Deepfake Detection Using Deep Learning

A Project Report submitted to

Dr. Santosh Kumar Majhi

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

MASTER OF COMPUTER APPLICATIONS (MCA)

By

Neeraj Kumar Sahu

Roll No.: 23072133

Enrollment No.: GGV/23/05027

Under the Guidance of

Dr. Santosh Kumar Majhi

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.)

Session: 2023-2025

Deepfake Detection Using Deep Learning

A Project Report submitted to

Dr. Santosh Kumar Majhi

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

MASTER OF COMPUTER APPLICATIONS (MCA)

By

Neeraj Kumar Sahu

Roll No.: 23072133

Enrollment No.: GGV/23/05027

Under the Guidance of

Dr. Santosh Kumar Majhi

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR (C.G.)

Session: 2023-2025

CERTIFICATE OF SUPERVISOR(S) /GUIDE

This is to certify that the work incorporated in the project "**Deepfake Detection using Deep Learning**" is a record of _____month project work assigned by our Institution, successfully carried out by Neeraj Kumar Sahu bearing Enrollment No. GGV/23/05027 under my guidance and supervision for the award of Degree of Master of Computer Applications (MCA) of **DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY, GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR C.G., INDIA**. To the best of my knowledge and belief, the report embodies the work of the candidate herself and has duly been successfully completed.

Signature of the Supervisor/Guide

Name:

Signature of HOD

Name:

DECLARATION BY THE CANDIDATE

I. Neeraj Kumar Sahu, Student of IV Semester MCA**, DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY, GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR**, bearing Enrollment Number **GGV/23/05027** hereby declare that the project titled "**Deepfake Detection using Deep Learning**" has been carried out by me under the Guidance/Supervision of **Dr. Santosh Kumar Majhi,** Associate submitted in partial fulfillment of the requirements for the award of the Degree of Master of Computer Application (MCA) by the Department Of Computer Science & Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur during the academic year 2024-25 .This report has not been submitted to any other Organization/University for any award of Degree/Diploma**.**

(Signature Of Candidate)

Date:

Place:

ACKNOWLEDGEMENT

I have great pleasure in the submission of this project report entitled "**Deepfake Detection using Deep Learning**" **for Guru Ghasidas Vishwavidyalaya** in partial fulfilment of the degree of Master of Computer Applications. While submitting this Project report, I take this opportunity to thank those directly or indirectly related to the project work.

I would like to thank my guide **Dr. Santosh Kumar Majhi**, who has provided valuable guidance and support throughout the project. Without his active co-operation and continuous encouragement, it would have become very difficult to complete this task in time.

I would also like to express my sincere thanks to **Dr. Ratnesh Prasad Srivastava** Head of the Department of Computer Science & Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, for providing all the necessary facilities and a supportive environment.

Acknowledgement is also due to my parents, family members, friends and all those who supported me directly or indirectly in the successful completion of this project work.

(Neeraj Kumar Sahu)

CERTIFICATE BY THE EXAMINER

This is to certified that the project work entitled "**Deepfake Detection using Deep Learning**" submitted by **Neeraj Kumar Sahu** has completed under the supervision of **Dr. Santosh Kumar Majhi** Department of "Computer Science & Information Technology", GGU Bilaspur (C.G) has been examined by the undersigned as a part of examination for the award of Master Of Computer Applications (MCA) Degree in Department of "**Computer Science & Information Technology" in GURU GHASIDAS CENTRAL UNIVERSITY BILASPUR (C.G).**

"Project Examined & Approved"

…… **……**

Internal Examiner **External Examiner**

Date: **Date:**

Signature of H.O.D.(CSIT)

GURU GHASIDAS CENTRAL UNIVERSITY BILASPUR (C.G)

Table Of Content

List Of Figures:

1. Abstract

With the rapid advancement in deep learning and generative models, the creation of highly realistic synthetic media—commonly known as deepfakes—has become increasingly accessible. While these technologies offer promising applications in fields such as entertainment, education, and gaming, they simultaneously pose severe risks, including misinformation, political manipulation, identity theft, and cyber harassment. To address these challenges, this project proposes a deep learning-based approach for detecting deepfake video content by leveraging the efficiency and lightweight design of the MobileNetV2 architecture.

