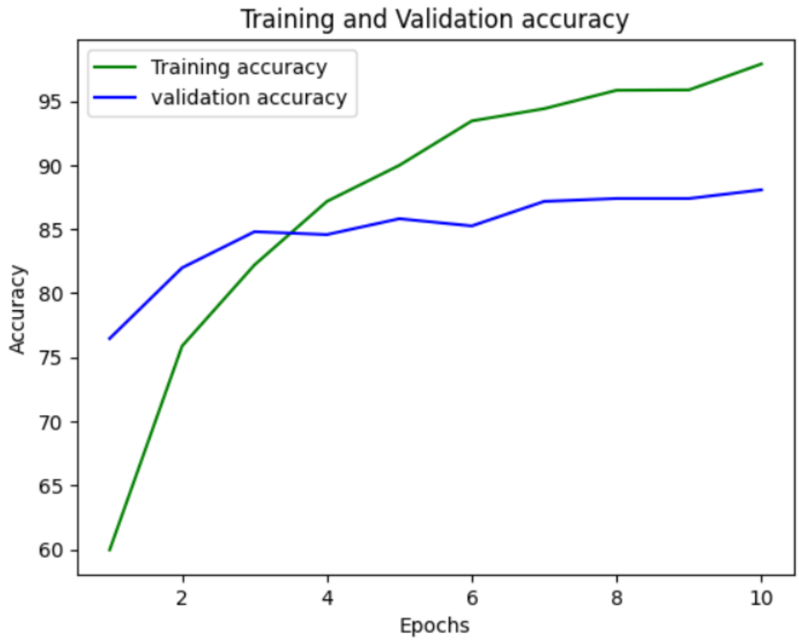


Fig 4.3 Training and Validation Accuracy

Fig 4.3 shows the training and validation accuracy of the ResNext101 model on our training and testing data. The training accuracy calculated for the model is 97.89% and the validation loss is 88.06%.

4.3 OUTCOME OF SPRINT 3

The result displayed for deepfake detection consists of two main components: the predicted class andthe corresponding confidence score.

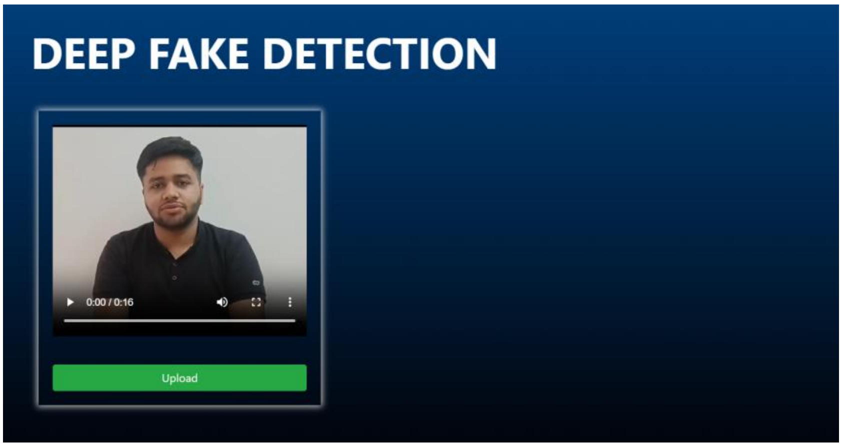
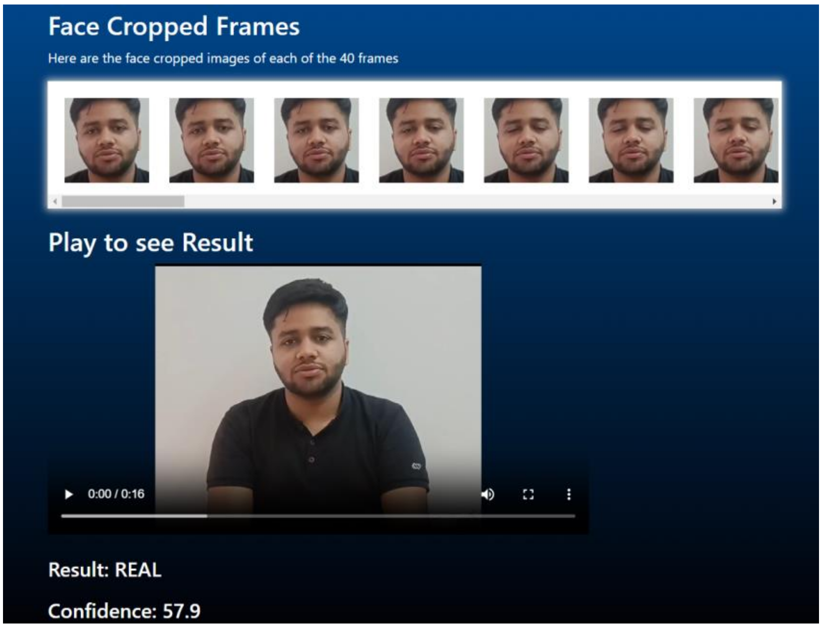

DEEP FAKE DETECTION

Fig 4.5 Input for Deepfake Detection of Real Video

Fig 4.5 shows the user interface where the user provides an input video for Deepfake Video Detection. After the videois successfully loaded, the user clicks on "Upload" button,to look for whether the video is real or fake.


Face Cropped Frames

Here are the face cropped images of each of the 40 frames

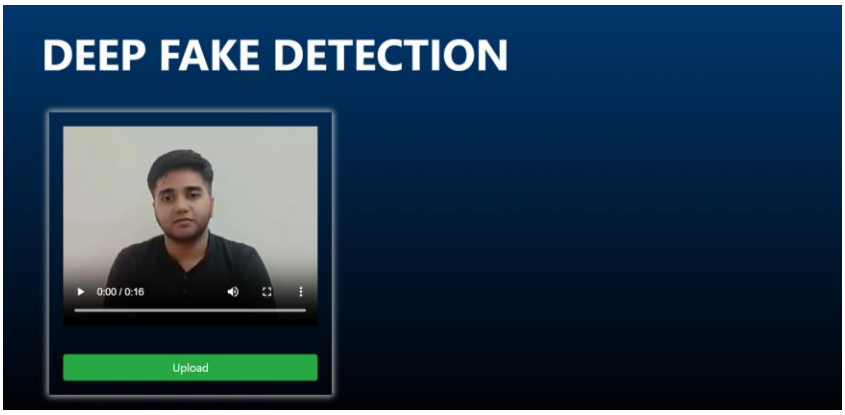
Play to see Result

Result: REAL

Confidence: 57.9

Fig 4.6 Output for Deepfake Detection of Real Video

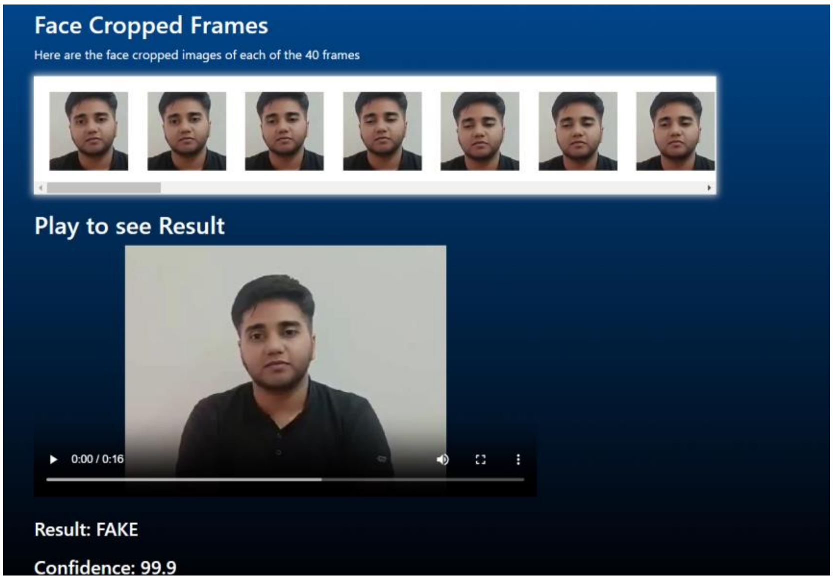
Fig 4.6 displays the desired output for the given video. It displays the face cropped images of each of the 40 frames extracted from the video along with the prediction whether it is “REAL” or “FAKE” and the confidence of the model.



