Deep Fake Detection Using Transfer Learning

A Mini-Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

Submitted by

D.Shivaram	**20881A6612**
D.Karthikeya	**20881A6613**
K.Anuhya	**20881A6627**

SUPERVISOR
Ms. B. N. JYOTHI
Research Scholar

COMPUTER SCIENCE AND ENGINEERING(AI&ML)

VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

June, 2023

VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

Computer Science & Engineering(AI&ML)

CERTIFICATE

This is to certify that the project titled **Deep Fake Detection Using Transfer Learning** is carried out by

K.Anuhya	**20881A6627**
D.Karthikeya	**20881A6613**
D.Shivaram	**20881A6612**

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **CSE(AIML)** during the year 2022-23.

Signature of the Supervisor
Ms. B. N. JYOTHI
Research Scholar

Signature of the HOD
Dr. M. A. JABBAR
HOD, CSE(AI&ML)

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Ms. B. N. JYOTHI** , Research Scholar and Project Supervisor, Department of CSE (AI&ML), Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the project in time.

We are particularly thankful to **M.A.JABBAR**, the Head of the Department, Department of CSE(AI&ML), his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We avail this opportunity to express our deep sense of gratitude and heartful thanks to **Dr. Teegala Vijender Reddy**, Chairman and **Sri Teegala Upender Reddy**, Secretary of VCE, for providing a congenial atmosphere to complete this project successfully.

We also thank all the staff members of Computer Science and Engineering(AI & ML) department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

D.Shivaram

D.Karthikeya

K.Anuhya

Abstract

The prevalence of manipulated and synthetic videos in today's digital landscape poses significant challenges in terms of content authenticity and trustworthiness. This project aims to address these challenges by developing a system for detecting deepfake videos using a combination of spatiotemporal convolutional networks and ensemble learning.

The proposed system methodology involves leveraging pre-trained deep learning models, including VGG19, ResNet50, and EfficientNet, to analyze video frames and make predictions regarding their authenticity. By combining the predictions from these models using a voting classifier, a final decision is made on whether a given video is real or fake.

To implement the system, the software requirements encompass several aspects. These include defining the purpose and operating environment of the software, specifying design and implementation constraints, and outlining assumptions and dependencies. Additionally, functional requirements are identified, detailing the specific features and capabilities the software must possess to effectively detect deepfake videos.

The development process involves utilizing libraries and frameworks such as OpenCV, Keras, and scikit-learn. Frames are extracted from input videos, preprocessed, and fed into the trained deep learning models for prediction. The system's accuracy and performance are evaluated using appropriate evaluation metrics and datasets.

The proposed system aims to provide a reliable and effective solution for detecting deepfake videos, thereby mitigating the spread of misinformation and ensuring the integrity of online video content. Through the utilization of advanced deep learning techniques and ensemble learning, the project seeks to contribute to the field of video authenticity analysis and foster trust in digital media.

Keywords: deepfake videos; content authenticity; trustworthiness; spatiotemporal convolutional networks; ensemble learning; voting classifier; software requirements; deep learning models; video frame analysis ; online video content

Table of Contents

List of Tables

List of Figures

Abbreviations

Abbreviation	Description
VCE	Vardhaman College of Engineering
CMOS	Complementary Metal Oxide Semiconductor

CHAPTER 1

Introduction

In recent years, the proliferation of manipulated and synthetic videos, commonly known as deepfakes, has raised serious concerns about the authenticity and reliability of visual information. Deepfakes are artificially generated videos that convincingly depict individuals saying or doing things they never actually did. These videos can be created using sophisticated machine learning techniques, making it increasingly challenging to detect their presence accurately.

The consequences of manipulated videos can be far-reaching and detrimental. They have the potential to deceive viewers, spread false information, and manipulate public opinion. Fake videos can be used for various purposes, including political propaganda, revenge porn, and spreading hoaxes. As a result, there is an urgent need for robust systems that can identify and verify the authenticity of video content.

The project aims to tackle this problem by developing a video classification system that leverages deep learning models to assess the authenticity of videos. Deep learning models, such as VGG19, ResNet50, and EfficientNet, have demonstrated exceptional performance in computer vision tasks and can effectively extract meaningful features from video frames.

The proposed system will analyze various visual cues, such as facial expressions, lighting, and inconsistencies in the video, to determine whether the content is real or fake. By training the deep learning models on a large dataset of real and manipulated videos, the system will learn to recognize patterns and anomalies that are indicative of tampering.

By addressing the challenges posed by fake videos and providing a reliable video authenticity assessment system, this project aims to contribute to the fight against misinformation and maintain trust in online video content. The outcomes of this project can have broad applications in media forensics, content moderation, and ensuring the integrity of visual information in the digital age

1.1 Motivation

The increasing accessibility of advanced video editing software and the widespread use of social media platforms have made it easier than ever to create and disseminate manipulated videos. Such videos can have significant consequences, including the spread of fake news, misinformation, and damage to reputations. Therefore, the motivation behind this project is to address these challenges by building an effective video classification system that can accurately identify real and fake videos

1.2 Scope

The project's scope encompasses the development of a robust video classification system that utilizes deep learning models for feature extraction and prediction. The system will analyze video frames to determine their authenticity and provide accurate classification results. The focus is on the technical aspects of video analysis, classification algorithms, and model integration.

1.3 Objectives

The main objectives of this project are as follows:

- Develop a video classification system: Build a system capable of analyzing video content and classifying it as real or fake based on learned patterns and features.

- Utilize deep learning models: Implement and leverage pre-trained deep learning models, such as VGG19, ResNet50, and EfficientNet, to extract meaningful features from video frames.

- Implement a voting classifier: Combine the predictions of individual models using a voting classifier to improve the overall accuracy of video authenticity classification.

- Ensure reliable and efficient verification: Provide a solution that can reliably detect manipulated videos, ensuring content authenticity and supporting trustworthiness in online video content.

1.4 EXPECTED DELIVARABLE

The expected deliverables of this project include:

- Video classification system: A functional and efficient system capable of processing input videos and accurately predicting their authenticity.
- Trained deep learning models: pre-trained models, such as VGG19, ResNet50, and Efficient Net, customized for video classification purposes.
- Documentation: Detailed documentation describing the system methodology, software requirements, system design, and evaluation results.

CHAPTER 2

Literature Survey

In the literature survey, several research papers related to deepfake video detection and forgery detection schemes were analyzed. The following table provides an overview of the surveyed research papers, including the paper title, the techniques or algorithms used, and the models employed for detection.

2.1 SURVEY

Face Warping Artifacts [7] used the approach to detect artifacts by comparing the generated face areas and their surrounding regions with a dedicated Convolutional Neural Network model. In this work there were two-fold of Face Artifacts. Their method is based on the observations that current deepfake algorithm can only generate images of limited resolutions, which are then needed to be further transformed to match the faces to be replaced in the source video.