The model is developed using a transfer learning strategy, followed by fine-tuning to enhance detection accuracy on frame-level video inputs. We employ the FaceForensics++ dataset, a widely recognized benchmark for deepfake detection, which includes a balanced collection of authentic and manipulated videos. Each video is decomposed into frames, resized to 224x224 pixels, and normalized to standardize input for the neural network. The dataset is then partitioned into training (70%), validation (15%), and test (15%) sets to ensure unbiased evaluation.

To evaluate model performance, we apply a comprehensive set of metrics, including accuracy, precision, recall, F1-score, and a confusion matrix. After training over 10 epochs with early stopping and data augmentation. Moreover, the use of MobileNetV2 ensures that the detection system remains computationally efficient, making it suitable for real-time or edge-device deployment in digital forensics and content verification platforms.

This project illustrates how AI-powered systems, when responsibly designed, can be effective countermeasures against the misuse of AI-generated synthetic media, and provides a scalable framework for future enhancements in the field of multimedia forensics.

Keywords

- Deepfake Detection
- MobileNetV2
- Transfer Learning
- FaceForensics++
- Model Evaluation
- Synthetic Media

2. Synopsis

Deepfake technology involves the synthesis of hyper-realistic human images and videos using advanced deep learning techniques, most notably Generative Adversarial Networks (GANs) and Autoencoders. These techniques are capable of replacing or manipulating facial features and entire faces in video sequences, often producing outputs that are visually indistinguishable from genuine footage. While the technology has transformative potential in industries like film production, virtual reality, and education, its misuse raises significant concerns. Deepfakes have been used for spreading disinformation, manipulating political narratives, committing identity fraud, and perpetrating various forms of cyber harassment.

In response to these growing threats, this project proposes a deep learning-based detection framework aimed at identifying manipulated media. Rather than depending on manual detection, which is often unreliable and subjective, our method leverages the subtle artifacts and inconsistencies unintentionally introduced during deepfake generation. These imperfections—though often imperceptible to the human eye—can be detected by deep convolutional neural networks (CNNs), which are adept at learning hierarchical patterns in image data.

The proposed model utilizes the MobileNetV2 architecture for its computational efficiency, lightweight structure, and strong performance on visual recognition tasks. This architecture is fine-tuned on frame-level image data extracted from the FaceForensics++ dataset, one of the most comprehensive and diverse datasets for deepfake detection research. Preprocessing steps such as frame extraction, resizing to 224x224 pixels, normalization, and dataset splitting are meticulously performed to ensure high-quality input to the neural network.

The overarching objective is to develop a model that is not only accurate but also efficient enough for real-time deployment scenarios, including mobile devices or edge-based surveillance systems. Model evaluation is conducted using multiple performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix to ensure its robustness and generalization across unseen samples. The success of this approach underlines the role of AI in developing defensive strategies against the malicious use of AI-generated media.

2.1 Area of the Project

This project is situated at the intersection of Deep Learning and Computer Vision, both of which are crucial subfields within Artificial Intelligence (AI). Deep Learning involves training multi-layered neural networks to identify patterns and features in data, mimicking human cognitive functions. Within this framework, Computer Vision plays a pivotal role by enabling machines to interpret and analyze visual inputs like images and video frames.

Specifically, this project employs OpenCV for efficient video frame extraction and manipulation, preparing the data pipeline for training a CNN-based deepfake detection model. The selected model architecture, MobileNetV2, is well-suited for deployment in environments with limited computational resources due to its streamlined design and lower parameter count.