DEEP FAKE DETECTION

Fig 4.7 Input for Deepfake Detection of Fake Video

Fig 4.7 shows the user interface where the user provides an input video for Deepfake Video Detection. After the videois successfully loaded, the user clicks on “Upload” button,to look for whether the video is real or fake.



Face Cropped Frames

Here are the face cropped images of each of the 40 frames

Play to see Result

Result: FAKE

Confidence: 99.9

Fig 4.8 Output for Deepfake Detection of Fake Video

Fig 4.8 displays the desired output for the given video. It displays the face cropped images of each of the 40 frames extracted from the video along with the prediction whether it is “REAL” or “FAKE” and the confidence of the model.

Following the trained deepfake detection model's processing of the input video frames, the class with the highest probability among the output probabilities is used topredict the class. The main indicator of whether the input video is categorized as a real or deepfake is this predicted class. The confidence score, in addition to the expectedclass, offers important information about how confident the model is in its prediction. Higher confidence scores indicategreater possibility that the predicted class is correct. The confidence score is a representation of this likelihood. Amodel may be 80% convinced that a video containsmanipulations if, for example, it forecasts the video to be a deepfake with a confidence score of 80%. In order to make well-informed decisions about the legitimacy of the input video content, users and analysts must be able to comprehendthe significance of this outcome in terms of the prediction accuracy of the deepfake detection system.

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the deepfake detection system that has been described is a big step forward in the process of tackling the issues that have been brought about by the fast-paced development of deepfake technology. It is possible to ensure the adaptability and resilience of the model in identifying a wide range of deepfake variants by using a diverse dataset that covers a variety of settings and contexts. Through its emphasis on facial area extraction, the preprocessing pipeline establishes a strong basis for the training of deepfake detection models that are both accurate and operationally efficient. A specific method for capturing both spatial and temporal variations within video data is reflected in the model architecture that was selected. This architecture includes the ResNeXt-101 backbone and the LSTM layer. The objective of the model is to acquire a full understanding of deepfake manipulations by utilizing neural network architectures and a wide variety of datasets. With the incorporation of a preprocessing methodology and a model architecture, this system is positioned to become an successful tool in the fight against the misuse of artificial intelligence based - generated altered media.

REFERENCES

[1] Li, Y., & Lyu, S. (2018). Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*.

[2] Agarwal, S., Hu, L., Ng, E., Darrell, T., Li, H., & Rohrbach, A. (2023). Watch those words: Video falsification detection using word-conditioned facial motion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 4710-4719).

[3]Nguyen, H. H., Yamagishi, J., & Echizen, I. (2019, May). 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)* (pp. 2307-2311). IEEE.

[4] 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*.

[5] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems, 27*.

[6]Trefný, J., & Matas, J. (2010, February). Extended set of local binary patterns for rapid object detection. In *Computer vision winter workshop* (pp. 1-7).

[7] Güera, D., & Delp, E. J. (2018, November). 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.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[9]R. Raghavendra, Kiran B. Raja, Sushma Venkatesh, and Christoph Busch, “Transferable deep-CNN features for detecting digital and print-scanned morphed face images,” in CVPRW. IEEE, 2017.

[10] Tiago de Freitas Pereira, Andre Anjos, Jose Mario DeMartino, and Sebastien Marcel, “Can face anti spoofing countermeasures work in a real world scenario?”, in ICB. IEEE, 2013.

[11] W. Quan, K. Wang, D.-M. Yan, and X. Zhang, “Distinguishing between natural and computer-generated images using convolutional neural networks,” Trans. Inform. Forensics Secur., vol. 13, no. 11, pp. 2772–2787, 2018.

[12] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. € Nießner, “Faceforensics: A large-scale video dataset for forgery detection in human faces,” 2018, arXiv:1803.09179.

[13] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, “Recurrent convolutional strategies for face manipulation detection in videos,” CVPRw, pp. 80–87, 2019.

[14] F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” in Proc. Winter Conf. Appl. Comput. Vis. Workshops, 2019, pp. 83–92.

[15] H. H. Nguyen, T. Tieu, H.-Q. Nguyen-Son , V. Nozick, J. Yamagishi, and I. Echizen, “Modular convolutional neural network for discriminating between computer-generated images and photographic images,” in Proc. Int. Conf. Availability, Rel. Secur., 2018, pp. 1–10.

[16] H. H. Nguyen, J. Yamagishi, and I. Echizen, “Use of a capsule network to detect fake images and videos,” 2019, arXiv:1910.12467.

[17] S.-Y. Wang, O. Wang, A. Owens, R. Zhang, and A. A. Efros, “Detecting photoshopped faces by scripting photoshop,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 10072–10081.

[18] X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2019, pp. 8261–8265.

[19] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, “Two-stream neural networks for tampered face detection,” in Proc. Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 1831–1839

APPENDIX A

PATENT DISCLOSURE FORM

Application for patent filing

Date	D	D	M	M	Y	Y	Y	Y
	3	0	0	4	2	0	2	4

Name of the Faculty	:	**Dr. Antony Sophia N**
Department	:	**Computational Intelligence**
Faculty ID Number	:	
Official E-mail ID	:	
Contact no. of all Inventors		**Shresth Gupta (6206048094)** **Utkarsh Srivastava (8545831533)** **Dr. Antony Sophia N (9894119157)**
Major area of invention	:	**The developed product focuses on improving the response against cybercrimes by providing easy access to a powerful tool segregating real or fake multimedia content.**
Narrow focus area of invention	:	**The narrow focus of the project is to provide the functionality of this application on various social media platforms for strengthening digital security**
Title of the invention	:	**Deepfake Detection using ResNext101 and LSTM**
Earlier status of research	:	**Research has already been done for this**
How different your invention from similar research / others - **Novelty**?	:	**The novelty of this invention is to reduce the computational power and make the application more scalable and compatible on low-end devices**
Possible domain for field application	:	**Web Application**
Possible sector for commercialization	:	**The product will be present in the internet for people to detect a particular video is real or fake**
Faculty Signature with date	:	
HoD Remarks / Recommendation for filing patent	:	

Application received by The Review committee on	:	
Review committee remarks	:	

<u>Invention Disclosure Form</u>
To be filled by the inventors

Please provide highly relevant information for details asked below and use consistent language while describing the specific feature or element in the invention disclosure.

1. **Title of invention** (Please indicate a title for the invention and technology of the invention)

   The title of invention is "**Deepfake Detection using ResNext101 and LSTM**".

2. **Describe the invention**. (Please describe specifically about the general purpose of invention. Is the invention a new process, device of product, system, software or a combination of these elements?)

   The general purpose of this invention is to detect whether the video is real or fake. This invention is software that will be present on the internet,

3. Does the invention provide a **new use of or improvement to an existing product or process**? (Highlight the use or improvements from the existing with recent and relevant references)

   This project is computationally efficient as it could be run on low-end devices with not much computational power.

4. State the **Novelty** of the invention and specify the claims in the invention

   The novelty of the invention lies in the integration of ResNext101 and LSTM, offering improved accuracy and reliability.

5. Describe the **advantages of the present invention over the existing technologies** (please identity the advantages e.g. efficiency, cost benefits, simplicity etc.)

   The present invention provides the advantage of increased accuracy, reliability and scalability over existing technologies.

6. Describe how the **present invention overcomes the drawbacks** of currently available technology related to your invention. (please include the relevant references)

   The present invention analyses both spatial and temporal aspects of videos which was a drawback in other inventions.