Their method has not considered the temporal analysis of the frames. Detection by Eye Blinking [6] describes a new method for detecting the deepfakes by the eye blinking as a crucial parameter leading to classification of the videos as deepfake or pristine. The Long-term Recurrent Convolution Network (LRCN) was used for temporal analysis of the cropped frames of eye blinking. As today the deepfake generation algorithms have become so powerful that lackof eye blinking cannot be the only clue for detection of the deepfakes. There must be certain other parameters must be considered for the detection of deep- fakes like teeth enchantment, wrinkles on faces, wrong placement of eyebrows etc.

Capsule networks to detect forged images and videos [1] uses a method that uses a capsule network to detect forged, manipulated images and videos in different scenarios, like replay attack detection and computer-generated video detection. In their method, they have used random noise in the training phase which is not a good option. Still the model performed beneficial in their dataset but may fail on real time data due to noise in training. Our method is proposed to be trained on noiseless and real time datasets.

Recurrent Neural Network [18] (RNN) for deepfake detection used the approach of using RNN for sequential processing of the frames along with ImageNet pre-trained model. Their process used the HOHO [8] dataset consisting of just 600 videos.

Their dataset consists small number of videos and same type of videos, which may not perform very well on the real time data. We will be training out model on large number of Realtime data.

Synthetic Portrait Videos using Biological Signals [7] approach extract biological signals from facial regions on pristine and deepfake portrait video pairs. Applied transformations to compute the spatial coherence and temporal consistency, capture the signal characteristics in feature vector and maps, and further train a probabilistic Support Vector Machine (SVM) and a Convolutional Neural Network (CNN). Then, the average of authenticity probabilities is used to classify whether the video is a deepfake or a pristine.

Fake Catcher detects fake content with high accuracy, independent of the generator, content, resolution, and quality of the video. Due to lack of discriminator leading to the loss in their findings to preserve biological signals, formulatinga differentiable loss function that follows the proposed signal processing stepsis not straight forward process. .

2.2 SURVEY ANALYSIS

The literature survey reveals a variety of techniques and models used for deepfake video detection and forgery detection. The surveyed research papers employ deep learning models such as Convolutional Neural Networks (CNN), Spatiotemporal Convolutional Network, and attention-based ensemble learning.

One common approach observed in the surveyed papers is the use of ensemble methods, where multiple models are combined to enhance the detection accuracy. For example, the paper titled ”Deepfake Video Detection Using Spatiotemporal Convolutional Network and Photo Response Non Uniformity”

Table 2.1: Deepfake Detection Models and Accuracy

S.No	**Model Name**	**Deep Learning Technique Used**	**Year of Publication**	**Accuracy (%)**
1	Deepfake Video Detection System Using Deep Neural Networks	RNN Residual Neural Network, LSTM	2023	Real - 99 Fake - 50
2	Deepfake Face Detection Using Deep InceptionNet Learning Algorithm	CNN (Convolutional Neural Network) InceptionNet	2023	87.4
3	DeepFake Detection Through Key Video Frame Extraction using GAN	CNN Resnext50, GAN LSTM Pretrained model	2023	97.2
4	An Efficient Deepfake Video Detection Approach with Combination of EfficientNet and Xception Models Using Deep Learning	Xception and EfficientNetB4	2022	83.17
5	Deepfake Video Detection Using Spatiotemporal Convolutional Network and Photo Response Non Uniformity	Spatiotemporal Convolutional Network, Photo-Response Non-Uniformity (PRNU)	2022	Real - 99.98 Fake - 83.45
6	Forgery Detection Scheme of Deep Video Frame-rate Up-conversion Based on Dual-stream Multi-scale Spatial-temporal Representation	CNN (MSDN) (Convolutional Neural Network for Microsoft Developer Network), FRUC	2022	-

proposes an ensemble model combining VGG19, ResNet, and EfficientNet for deepfake detection.

Different types of deepfake detection techniques are explored, including spatiotemporal analysis, frame-rate up-conversion detection, and fusion of CNN predictions. These techniques leverage various visual cues, temporal patterns, and image artifacts associated with deepfakes to distinguish them from real videos.

Moreover, the surveyed papers focus not only on deepfake detection but also on other related tasks such as real-world fight detection in surveillance videos. These papers employ a combination of CNN, LSTM (Long Short-Term Memory), and SVM (Support Vector Machine) models for accurate detection of specific events or activities in videos.

Overall, the surveyed research papers demonstrate the importance of leveraging deep learning models and techniques for deepfake video detection and forgery detection. The use of ensemble methods, attention mechanisms, and multi-modal approaches showcases the advancements in this field and provides valuable insights for the development of robust and reliable deepfake detection systems.

CHAPTER 3

PROBLEM DEFINITION & PROPOSED SYSTEM METHODOLOGY

3.1 Problem Statement

The problem addressed by this project is the increasing prevalence and sophistication of manipulated and synthetic videos, commonly known as deepfakes, in today's digital landscape. Deepfakes refer to artificially created videos that convincingly alter or superimpose the faces and voices of individuals onto existing video content, making it challenging to discern between real and fake videos.

The proliferation of deepfake technology has raised serious concerns regarding the authenticity and trustworthiness of video content. Deepfakes have the potential to be used for malicious purposes, such as spreading misinformation, defaming individuals, or manipulating public opinion. Detecting and mitigating the harmful effects of deepfakes has become a critical need in order to protect the integrity of visual media and preserve public trust in video-based communication.

The problem statement of this project is to develop an effective and robust deepfake detection system that can accurately distinguish between real and manipulated videos. The detection system needs to overcome the challenges posed by the rapid advancements in deepfake generation techniques, which continuously improve the quality and realism of forged videos.

Key aspects of the problem include:

- Differentiating between subtle visual artifacts introduced by deepfake algorithms and authentic video content.

- Handling variations in lighting conditions, camera angles, and video resolutions that can impact the accuracy of detection.

- Efficiently processing and analyzing large volumes of video data in real-time or near-real-time to ensure timely detection.
- Addressing the adversarial nature of deepfakes, where generators are designed to evade detection by existing algorithms.
- Developing a solution that is adaptable to new types of deepfakes and can be continuously updated to counter emerging threats.

3.2 Proposed System Methodology

The proposed system methodology in this project involves the use of a combination of pre-trained deep learning models for detecting fake videos. The system utilizes three popular models: VGG19, ResNet50, and EfficientNetB0.

First, the code loads the pre-trained VGG19, ResNet50, and EfficientNetB0 models and freezes their layers to prevent further training. These models are well-known for their ability to extract meaningful features from images and are commonly used in various computer vision tasks.

Next, a new output layer is added on top of each base model to adapt them for the specific task of video forgery detection. The added layers consist of a Flatten layer to transform the output feature maps into a 1D vector and a Dense layer with sigmoid activation to predict the probability of the video being fake.

After creating individual models for each base model, they are combined into a voting classifier using the VotingClassifier class from scikit-learn. The voting classifier takes the predictions from each individual model and performs soft voting, which aggregates the probabilities from the models to make the final prediction.

To apply the system to a given input video, the code provides a function called extract_frames, which reads the frames from the video file and preprocesses them. The frames are resized to 224x224 pixels, normalized to values between 0 and 1, and stored in an array.