7. Describe **uses, applications and benefits** of the invention.

   The primary application of the invention is in social media platform where the model can flag and remove deepfake videos thus preserving integrity of user-generated content

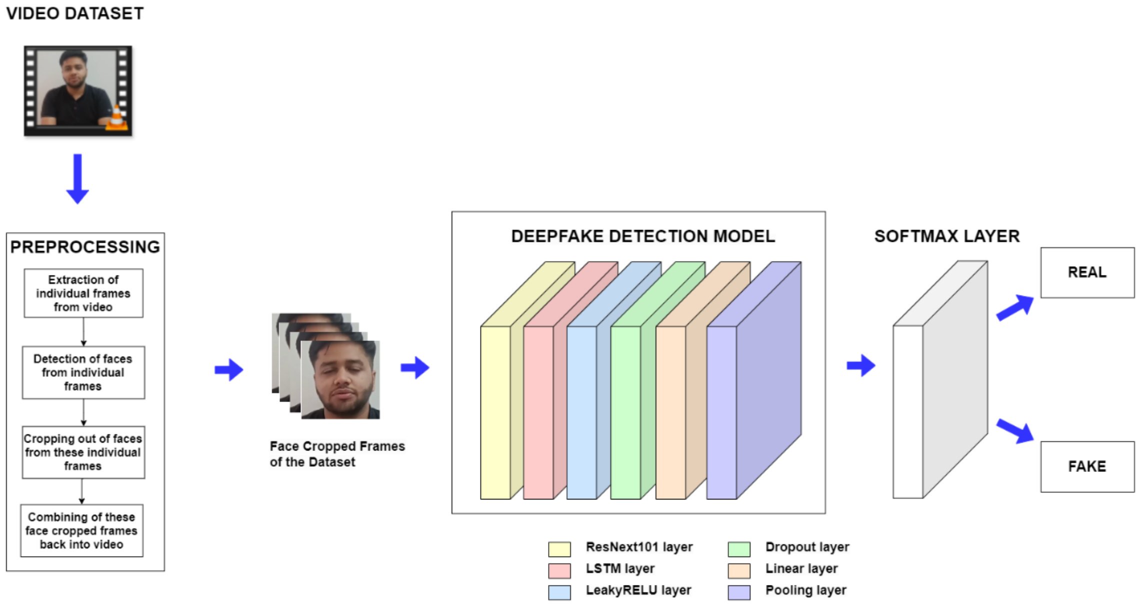
8. Does the focus of the invention results in **societal impact technology**? (Please describe how in detail, also specify the commercial applications, market need of product/ service of invention and why?)

   This invention contributes to combatting misinformation and flag manipulated content thus safeguarding societal trust and stability.

9. Characterize the **disadvantages and limitations** of the invention

   The major disadvantage of the invention is it still requires significant computational resources for training and inference. Also, the quality of dataset available is also one of the limitations.

10. Enclose the **sketches, drawings, photographs** and other materials that help in better understating/ illustration of the novelty in the invention.



VIDEO DATASET
PREPROCESSING
Extraction of individual frames from video
Detection of faces from individual frames
Cropping out of faces from these individual frames
Combining of these face cropped frames back into video
Face Cropped Frames of the Dataset
DEEPFAKE DETECTION MODEL
SOFTMAX LAYER
REAL
FAKE
ResNext101 layer
LSTM layer
LeakyRELU layer
Dropout layer
Linear layer
Pooling layer

Fig. Architecture Diagram of Proposed Model

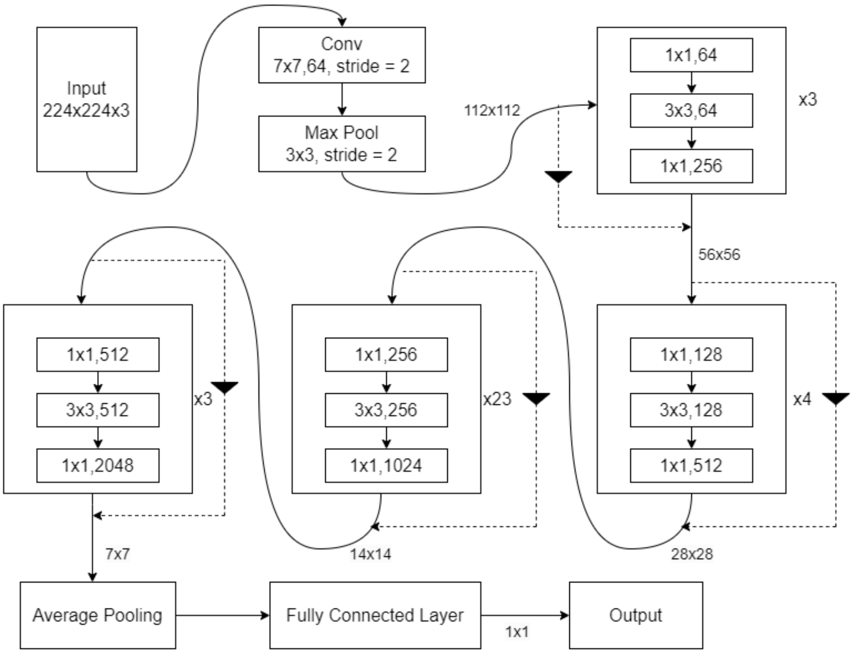


Fig. Architecture of ResNext101 Model

11 Current development status of the invention

A. Has your invention been tested experimentally

. Yes, the invention has been tested experimentally.

B. Describe the experimental approach of the invention also state the methods adopted in the experiment.

The experimental approach adopted for this experiment is:
Phase 1: Data Collection and Preprocessing
Phase 2: Model Creation and Training
Phase 3: Prediction and Front-End Development'=

C. Are the experimental data is documented in a formal log or any instrumental confirmation available for the invention (kindly provide the details)

The experimental data used for the invention are:

1. Deepfake Detection Challenge (DFDC): It is a dataset for deepfake detection consisting of more than 100,000 videos.
2. CelebDF: It is a large-scale dataset for deepfake forensics. It includes 590 original videos with subjects of different ages, ethnic groups and genders, and 5639 corresponding DeepFake videos.

D. Is further development of your invention is necessary or development of the invention is in progress (provide the relevant information)

The development of the invention is still in progress.

12. **Please list any of your publications** (including abstracts, posters, news releases, etc.) to emphasize the present invention background.

    Li, Y., & Lyu, S. (2018). Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*.

13. **INVENTOR(S) AND/OR CONTRIBUTOR(S):**

	INVENTOR (1)	INVENTOR (2)
Signature:		
Name:	Shresth Gupta	Utkarsh Srivastava
Address:	Department, SRM IST, Kattankulathur campus-603203	Department, SRM IST, Kattankulathur campus-603203
City and State:	Chennai &Tamilnadu	Chennai & Tamilnadu
Citizenship (Country):	INDIAN	INDIAN

	INVENTOR (3)
Signature:	
Name:	Dr. Antony Sophia N
Address:	Department, SRM IST, Kattankulathur campus-603203
City and State:	Chennai &Tamilnadu
Citizenship (Country):	INDIAN

14. ASSIGNMENT DETAILS: Assignee is the entity or individual who holds the patent.

Signature: (To be signed by the authorized signatory on behalf of the assignee)

Name of the Authorized Signatory and Designation

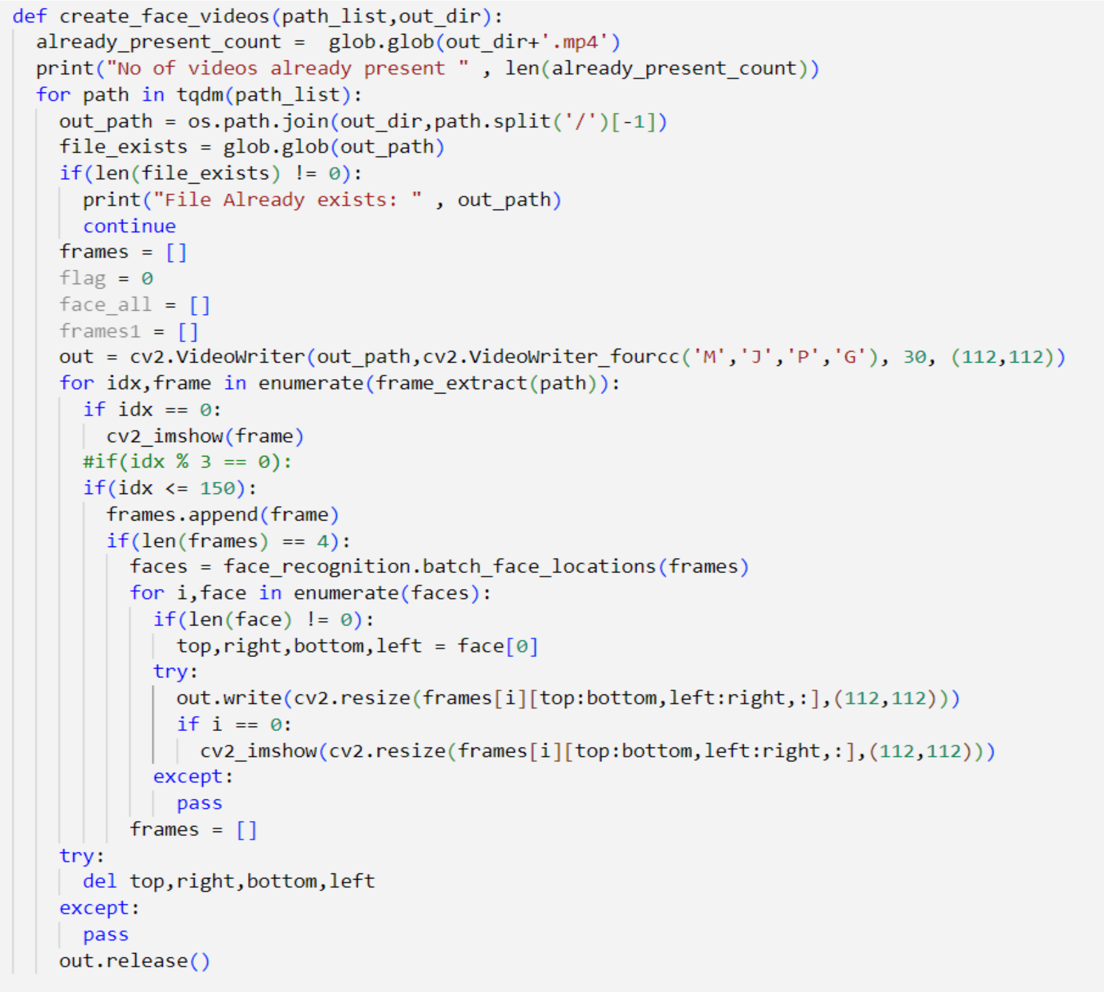
Address:

City and State:

Citizenship (Country):

APPENDIX B

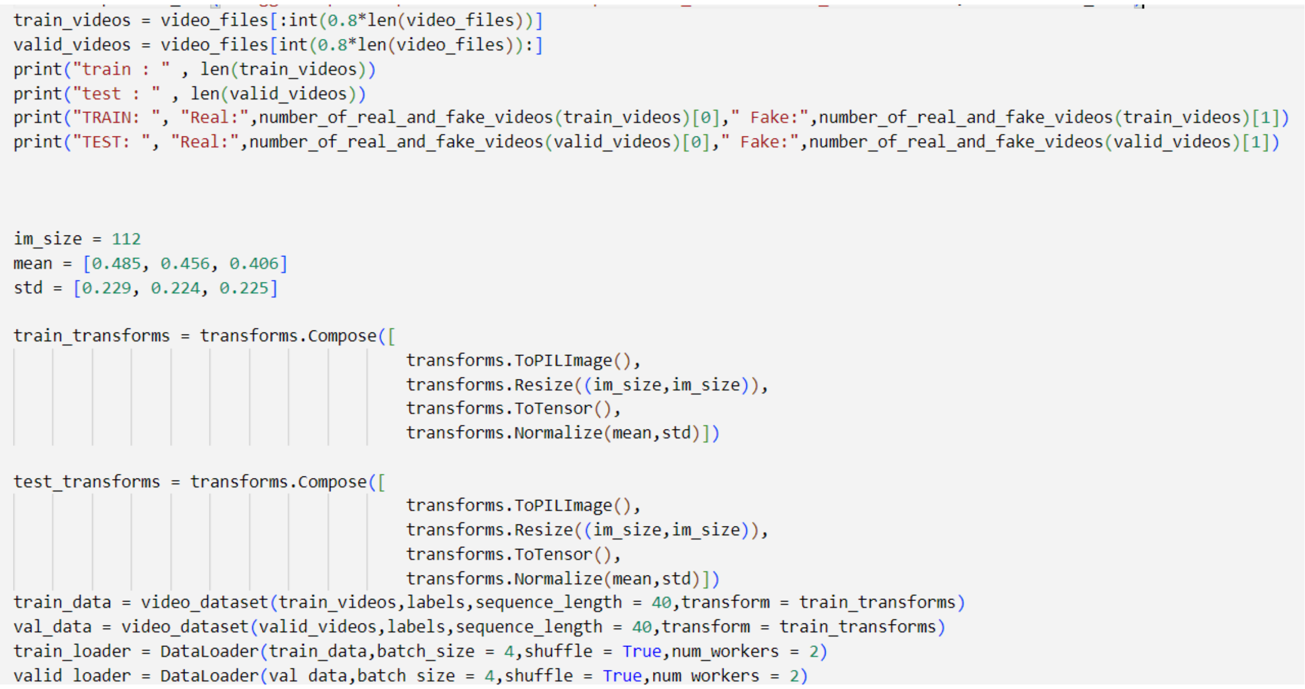
SAMPLE CODING



```
def create_face_videos(path_list,out_dir):
    already_present_count =  glob.glob(out_dir+'.mp4')
    print("No of videos already present " , len(already_present_count))
    for path in tqdm(path_list):
        out_path = os.path.join(out_dir,path.split('/')[-1])
        file_exists = glob.glob(out_path)
        if(len(file_exists) != 0):
            print("File Already exists: " , out_path)
            continue
        frames = []
        flag = 0
        face_all = []
        frames1 = []
        out = cv2.VideoWriter(out_path,cv2.VideoWriter_fourcc('M','J','P','G'), 30, (112,112))
        for idx,frame in enumerate(frame_extract(path)):
            if idx == 0:
                cv2_imshow(frame)
            #if(idx % 3 == 0):
            if(idx <= 150):
                frames.append(frame)
                if(len(frames) == 4):
                    faces = face_recognition.batch_face_locations(frames)
                    for i,face in enumerate(faces):
                        if(len(face) != 0):
                            top,right,bottom,left = face[0]
                        try:
                            out.write(cv2.resize(frames[i][top:bottom,left:right,:],(112,112)))
                            if i == 0:
                                cv2_imshow(cv2.resize(frames[i][top:bottom,left:right,:],(112,112)))
                        except:
                            pass
                    frames = []
        try:
            del top,right,bottom,left
        except:
            pass
        out.release()
```