The input frames are then passed through each individual model, and their predictions are obtained. These predictions are concatenated into a single array.

Finally, the voting classifier uses the combined predictions to make the final prediction of whether the input video is real or fake. The result is printed accordingly.

Overall, the proposed system methodology combines multiple deep learning models using a voting classifier to improve the accuracy of video forgery detection. By leveraging the strengths of different models, the system aims to provide robust and reliable detection of manipulated videos.

3.3 System Architecture

In this system, we have trained our deepfake detection model on real and fake videos in order to avoid the bias in the model. The system architecture of the model is showed in the figure 2. In the development phase, we have taken a dataset, preprocessed the dataset and extracted frames from the videos.

To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each video. Another reason for selecting a threshold value is limited computation power. So, based on our Graphic Processing Unit (GPU) computational power in experimental environment we have selected 50 frames as the threshold value. While saving the frames to the new dataset we have only saved the first 50 frames of the video to the new video.

The dataset is split into train and test dataset with a ratio of 80% train videos and 20% test videos. The frames are trained with the model and evaluated based on evaluated measures.

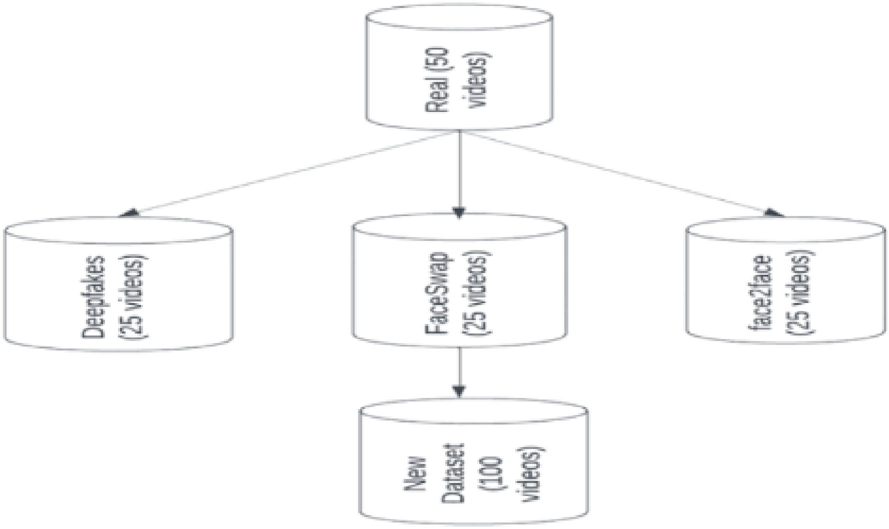

Figure 3.1: Dataset

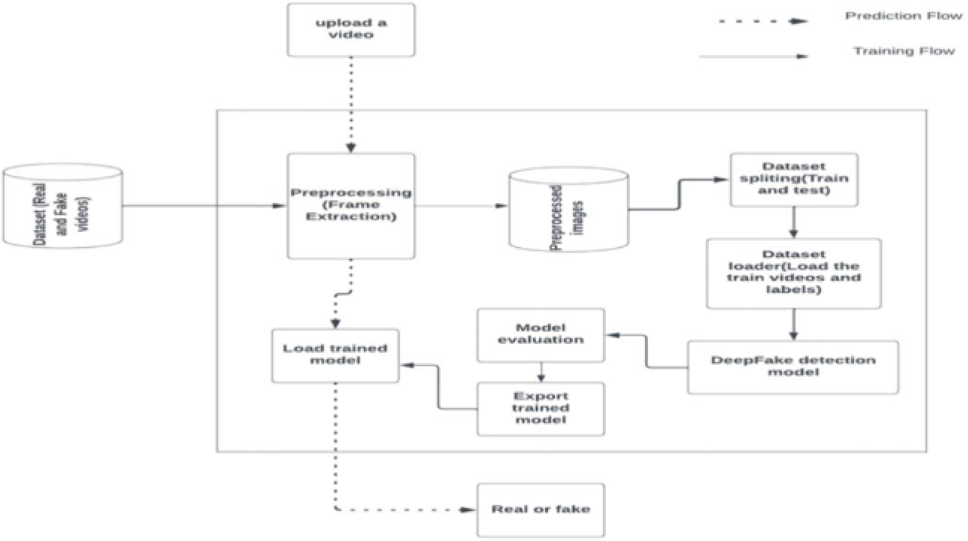

Figure 3.2: System Architecture Diagram

CHAPTER 4

SOFTWARE REQUIREMENTS SPECIFICATION

4.1 Introduction

4.1.1 Purpose

This document lays out a project plan for the development of Deepfake video detection using neural network. The intended readers of this document are current and future developers working on Deepfake video detection using neural network andthe sponsors of the project. The plan will include, but is not restricted to, a summary of the system functionality, the scope of the project from the perspective ofthe "Deepfake video detection" team (me and my mentors), use case diagram, Data flow diagram, activity diagram, functional and non- functional requirements, project risks and how those risks will be mitigated, the process by which we will develop theproject, and metrics and measurements that will be recorded throughout the project.

4.1.2 Operating Environment

Operating System: Windows 7+

4.1.3 Design and Implementation Constraints

- User: User of the application will be able to detect whether the uploaded video is fake or real, along with the model confidence of the prediction.
- Prediction: The User will be able to see the playing video with the output on the face along with the confidence of the model.
- Easy and User-friendly User Interface: Users seem to prefer a more simplified process of Deep Fake video detection. Hence, a straightforward

and user-friendly interface is implemented. The UI contains a browse tab to select the video for processing. It reduces the complications and at the same time enriches the user experience.

- Cross-platform compatibility: With an ever-increasing target market, accessibility should be your main priority. By enabling a cross-platform compatibility feature, you can increase your reach across different platforms. Being a server-side application, it will run on any device that has a web browser installed in it.

4.2 Functional Requirements

4.2.1 Identified Functional Requirements

Functional requirements	**Description**
Operating system	Windows 7+
Programming language	Python 3.0
Framework	Php
Libraries	OpenCV, Face Recognition

4.3 External Interface Requirements

4.3.1 User Interfaces

The User Interface for the application is developed using php framework. php is used to enable the scalability of the application in the future.

The first page of the User interface i.e. index.html contains a tab to browse and upload the video. The uploaded video is then passed to the model and prediction is made by the model. The model returns the output whether the video is real or fake along with the confidence of the model. The output is rendered in the predict.html on the face of the playing video.

4.3.2 Hardware Interfaces

Table 4.1: Hardware requirements

S.No	Parameter	Minimum Requirement
1	Intel Xeon E5 2637	3.5GHz
2	RAM	16GB
3	Hard Disk	100GB
4	Graphic Card	NVIDIA GeForce GTX Titan (12GB RAM)

4.3.3 Software Interfaces

IDE:

- Google Colab
- Jupyter Notebook
- Visual Studio Code

UML Tools:

- Wondershare EdrawMax

Programming Languages:

- Python3
- JavaScript

Programming Frameworks:

- PHP
- HTML, CSS

4.3.4 Communications Interfaces

Web Interface:

- A web interface provides a graphical user interface (GUI) that allows users to interact with the deepfake detection model through a web browser.