Fig B.1 Preprocessing of the videos

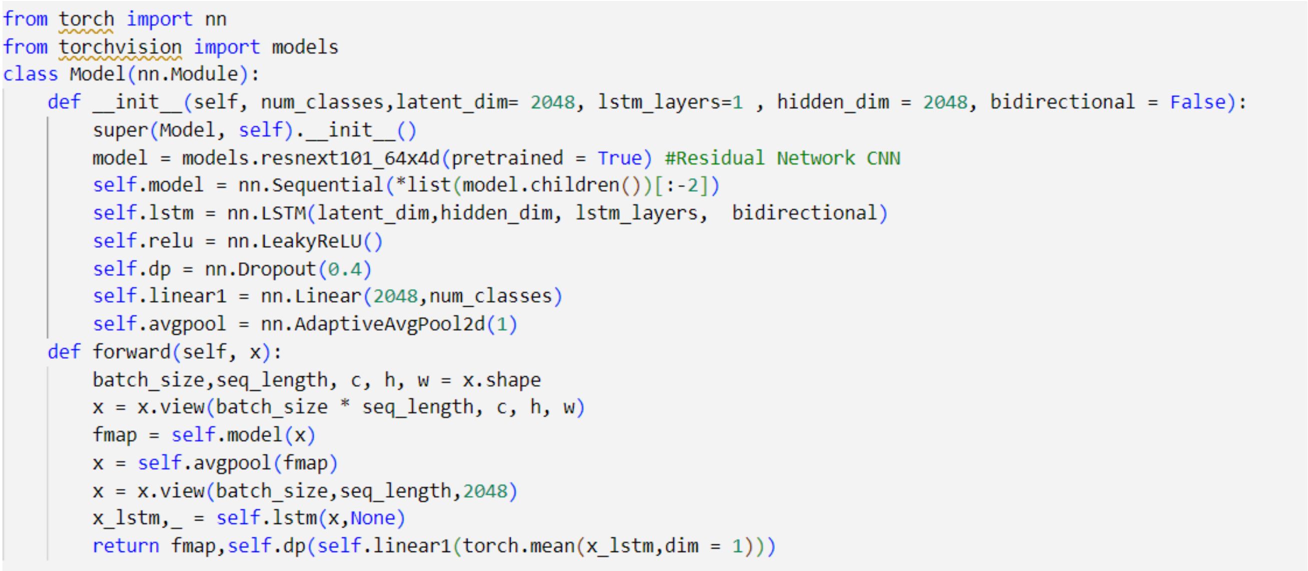
Fig B.1 shows the function where the preprocessing of the input video which includes the extraction of the frames from the videos, capturing the face cropped frames from the video and then converting back into a video so that the preprocessed input for the given input is a face cropped of the same video.



```
train_videos = video_files[:int(0.8*len(video_files))]
valid_videos = video_files[int(0.8*len(video_files)):]
print("train : " , len(train_videos))
print("test : " , len(valid_videos))
print("TRAIN: ", "Real:",number_of_real_and_fake_videos(train_videos)[0]," Fake:",number_of_real_and_fake_videos(train_videos)[1])
print("TEST: ", "Real:",number_of_real_and_fake_videos(valid_videos)[0]," Fake:",number_of_real_and_fake_videos(valid_videos)[1])


im_size = 112
mean = [0.485, 0.456, 0.406]
std = [0.229, 0.224, 0.225]

train_transforms = transforms.Compose([
                                        transforms.ToPILImage(),
                                        transforms.Resize((im_size,im_size)),
                                        transforms.ToTensor(),
                                        transforms.Normalize(mean,std)])

test_transforms = transforms.Compose([
                                        transforms.ToPILImage(),
                                        transforms.Resize((im_size,im_size)),
                                        transforms.ToTensor(),
                                        transforms.Normalize(mean,std)])
train_data = video_dataset(train_videos,labels,sequence_length = 40,transform = train_transforms)
val_data = video_dataset(valid_videos,labels,sequence_length = 40,transform = train_transforms)
train_loader = DataLoader(train_data,batch_size = 4,shuffle = True,num_workers = 2)
valid_loader = DataLoader(val_data,batch_size = 4,shuffle = True,num_workers = 2)
```

Fig B.2 Data Splitting and Data Loading

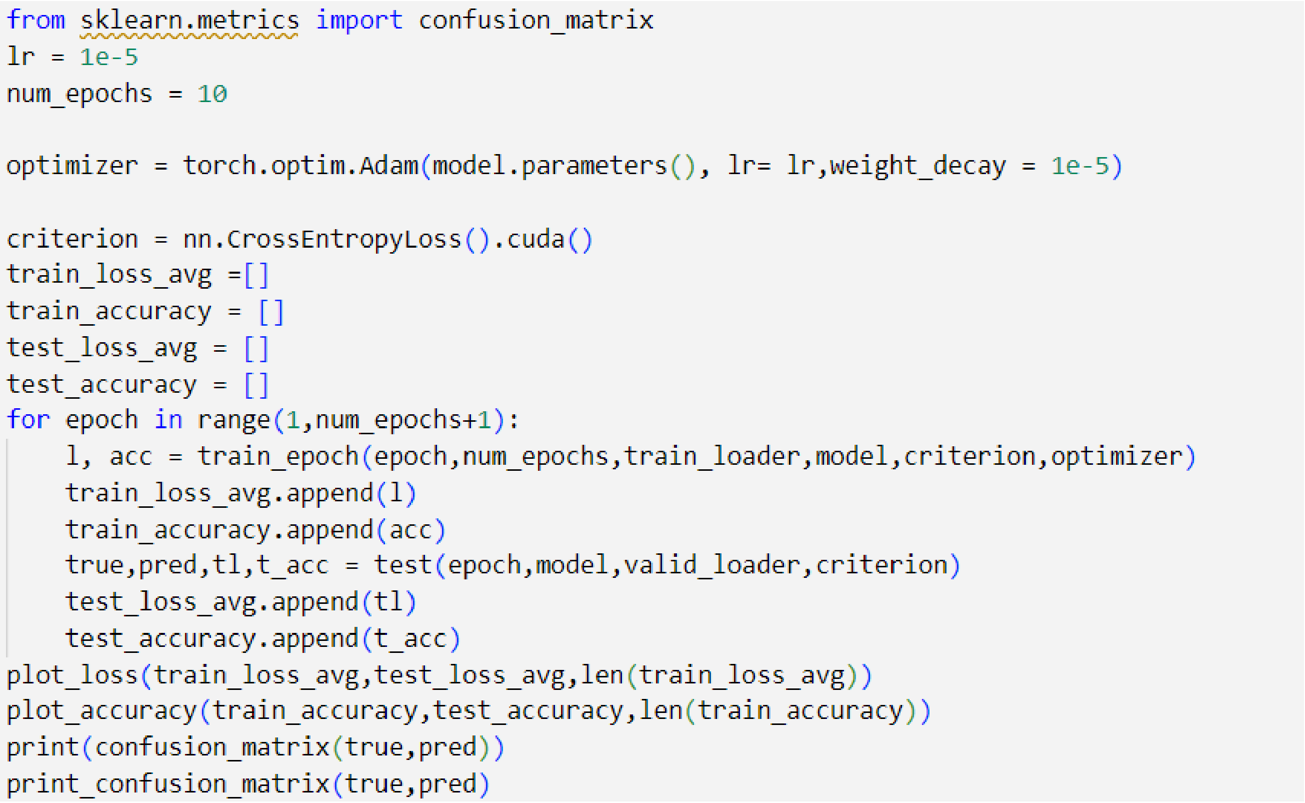
Fig B.2 shows process of data splitting and data loading which is used for training and testing the model. The video dataset is broken down into training data and testing data in the ratio of 4:1. The videos have sequence length of 40 and are loaded into the model with batch size of 4.



```
from torch import nn
from torchvision import models
class Model(nn.Module):
    def __init__(self, num_classes,latent_dim= 2048, lstm_layers=1 , hidden_dim = 2048, bidirectional = False):
        super(Model, self).__init__()
        model = models.resnext101_64x4d(pretrained = True) #Residual Network CNN
        self.model = nn.Sequential(*list(model.children())[:-2])
        self.lstm = nn.LSTM(latent_dim,hidden_dim, lstm_layers,  bidirectional)
        self.relu = nn.LeakyReLU()
        self.dp = nn.Dropout(0.4)
        self.linear1 = nn.Linear(2048,num_classes)
        self.avgpool = nn.AdaptiveAvgPool2d(1)
    def forward(self, x):
        batch_size,seq_length, c, h, w = x.shape
        x = x.view(batch_size * seq_length, c, h, w)
        fmap = self.model(x)
        x = self.avgpool(fmap)
        x = x.view(batch_size,seq_length,2048)
        x_lstm,_ = self.lstm(x,None)
        return fmap,self.dp(self.linear1(torch.mean(x_lstm,dim = 1)))
```

Fig B.3 ResNext101 and LSTM Model

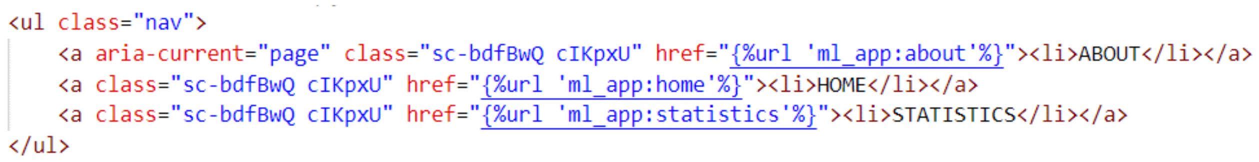
Fig B.3 shows the ResNext101 and LSTM model that is used for creating a deepfake detection model by training the data on the train video dataset. It has our ResNext101_64*4d model along with 1 layer of LSTM, LeakyRELU activation function and Dropout for regularization of the model.



```
from sklearn.metrics import confusion_matrix
lr = 1e-5
num_epochs = 10

optimizer = torch.optim.Adam(model.parameters(), lr= lr,weight_decay = 1e-5)

criterion = nn.CrossEntropyLoss().cuda()
train_loss_avg =[]
train_accuracy = []
test_loss_avg = []
test_accuracy = []
for epoch in range(1,num_epochs+1):
    l, acc = train_epoch(epoch,num_epochs,train_loader,model,criterion,optimizer)
    train_loss_avg.append(l)
    train_accuracy.append(acc)
    true,pred,tl,t_acc = test(epoch,model,valid_loader,criterion)
    test_loss_avg.append(tl)
    test_accuracy.append(t_acc)
plot_loss(train_loss_avg,test_loss_avg,len(train_loss_avg))
plot_accuracy(train_accuracy,test_accuracy,len(train_accuracy))
print(confusion_matrix(true,pred))
print_confusion_matrix(true,pred)
```