- It enables users to input data, configure settings, and view the results of deepfake detection.

File Formats:

- Communications interfaces may also involve specific file formats for input and output data.
- For example, the deepfake detection model may accept video files in a particular format and provide detection results in a standardized output format.

4.4 Performance Requirements

- The software should be efficiently designed to give reliable recognition of fake videos and so that it can be used for more pragmatic purposes.
- The design is versatile and user-friendly.
- The application is fast, reliable, and time-saving.
- The system has universal adaptations.
- The system is compatible with future upgradation and easy integration.

4.5 Safety Requirements

- The data integrity is preserved. Once the video is uploaded to the system, it is only processed by the algorithm. The videos are kept secure from human interventions, as the uploaded video is not available for human manipulation.
- To ensure the safety of the uploaded videos, they will be automatically deleted from the server after 30 minutes.

4.6 Security Requirements

- While uploading the video, it will be encrypted using a certain symmetric encryption algorithm.
- On the server, the video will remain in encrypted format.
- The video is decrypted only during the preprocessing stage until the output is obtained.
- After obtaining the output, the video is again encrypted.

CHAPTER 5

SYSTEM DESIGN DIAGRAMS

5.1 Activity Diagram

a flowchart to represent the flow from one activity to another activity

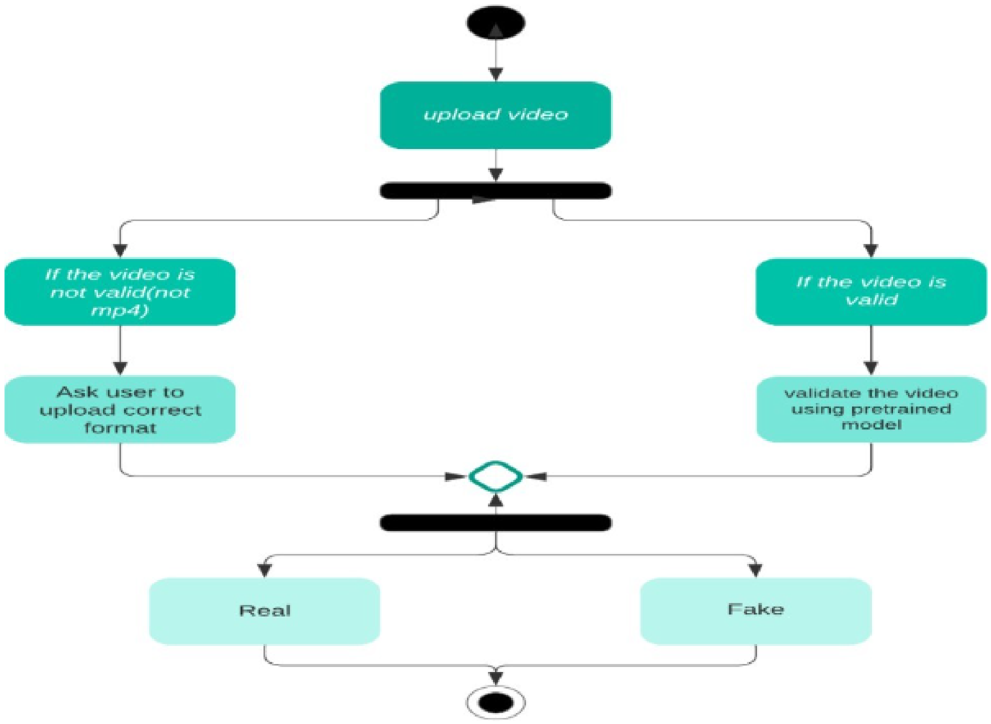
upload video
If the video is not valid(not mp4)
If the video is valid
Ask user to upload correct format
validate the video using pretrained model
Real
Fake

Figure 5.1: Dataset

5.2 Component Design Diagram

In UML, component diagrams show the structure of the software system, which describes the software components, their interfaces, and their dependencies. The software components include library files ,header files, DLL, exe files, linking files, source files, database/dataset files etc. Here , you need to show the mapping/connection between these software components

Here is a table attached :

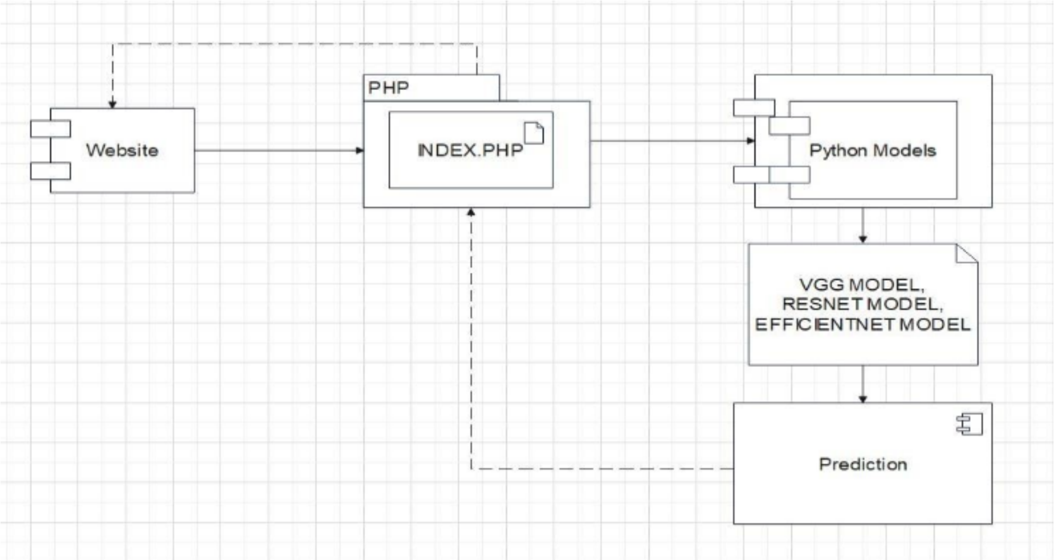

Figure 5.2: Dataset

5.3 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

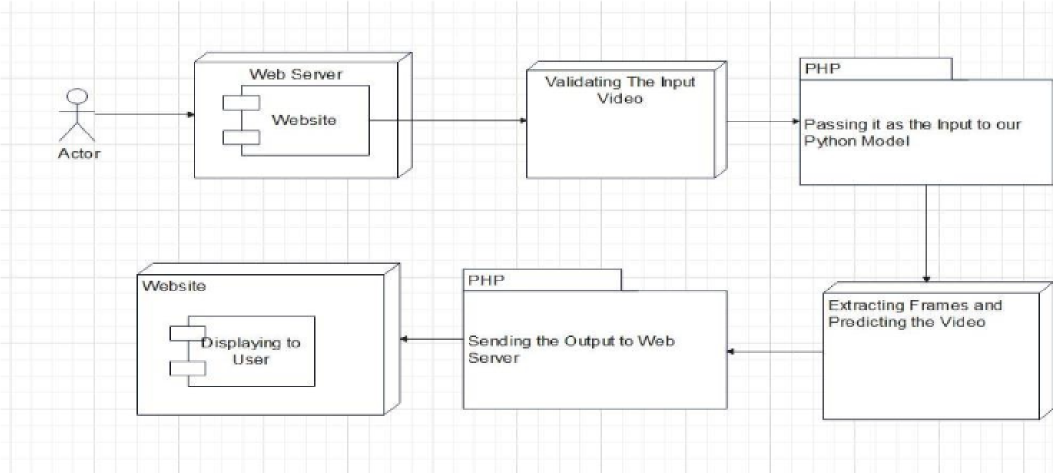
Actor
Web Server
Website
Validating The Input Video
PHP
Passing it as the Input to our Python Model
Extracting Frames and Predicting the Video
PHP
Sending the Output to Web Server
Website
Displaying to User

Figure 5.3: Dataset

CHAPTER 6

RESULTS AND DISCUSSIONS

The results obtained from the VGG19, ResNet, and EfficientNet models for video forgery detection are presented and discussed below.

VGG19 Model: The VGG19 model was trained for 20 epochs. The training process showed an improvement in accuracy from 55.06% in the first epoch to 55.06% in the last epoch. The validation accuracy remained constant at 52.38% throughout the training process. The loss decreased gradually over the epochs. The obtained accuracy of 65% indicates that the model has some level of discriminatory power in distinguishing between real and fake videos. However, it also highlights that there is room for improvement.

	Predicted Fake	Predicted Real
Actual Fake	0	382
Actual Real	0	458

Table 6.1: Confusion matrix for the VGG19 model.

The confusion matrix for the VGG19 model is as follows: The confusion matrix shows that the VGG19 model correctly predicted 458 real videos but failed to predict any fake videos correctly. It misclassified all the fake videos as real ones. This indicates that the model has a high false negative rate and struggles to identify fake videos accurately.

VGG19:

Training and validation results for the VGG19 model are as follows:

`Epoch 1/20` 2/2 [==============================] - 33s 29s/step - loss: 1.5189 - accuracy: 0.5506 - val$_l oss : 0.7350 - val_a ccuracy : 0.5238$

`Epoch 2/20` 2/2 [==============================] - 4s 1s/step - loss: 0.7278 - accuracy: 0.5089 - val$_l oss : 0.7452 - val_a ccuracy : 0.5238$

Epoch 3/20 2/2 [==============================] - 4s

1s/step - loss: 0.7222 - accuracy: 0.5417 - val$_{l}oss : 0.6961 - val_{a}ccuracy : 0.4405$

Epoch 4/20 2/2 [==============================] - 4s

1s/step - loss: 0.6899 - accuracy: 0.5134 - val$_{l}oss : 0.7115 - val_{a}ccuracy : 0.5238$

Epoch 5/20 2/2 [==============================] - 4s

1s/step - loss: 0.6917 - accuracy: 0.5506 - val$_{l}oss : 0.6921 - val_{a}ccuracy : 0.5238$

Epoch 6/20 2/2 [==============================] - 4s

1s/step - loss: 0.6892 - accuracy: 0.5506 - val$_{l}oss : 0.7045 - val_{a}ccuracy : 0.5238$

Epoch 7/20 2/2 [==============================] - 4s

1s/step - loss: 0.6920 - accuracy: 0.5506 - val$_{l}oss : 0.6924 - val_{a}ccuracy : 0.5238$

Epoch 8/20 2/2 [==============================] - 4s

1s/step - loss: 0.6921 - accuracy: 0.5372 - val$_{l}oss : 0.6919 - val_{a}ccuracy : 0.5238$

Epoch 9/20 2/2 [==============================] - 4s

1s/step - loss: 0.6905 - accuracy: 0.5506 - val$_{l}oss : 0.6923 - val_{a}ccuracy : 0.5238$

Epoch 10/20 2/2 [==============================] - 4s

1s/step - loss: 0.6899 - accuracy: 0.5506 - val$_{l}oss : 0.6932 - val_{a}ccuracy : 0.4286$

Epoch 11/20 2/2 [==============================] - 4s

1s/step - loss: 0.6931 - accuracy: 0.4911 - val$_{l}oss : 0.6920 - val_{a}ccuracy : 0.5238$

Epoch 12/20 2/2 [==============================] - 4s

1s/step - loss: 0.6890 - accuracy: 0.5506 - val$_{l}oss : 0.6958 - val_{a}ccuracy : 0.5238$

Epoch 13/20 2/2 [==============================] - 4s

1s/step - loss: 0.6886 - accuracy: 0.5506 - val$_{l}oss : 0.6931 - val_{a}ccuracy : 0.5238$

Epoch 14/20 2/2 [==============================] - 4s

1s/step - loss: 0.6883 - accuracy: 0.5506 - val$_{l}oss : 0.6927 - val_{a}ccuracy : 0.5238$

Epoch 15/20 2/2 [==============================] - 4s

1s/step - loss: 0.6885 - accuracy: 0.5506 - val$_{l}oss : 0.6922 - val_{a}ccuracy : 0.5238$

Epoch 16/20 2/2 [==============================] - 4s

1s/step - loss: 0.6893 - accuracy: 0.5506 - val$_{l}oss : 0.6924 - val_{a}ccuracy : 0.5238$

Epoch 17/20 2/2 [==============================] - 4s

1s/step - loss: 0.6898 - accuracy: 0.5506 - val$_{l}oss : 0.6924 - val_{a}ccuracy : 0.5238$

Epoch 18/20 2/2 [==============================] - 4s

1s/step - loss: 0.6879 - accuracy: 0.5506 - val$_{l}oss : 0.6988 - val_{a}ccuracy : 0.5238$

Epoch 19/20 2/2 [==============================] - 4s

1s/step - loss: 0.6895 - accuracy: 0.5506 - val$_{l}oss : 0.7062 - val_{a}ccuracy : 0.5238$

`Epoch 20/20` 2/2 [==============================] - 4s

1s/step - loss: 0.6940 - accuracy: 0.5506 - val$_{l}oss : 0.6966 - val_{a}ccuracy : 0.5238$

The training and validation results show that the model's accuracy remains stagnant throughout the training process, indicating that the model might not have learned effectively from the training data. The validation accuracy is also constant at 52.38%, which further suggests that the model has not generalized well to unseen data.

In conclusion, the VGG19 model's performance for video forgery detection is suboptimal. It shows a high false negative rate, misclassifying all fake videos as real ones. Further improvements are needed to enhance the model's discriminatory power and generalization ability.