Fig B.4 Model Training

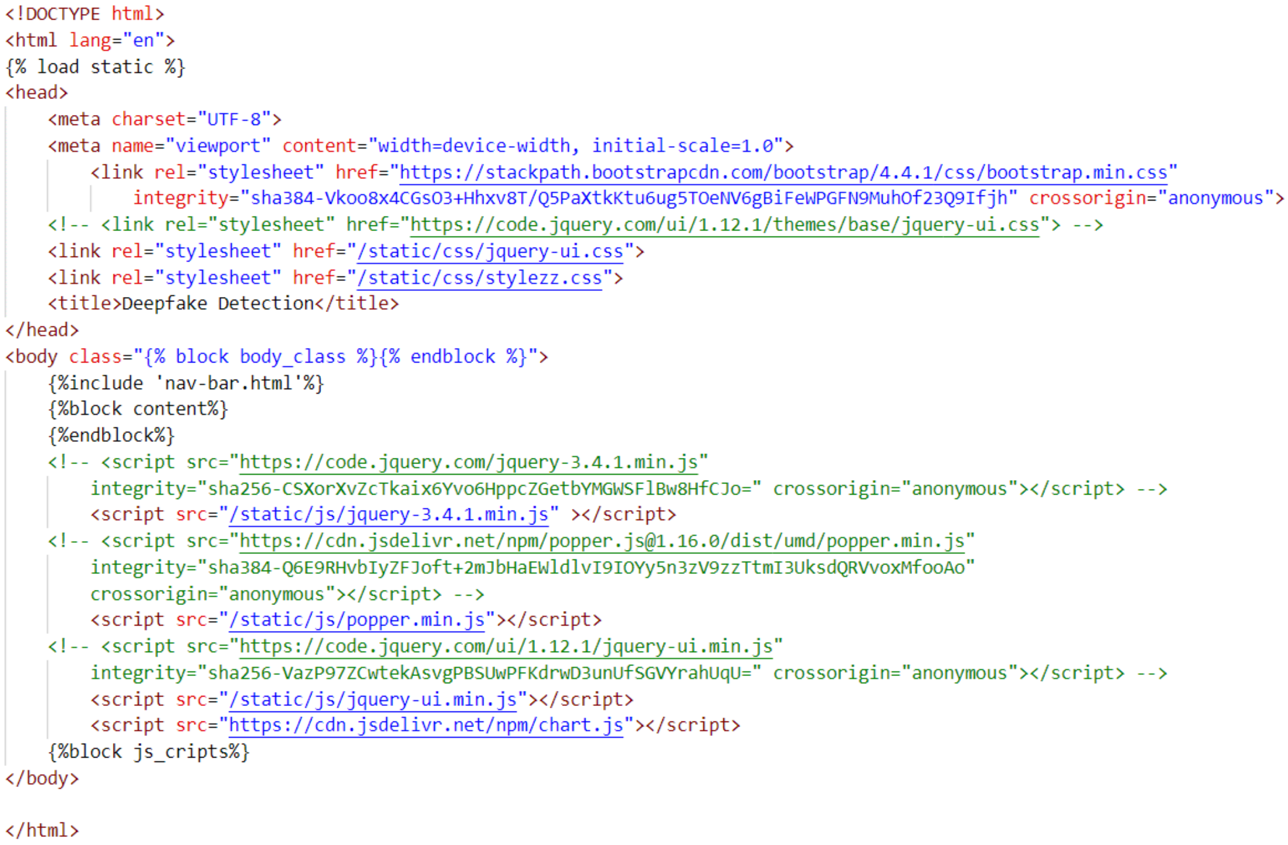
Fig B.4 shows the model training of the ResNext101 and LSTM. We have considered learning rate of 1e^{-5} and take number of epochs as 10. The training loss, training accuracy, testing loss and testing accuracy are stored in different lists to create two graphs , one for training and testing loss and another for training and testing accuracy.



```
<ul class="nav">
    <a aria-current="page" class="sc-bdfBwQ cIKpxU" href="{%url 'ml_app:about'%}"><li>ABOUT</li></a>
    <a class="sc-bdfBwQ cIKpxU" href="{%url 'ml_app:home'%}"><li>HOME</li></a>
    <a class="sc-bdfBwQ cIKpxU" href="{%url 'ml_app:statistics'%}"><li>STATISTICS</li></a>
</ul>
```

Fig B.5 nav-bar.html file

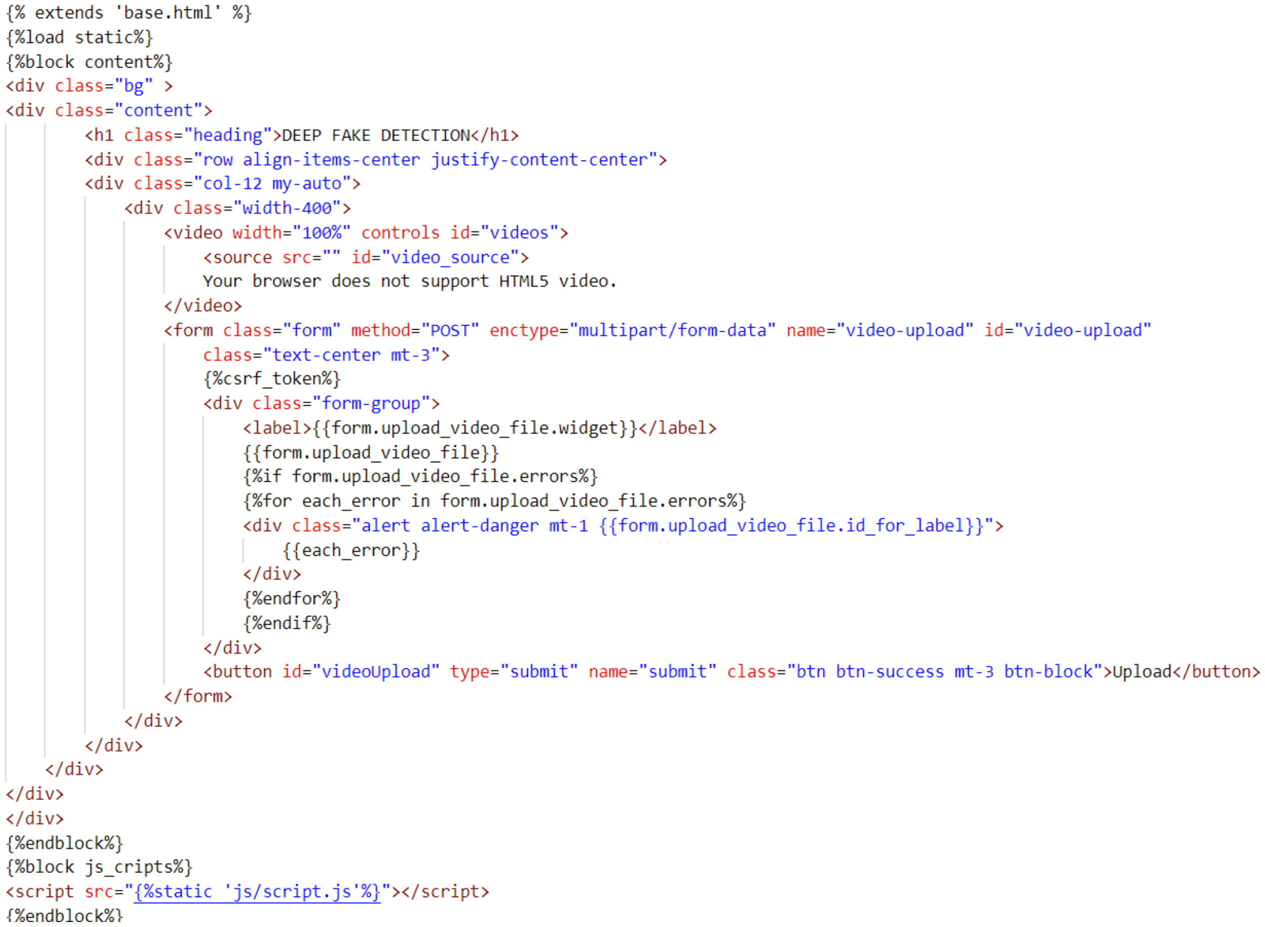
Fig B.5 shows the HTML code for the nav bar that will be displayed at the top of the website with which the user can navigate between the three pages: about page, home page and the statistics page.



```
<!DOCTYPE html>
<html lang="en">
{% load static %}
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
            integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">
    <!-- <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"> -->
    <link rel="stylesheet" href="/static/css/jquery-ui.css">
    <link rel="stylesheet" href="/static/css/stylezz.css">
    <title>Deepfake Detection</title>
</head>
<body class="{% block body_class %}{% endblock %}">
    {%include 'nav-bar.html'%}
    {%block content%}
    {%endblock%}
    <!-- <script src="https://code.jquery.com/jquery-3.4.1.min.js"
        integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo=" crossorigin="anonymous"></script> -->
        <script src="/static/js/jquery-3.4.1.min.js" ></script>
    <!-- <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
        integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
        crossorigin="anonymous"></script> -->
        <script src="/static/js/popper.min.js"></script>
    <!-- <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"
        integrity="sha256-VazP97ZCwtekAsvgPBSUwPFKdrwD3unUfSGVYrahUqU=" crossorigin="anonymous"></script> -->
        <script src="/static/js/jquery-ui.min.js"></script>
        <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    {%block js_cripts%}
</body>

</html>
```