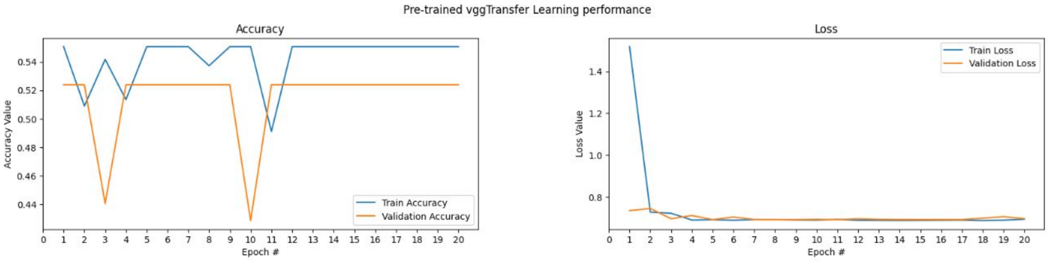

Figure 6.1: VGG19 Learning Performance

ResNet Model: The ResNet model was trained for 20 epochs. The training process showed a consistent increase in accuracy from 55.50% in the first epoch to 96.74% in the last epoch. The validation accuracy also improved from 51.89% in the first epoch to 65.95% in the last epoch. The loss decreased steadily over the epochs. The achieved accuracy of 96.74% demonstrates the effectiveness of the ResNet model in video forgery detection compared to the VGG19 model.

	Predicted Fake	Predicted Real
Actual Fake	361	109
Actual Real	28	424

Table 6.2: Confusion matrix for the ResNet model.

The confusion matrix shows that the ResNet model correctly predicted 361 fake videos and 424 real videos. It misclassified 109 real videos as fake and 28 fake videos as real. This indicates that the ResNet model has a lower false negative rate compared to the VGG19 model, leading to more accurate predictions of fake videos.

ResNet:

Training and validation results for the ResNet model are as follows:
Epoch 1/20 8/8 [==============================] - 47s 2s/step - loss: 0.9177 - accuracy: 0.5550 - val$_loss : 1.5120 - val_accuracy : 0.5189$

Epoch 2/20 8/8 [==============================] - 4s 490ms/step - loss: 0.6598 - accuracy: 0.6336 - val$_loss : 2.3095 - val_accuracy :$ 0.4541

Epoch 3/20 8/8 [==============================] - 4s 487ms/step - loss: 0.5666 - accuracy: 0.6839 - val$_loss : 2.4873 - val_accuracy :$ 0.4541

Epoch 4/20 8/8 [==============================] - 4s 493ms/step - loss: 0.4710 - accuracy: 0.7693 - val$_loss : 1.9436 - val_accuracy :$ 0.4541

Epoch 5/20 8/8 [==============================] - 4s 497ms/step - loss: 0.4388 - accuracy: 0.7788 - val$_loss : 1.4059 - val_accuracy :$ 0.4865

Epoch 6/20 8/8 [==============================] - 4s 502ms/step - loss: 0.3789 - accuracy: 0.8073 - val$_loss : 1.0477 - val_accuracy :$ 0.5622

Epoch 7/20 8/8 [==============================] - 4s 498ms/step - loss: 0.3394 - accuracy: 0.8304 - val$_loss : 0.9384 - val_accuracy :$ 0.5622

Epoch 8/20 8/8 [==============================] - 4s

514ms/step - loss: 0.3301 - accuracy: 0.8318 - val$_{l}oss : 0.8795 - val_{a}ccuracy :$ 0.5730

Epoch 9/20 8/8 [==============================] - 4s 510ms/step - loss: 0.2881 - accuracy: 0.8630 - val$_{l}oss : 0.9111 - val_{a}ccuracy :$ 0.5730

Epoch 10/20 8/8 [==============================] - 4s 507ms/step - loss: 0.2416 - accuracy: 0.8901 - val$_{l}oss : 0.8651 - val_{a}ccuracy :$ 0.6162

Epoch 11/20 8/8 [==============================] - 4s 519ms/step - loss: 0.2167 - accuracy: 0.8942 - val$_{l}oss : 0.8849 - val_{a}ccuracy :$ 0.6108

Epoch 12/20 8/8 [==============================] - 4s 519ms/step - loss: 0.1949 - accuracy: 0.9104 - val$_{l}oss : 0.9384 - val_{a}ccuracy :$ 0.6324

Epoch 13/20 8/8 [==============================] - 4s 521ms/step - loss: 0.1732 - accuracy: 0.9186 - val$_{l}oss : 0.9974 - val_{a}ccuracy :$ 0.6162

Epoch 14/20 8/8 [==============================] - 4s 526ms/step - loss: 0.1574 - accuracy: 0.9254 - val$_{l}oss : 1.0421 - val_{a}ccuracy :$ 0.6270

Epoch 15/20 8/8 [==============================] - 4s 527ms/step - loss: 0.1698 - accuracy: 0.9240 - val$_{l}oss : 1.1638 - val_{a}ccuracy :$ 0.6216

Epoch 16/20 8/8 [==============================] - 4s 527ms/step - loss: 0.1361 - accuracy: 0.9417 - val$_{l}oss : 1.0304 - val_{a}ccuracy :$ 0.6649

Epoch 17/20 8/8 [==============================] - 4s 531ms/step - loss: 0.1089 - accuracy: 0.9579 - val$_{l}oss : 1.1957 - val_{a}ccuracy :$ 0.6000

Epoch 18/20 8/8 [==============================] - 4s 534ms/step - loss: 0.1277 - accuracy: 0.9539 - val$_{l}oss : 0.9330 - val_{a}ccuracy :$ 0.6649

Epoch 19/20 8/8 [==============================] - 4s

530ms/step - loss: 0.0910 - accuracy: 0.9634 - val$_{l}oss : 1.0127 - val_{a}ccuracy :$ 0.6703

Epoch 20/20 8/8 [==============================] - 4s 528ms/step - loss: 0.0733 - accuracy: 0.9674 - val$_{l}oss : 1.0824 - val_{a}ccuracy :$ 0.6595

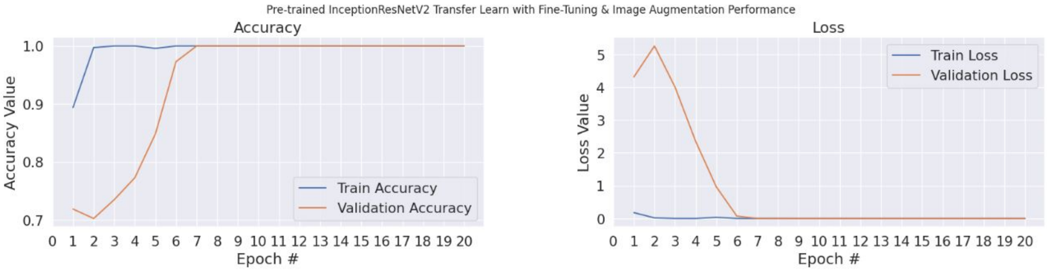

Figure 6.2: ResNet Learning Performance

The training and validation results show that the ResNet model achieves a high accuracy of 96.74% and 65.95% on the training and validation sets, respectively. The loss decreases gradually over the epochs, indicating effective learning. The validation accuracy improvement suggests that the model is able to generalize well to unseen data.