Fig B.6 base.html file

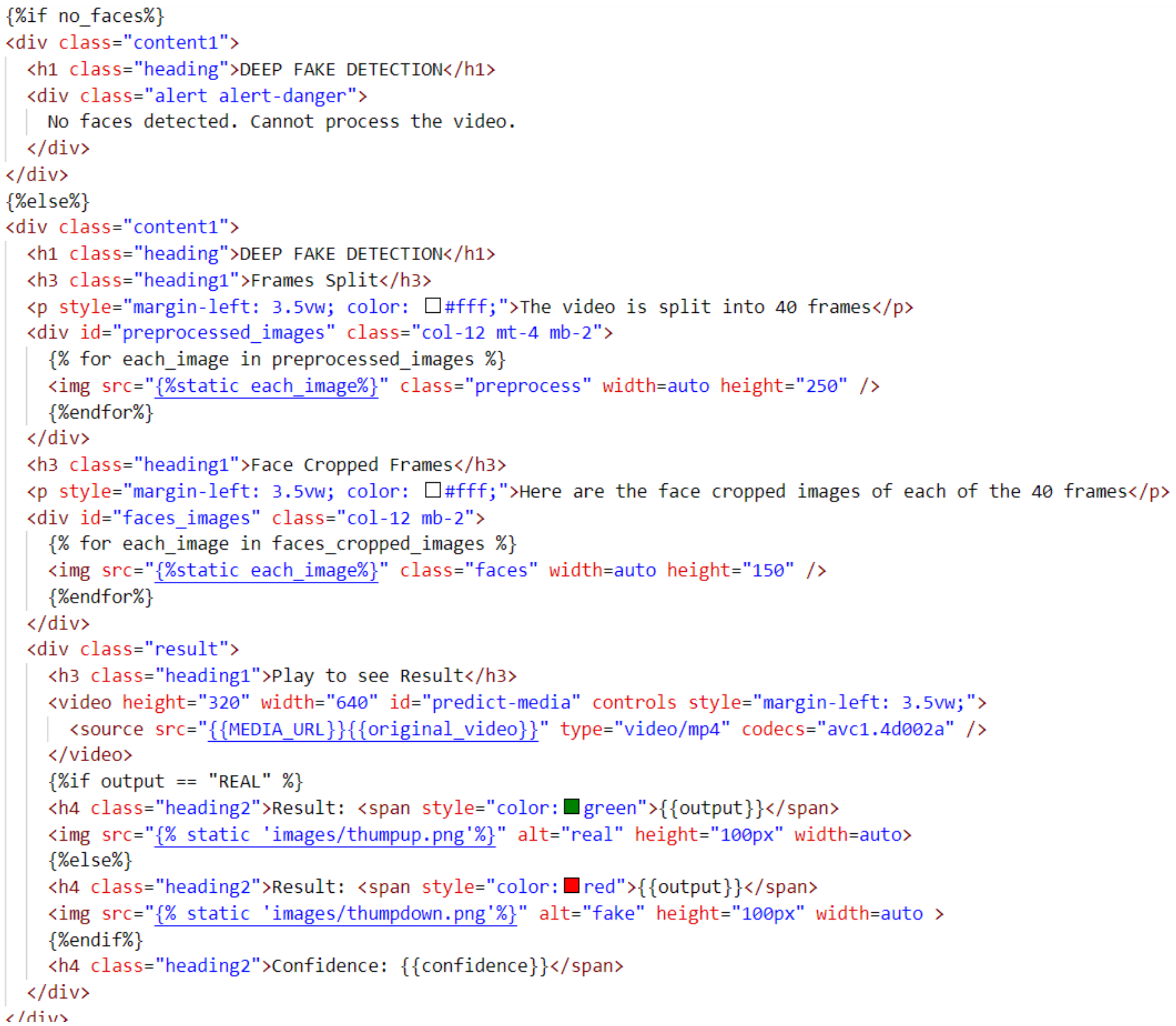
Fig B.6 shows the HTML code for base.html file which will be the base of all the webpages. It also contains the scripts to various javascript codes which will help in the smooth functionality of the page



```
{% extends 'base.html' %}
{%load static%}
{%block content%}
<div class="bg" >
<div class="content">
        <h1 class="heading">DEEP FAKE DETECTION</h1>
        <div class="row align-items-center justify-content-center">
        <div class="col-12 my-auto">
            <div class="width-400">
                <video width="100%" controls id="videos">
                    <source src="" id="video_source">
                    Your browser does not support HTML5 video.
                </video>
                <form class="form" method="POST" enctype="multipart/form-data" name="video-upload" id="video-upload"
                    class="text-center mt-3">
                    {%csrf_token%}
                    <div class="form-group">
                        <label>{{form.upload_video_file.widget}}</label>
                        {{form.upload_video_file}}
                        {%if form.upload_video_file.errors%}
                        {%for each_error in form.upload_video_file.errors%}
                        <div class="alert alert-danger mt-1 {{form.upload_video_file.id_for_label}}">
                            {{each_error}}
                        </div>
                        {%endfor%}
                        {%endif%}
                    </div>
                    <button id="videoUpload" type="submit" name="submit" class="btn btn-success mt-3 btn-block">Upload</button>
                </form>
            </div>
        </div>
    </div>
</div>
</div>
{%endblock%}
{%block js_cripts%}
<script src="{%static 'js/script.js'%}"></script>
{%endblock%}
```

Fig B.7 index.html file

Fig B.7 shows the HTML code for index.html file which contains a form where the user can upload the video for deepfake detection. After uploading the video, the user clicks on “Upload” button for the video to be sent for preprocessing before going for model prediction.



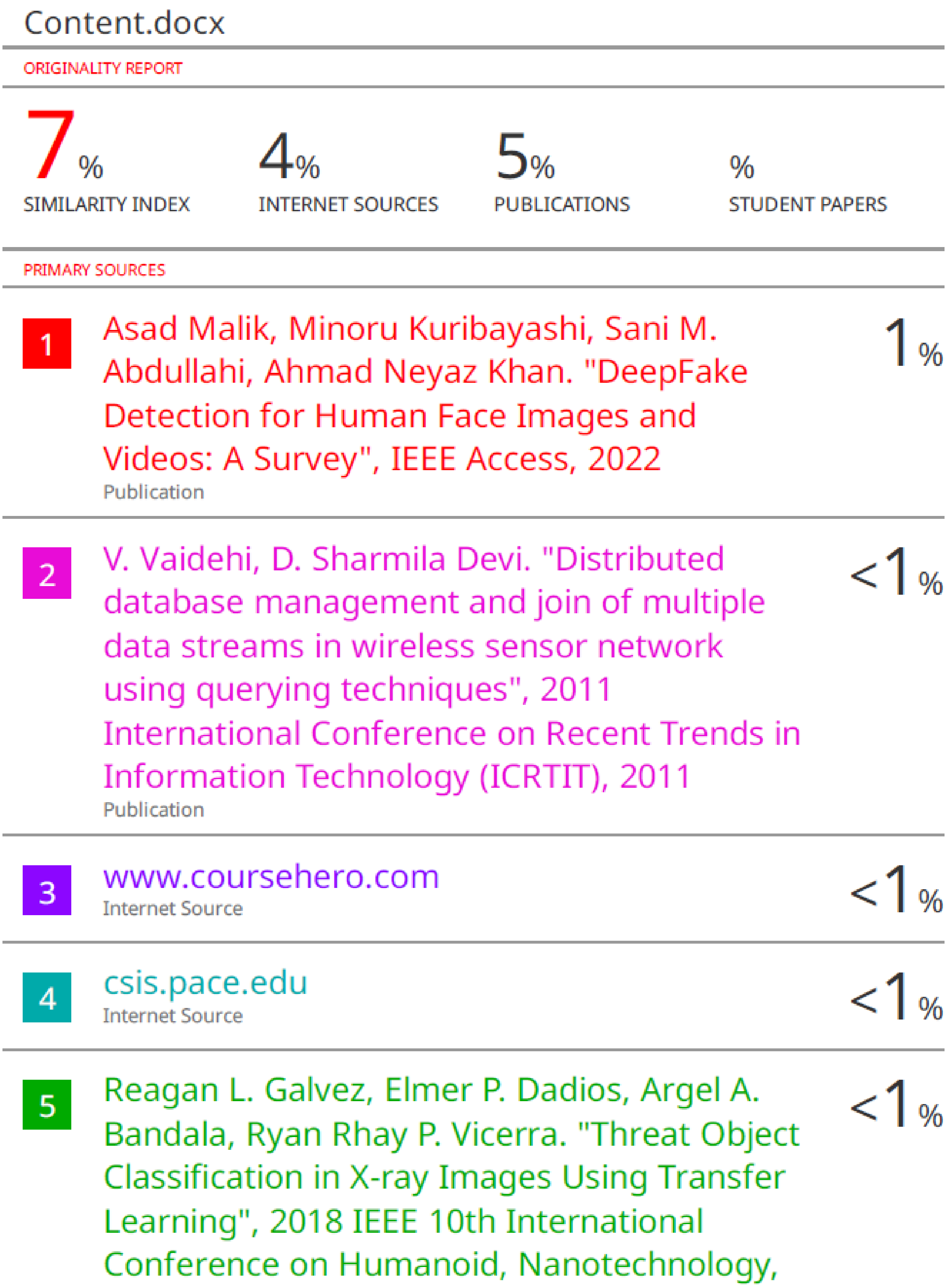
```
{%if no_faces%}
<div class="content1">
  <h1 class="heading">DEEP FAKE DETECTION</h1>
  <div class="alert alert-danger">
    No faces detected. Cannot process the video.
  </div>
</div>
{%else%}
<div class="content1">
  <h1 class="heading">DEEP FAKE DETECTION</h1>
  <h3 class="heading1">Frames Split</h3>
  <p style="margin-left: 3.5vw; color: □#fff;">The video is split into 40 frames</p>
  <div id="preprocessed_images" class="col-12 mt-4 mb-2">
    {% for each_image in preprocessed_images %}
    <img src="{%static each_image%}" class="preprocess" width=auto height="250" />
    {%endfor%}
  </div>
  <h3 class="heading1">Face Cropped Frames</h3>
  <p style="margin-left: 3.5vw; color: □#fff;">Here are the face cropped images of each of the 40 frames</p>
  <div id="faces_images" class="col-12 mb-2">
    {% for each_image in faces_cropped_images %}
    <img src="{%static each_image%}" class="faces" width=auto height="150" />
    {%endfor%}
  </div>
  <div class="result">
    <h3 class="heading1">Play to see Result</h3>
    <video height="320" width="640" id="predict-media" controls style="margin-left: 3.5vw;">
      <source src="{{MEDIA_URL}}{{original_video}}" type="video/mp4" codecs="avc1.4d002a" />
    </video>
    {%if output == "REAL" %}
    <h4 class="heading2">Result: <span style="color:🟩green">{{output}}</span>
    <img src="{% static 'images/thumpup.png'%}" alt="real" height="100px" width=auto>
    {%else%}
    <h4 class="heading2">Result: <span style="color:🟥red">{{output}}</span>
    <img src="{% static 'images/thumpdown.png'%}" alt="fake" height="100px" width=auto >
    {%endif%}
    <h4 class="heading2">Confidence: {{confidence}}</span>
  </div>
</div>
```

Fig B.8 predict.html file

Fig B.8 shows the HTML code for predict.html file where the user is shown the first 40 frames of the video, along with the face cropped images of the 40 frames and then the prediction made by the model, whether the uploaded video is real or fake according to the model along with the confidence which shows how sure the model is of the prediction made by it

APPENDIX C

PLAGIARISM REPORT



Content.docx

ORIGINALITY REPORT

7%	4%	5%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Asad Malik, Minoru Kuribayashi, Sani M. Abdullahi, Ahmad Neyaz Khan. "DeepFake Detection for Human Face Images and Videos: A Survey", IEEE Access, 2022 Publication	1 %
2	V. Vaidehi, D. Sharmila Devi. "Distributed database management and join of multiple data streams in wireless sensor network using querying techniques", 2011 International Conference on Recent Trends in Information Technology (ICRTIT), 2011 Publication	<1 %
3	www.coursehero.com Internet Source	<1 %
4	csis.pace.edu Internet Source	<1 %
5	Reagan L. Galvez, Elmer P. Dadios, Argel A. Bandala, Ryan Rhay P. Vicerra. "Threat Object Classification in X-ray Images Using Transfer Learning", 2018 IEEE 10th International Conference on Humanoid, Nanotechnology,	<1 %

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE SYNOPSIS/THESIS/DISSERTATION/PROJECT REPORTS

1	Name of the Candidate **(IN BLOCK LETTERS)**	**SHRESTH GUPTA** **UTKARSH SRIVASTAVA**
2	Address of the Candidate	**Mobile Number :6206048094** **Mobile Number :8545831533**
3	Registration Number	RA2011026010091 RA2011026010104
4	Date of Birth	13th June, 2002 18th January, 2002
5	Department	Computational Intelligence (CINTEL)
6	Faculty	Faculty of Engineering and Technology
7	Title of the Synopsis/ Thesis/ Dissertation/Project	Deepfake Detection using ResNext101 and LSTM
8	Name and address of the Supervisor / Guide	Dr. Antony Sophia N Assistant Professor Department of Computational Intelligence **Mail ID :** antonysn@srmist.edu.in **Mobile Number :** 9894119157
9	Name and address of the Co-Supervisor / Co- Guide (if any)	**Mail ID :** **Mobile Number :**
10	Software Used	Turnitin
11	Date of Verification	2nd April 2024

12	**Plagiarism Details: (to attach the final report)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	INTRODUCTION	1%	1%	1%
2	BACKLOG REFINEMENT	1.5%	1.5%	1.5%
3	SPRINT PLANNING	2.5%	2.5%	2.5%
4	RESULTS AND DISCUSSION	1%	1%	1%
5	CONCLUSION AND FUTURE ENHANCEMENT	<1%	<1%	<1%
6	REFERENCES	<1%	<1%	<1%
Thesis abstract		<1%	<1%	<1%
Appendices				
I/We declare that the above information have been verified and found true to the best of my/our knowledge.				
Signature of the Candidate		**Signature of the Supervisor / Guide**		
Signature of the Co-Supervisor/Co-Guide		**Signature of the HOD**		