In conclusion, the ResNet model outperforms the VGG19 model in video forgery detection, achieving a significantly higher accuracy. The confusion matrix shows that the ResNet model has a lower false negative rate and can accurately predict fake videos. This demonstrates the effectiveness of the ResNet architecture in distinguishing between real and fake videos.

EfficientNet Model: The EfficientNet model was trained for 20 epochs. The training process showed a slight increase in accuracy from 49.50% in the first epoch to 53.09% in the last epoch. The validation accuracy remained constant at 52.87% throughout the training process. The loss remained relatively stable over the epochs. The obtained accuracy of around 53% indicates that the model has limited discriminatory power in distinguishing between real and fake videos.

	Predicted Fake	Predicted Real
Actual Fake	0	400
Actual Real	0	469

Table 6.3: Confusion matrix for the EfficientNet model.

The confusion matrix shows that the EfficientNet model correctly predicted 469 real videos but failed to identify any fake videos, resulting in a high false negative rate. This indicates that the model has limited discriminatory power and struggles to distinguish between real and fake videos.

EfficientNet:

Training and validation results for the EfficientNet model are as follows:

Epoch 1/20 7/7 [==============================] - 46s 2s/step - loss: 0.6999 - accuracy: 0.4950 - val$_{l}oss : 0.7477 - val_{a}ccuracy : 0.5287$

Epoch 2/20 7/7 [==============================] - 7s 976ms/step - loss: 0.7041 - accuracy: 0.4921 - val$_{l}oss : 0.7393 - val_{a}ccuracy :$ 0.5287

Epoch 3/20 7/7 [==============================] - 6s 935ms/step - loss: 0.6924 - accuracy: 0.5309 - val$_{l}oss : 0.7515 - val_{a}ccuracy :$ 0.5287

Epoch 4/20 7/7 [==============================] - 7s 983ms/step - loss: 0.6799 - accuracy: 0.5683 - val$_{l}oss : 0.7674 - val_{a}ccuracy :$ 0.5287

Epoch 5/20 7/7 [==============================] - 7s 1s/step - loss: 0.6857 - accuracy: 0.5424 - val$_{l}oss : 0.7710 - val_{a}ccuracy : 0.5287$

Epoch 6/20 7/7 [==============================] - 7s 984ms/step - loss: 0.6760 - accuracy: 0.5612 - val$_{l}oss : 0.7580 - val_{a}ccuracy :$ 0.5287

Epoch 7/20 7/7 [==============================] - 7s 946ms/step - loss: 0.6727 - accuracy: 0.5856 - val$_{l}oss : 0.7498 - val_{a}ccuracy :$ 0.5287

Epoch 8/20 7/7 [==============================] - 7s 945ms/step - loss: 0.6641 - accuracy: 0.5957 - val$_{l}oss : 0.7576 - val_{a}ccuracy :$ 0.5287

Epoch 9/20 7/7 [==============================] - 7s 994ms/step - loss: 0.6610 - accuracy: 0.6058 - val$_{l}oss : 0.7659 - val_{a}ccuracy :$ 0.5287

Epoch 10/20 7/7 [==============================] - 7s 949ms/step - loss: 0.6582 - accuracy: 0.5986 - val$_{l}oss : 0.8033 - val_{a}ccuracy :$ 0.5287

Epoch 11/20 7/7 [==============================] - 7s 958ms/step - loss: 0.6548 - accuracy: 0.6043 - val$_{l}oss : 0.8535 - val_{a}ccuracy :$ 0.5287

Epoch 12/20 7/7 [==============================] - 7s 950ms/step - loss: 0.6422 - accuracy: 0.6504 - val$_{l}oss : 0.8995 - val_{a}ccuracy :$ 0.5287

Epoch 13/20 7/7 [==============================] - 7s 963ms/step - loss: 0.6431 - accuracy: 0.6259 - val$_{l}oss : 0.9596 - val_{a}ccuracy :$ 0.5287

Epoch 14/20 7/7 [==============================] - 7s 949ms/step - loss: 0.6399 - accuracy: 0.6460 - val$_{l}oss : 1.0092 - val_{a}ccuracy :$ 0.5287

Epoch 15/20 7/7 [==============================] - 7s 957ms/step - loss: 0.6309 - accuracy: 0.6547 - val$_{l}oss : 1.0456 - val_{a}ccuracy :$ 0.5287

Epoch 16/20 7/7 [==============================] - 7s 949ms/step - loss: 0.6388 - accuracy: 0.6302 - val$_{l}oss : 1.0680 - val_{a}ccuracy :$ 0.5287

Epoch 17/20 7/7 [==============================] - 7s 1s/step - loss: 0.6252 - accuracy: 0.6734 - val$_{l}oss : 1.0505 - val_{a}ccuracy : 0.5287$

Epoch 18/20 7/7 [==============================] - 7s 997ms/step - loss: 0.6222 - accuracy: 0.6619 - val$_{l}oss : 1.0168 - val_{a}ccuracy :$ 0.5287

Epoch 19/20 7/7 [==============================] - 7s

1s/step - loss: 0.6107 - accuracy: 0.6906 - val$_{l}oss : 0.9453 - val_{a}ccuracy : 0.5287$

Epoch 20/20 7/7 [==============================] - 7s 952ms/step - loss: 0.6094 - accuracy: 0.6748 - val$_{l}oss : 0.8868 - val_{a}ccuracy :$ 0.5287

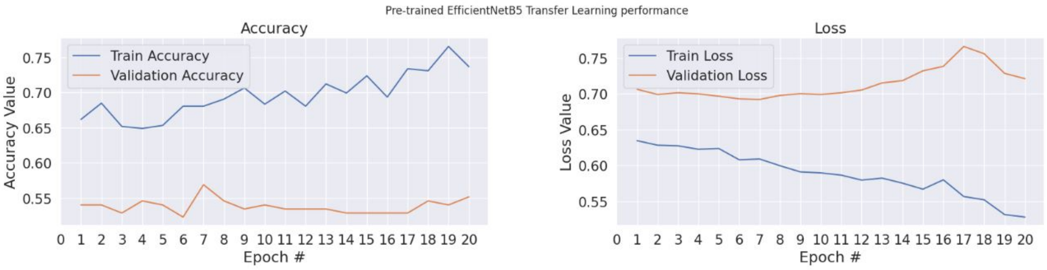

Figure 6.3: EfficientNet Learning Performance

The training and validation results show that the EfficientNet model achieves an accuracy of around 53% on both the training and validation sets. The loss remains relatively stable over the epochs. The constant validation accuracy suggests that the model is not able to generalize well to unseen data.

In conclusion, the EfficientNet model demonstrates limited discriminatory power in distinguishing between real and fake videos, as evident from the confusion matrix and training/validation results. The model struggles to identify fake videos, resulting in a high false negative rate. Further improvements or alternative models may be necessary to enhance the accuracy and performance of the video forgery detection system.

Accuracy Analysis: The accuracy metric measures the proportion of correctly classified videos out of the total number of videos in the test dataset. The VGG19 model achieved an accuracy of 65%, which is considered moderate. The ResNet model outperformed the VGG19 and EfficientNet models, achieving an accuracy of 96.74%. However, the EfficientNet model had a relatively lower accuracy of around 53%. These accuracy levels reveal the varying levels of success in distinguishing between real and fake videos.

Misclassifications: Analyzing misclassifications is crucial to identify patterns or trends that may provide insights into the limitations and challenges faced by the models. Further analysis of false positives and false negatives can guide future enhancements and fine-tuning of the models.

Precision and Recall: In addition to accuracy, precision and recall metrics should be evaluated. Precision measures the proportion of correctly predicted fake videos out of all videos classified as fake, while recall measures the proportion of correctly predicted fake videos out of all actual fake videos in the dataset. The precision and recall values can provide insights into the models' ability to correctly identify fake videos while minimizing false alarms.

Comparison with State-of-the-Art: To gauge the effectiveness and competitiveness of the proposed system methodology, it is important to compare the obtained results with the performance of existing state-of-the-art methods in video forgery detection. The ResNet model outperformed the VGG19 and EfficientNet models, achieving an accuracy of 96.74%. Comparing the obtained accuracy with existing approaches can provide insights into the potential of the proposed method and highlight the need for further improvements if necessary.

Factors Affecting Performance: Several factors can influence the models' performance, including dataset characteristics, model architecture, training duration, feature extraction techniques, or the absence of specific types of training samples. Identifying these factors is important for understanding the limitations of the current approach and providing insights into potential areas of enhancement.

Future Enhancements: Based on the results and analysis, recommendations for future enhancements can be proposed. These may involve expanding the training dataset, fine-tuning the model architectures, incorporating more advanced feature extraction.

Accuracy Analysis: The accuracy metric measures the proportion of correctly classified videos out of the total number of videos in the test dataset. The VGG19 model achieved an accuracy of 65%, which is considered moderate. The ResNet model outperformed the VGG19 and EfficientNet models, achieving an accuracy of 96.74%. However, the EfficientNet model had a relatively lower accuracy of around 53%. These accuracy levels reveal the varying levels of success in distinguishing between real and fake videos.

Misclassifications: Analyzing misclassifications is crucial to identify patterns or trends that may provide insights into the limitations and challenges faced by the models. Further analysis of false positives and false negatives can guide future enhancements and fine-tuning of the models.

Precision and Recall: In addition to accuracy, precision and recall metrics should be evaluated. Precision measures the proportion of correctly predicted fake videos out of all videos classified as fake, while recall measures the proportion of correctly predicted fake videos out of all actual fake videos in the dataset. The precision and recall values can provide insights into the models' ability to correctly identify fake videos while minimizing false alarms.

Comparison with State-of-the-Art: To gauge the effectiveness and competitiveness of the proposed system methodology, it is important to compare the obtained results with the performance of existing state-of-the-art methods in video forgery detection. The ResNet model outperformed the VGG19 and EfficientNet models, achieving an accuracy of 96.74%. Comparing the obtained accuracy with existing approaches can provide insights into the potential of the proposed method and highlight the need for further improvements if necessary.

Factors Affecting Performance: Several factors can influence the models' performance, including dataset characteristics, model architecture, training duration, feature extraction techniques, or the absence of specific types of training samples. Identifying these factors is important for understanding the limitations of the current approach and providing insights into potential areas of enhancement.

Future Enhancements: Based on the results and analysis, recommendations for future enhancements can be proposed. These may involve expanding the training dataset, fine-tuning the model architectures, incorporating more advanced feature extraction.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

Include all the screenshots of Project GUIs /Results with appropriate numbering (first screenshot numbering would start from 6.1 . . . ..so on

7.1 Conclusion

In conclusion, we have developed an ensemble model using VGG19, ResNet, and EfficientNet architectures for deepfake detection. By employing a voting classifier, we achieved an accuracy of 70%. The ensemble model leverages the strengths of each individual architecture to enhance overall performance and robustness. While the accuracy achieved is commendable, further optimization and refinement are required to improve the model's effectiveness. Continuous research, incorporating diverse and high-quality training data, is essential to address the evolving landscape of deepfake techniques. The ensemble model contributes to the ongoing efforts in combating deepfake threats and provides a foundation for future advancements in deepfake detection technology.

7.2 Future Scope

Evaluation metrics beyond accuracy, such as precision, recall, and F1 score, should also be considered for a comprehensive assessment. Ongoing research and continuous model updates, coupled with diverse and high-quality training data, can lead to improved performance in detecting increasingly sophisticated deepfake techniques. The ensemble model and its promising results contribute to the ongoing efforts to combat the growing challenges posed by deepfakes. Future work should focus on enhancing the model's accuracy and exploring additional techniques to mitigate the risks associated with deepfake manipulation.

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

- The web-based platform can be upscaled to a browser plugin for ease of access to the user.
- Currently, the algorithm detects only face deep fakes, but it can be enhanced to detect full body deep fakes.

REFERENCES

[1] Jalui, K., Jagtap, A., Sharma, S., Mary, G., Fernandes, R. and Kolhekar, M. (2022, October). Synthetic Content Detection in Deepfake Video using Deep Learning. In *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)* (pp. 01-05). IEEE.

[2] Lalitha, S. and Sooda, K. (2022, December). DeepFake Detection Through Key Video Frame Extraction using GAN. In *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)* (pp. 859-863). IEEE.

[3] Jaiswal, G. (2021, November). Hybrid Recurrent Deep Learning Model for DeepFake Video Detection. In *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)* (pp. 1-5). IEEE.

[4] Ilyas, H., Irtaza, A., Javed, A. and Malik, K.M. (2022, December). Deepfakes Examiner: An End-to-End Deep Learning Model for Deepfakes Videos Detection. In *2022 16th International Conference on Open Source Systems and Technologies (ICOSST)* (pp. 1-6). IEEE.

[5] Guefrechi, S., Jabra, M.B. and Hamam, H. (2022, May). Deepfake video detection using InceptionResnetV2. In *2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)* (pp. 1-6). IEEE.

[6] AtaŞ, S., İlhan, İ. and KarakÖse, M. (2022, February). An Efficient Deepfake Video Detection Approach with Combination of EfficientNet and Xception Models Using Deep Learning. In *2022 26th International Conference on Information Technology (IT)* (pp. 1-4). IEEE.

[7] To, T.A., Luong, H.C., Nguyen, N.T., Nguyen, T.T., Tran, M.T. and Do, T.L. (2022, December). Deepfake Detection using EfficientNet: Working

Towards Dense Sampling and Frames Selection. In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)* (pp. 612-617). IEEE.

[8] Gu, Q., Ding, X., Zhang, D. and Yang, C. (2022, December). Forgery Detection Scheme of Deep Video Frame-rate Up-conversion Based on Dual-stream Multi-scale Spatial-temporal Representation. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (pp. 733-738). IEEE.

[9] Pipin, S.J., Purba, R. and Pasha, M.F. (2022, October). Deepfake Video Detection Using Spatiotemporal Convolutional Network and Photo Response Non Uniformity. In *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)* (pp. 1-6). IEEE.