The entire project is implemented using TensorFlow and Keras frameworks within the Google Colab environment, allowing for scalable model training using GPU acceleration. The combination of practical tools and theoretical approaches highlights the synergy between applied machine learning and real-world challenges, making this project a compelling example of how technology can be harnessed responsibly to combat misuse.

2.2 Technical Keywords (with Brief Descriptions)

1. Deep Learning
   A subset of machine learning that uses neural networks with multiple layers to learn and represent data features automatically, especially effective in pattern recognition tasks like image classification.

2. Computer Vision
   The field of AI focused on enabling machines to interpret and make decisions based on visual inputs such as images and video frames.

3. MobileNetV2 Architecture
   A lightweight convolutional neural network optimized for mobile and embedded vision applications. It is used in this project for efficient deepfake classification.

4. Convolutional Neural Networks (CNNs)
   Deep learning models specially designed for analyzing visual data. CNNs extract spatial features from image pixels through convolution operations.

5. OpenCV
   An open-source computer vision library used for real-time image and video processing tasks, such as frame extraction and face cropping in this project.

6. Face Detection

   A computer vision technique used to identify and locate human faces in images or video frames, often a preprocessing step in deepfake detection.

7. Generative Adversarial Networks (GANs)

   A class of machine learning frameworks used to generate realistic synthetic data. GANs are commonly employed in deepfake creation.

8. TensorFlow / Keras

   Popular deep learning libraries used to build and train neural networks. In this project, they are used for model development and experimentation.

9. Transfer Learning

   A technique where a pre-trained model is adapted to a new but similar task, reducing training time and improving performance, especially when working with limited data.

10. Image Preprocessing

    The steps applied to raw image data to prepare it for training, including resizing, normalization, and augmentation.

11. Model Evaluation Metrics

    Quantitative measures such as accuracy, precision, recall, F1-score, and confusion matrix used to evaluate the performance of the trained model.

3. Introduction

3.1 Project Idea

In the modern digital era, the widespread use of social media platforms and rapid advancements in Artificial Intelligence (AI) have led to the emergence of deepfakes as a major technological and ethical threat. Deepfakes refer to the manipulation of videos or images using sophisticated AI techniques to create highly realistic but entirely fabricated content. Such synthetic media can be maliciously used to spread misinformation, incite political unrest, commit identity fraud, and cause serious ethical violations including revenge porn and impersonation.

The creation of deepfakes typically involves Generative Adversarial Networks (GANs), Autoencoders, and tools like FaceSwap or FaceApp, which are capable of superimposing one person's face onto another's body or altering facial expressions with astonishing realism. Due to their visual accuracy, it becomes extremely difficult for the human eye to distinguish between authentic and manipulated content.

To combat this growing concern, this project proposes a deep learning-based deepfake detection system that uses AI to fight AI. The proposed system uses a MobileNetV2-based Convolutional Neural Network (CNN) for frame-level spatial feature extraction. To model the temporal dependencies across video frames, we incorporate a Long Short-Term Memory (LSTM) network, which enables the classification of a video as either real or deepfake.

To enhance the robustness and generalization ability of the model, we train it on a diverse and balanced dataset that includes publicly available deepfake datasets such as:

- FaceForensics++
- Celeb-DF

Additionally, a lightweight front-end web application has been developed that allows users to upload videos for analysis. The system processes the input video, classifies it as either "Real" or "Deepfake," and displays the result along with a confidence score, providing a user-friendly interface suitable for real-time deployment and practical use.

3.2 Motivation of the Project

The democratization of mobile video technology and the explosive growth of video-sharing platforms have made it significantly easier to create and distribute video content. Simultaneously, the evolution of deep learning has enabled the development of powerful generative models capable of producing hyper-realistic media, including fake faces, speech, and even full-body animations.

While these models have various beneficial applications—such as aiding the visually impaired, supporting medical diagnostics, or enhancing visual effects in media—their misuse has raised serious ethical and societal challenges. Among these, deepfakes represent one of the most alarming threats. Since 2017, the rise of open-source deepfake generation tools has made it easier than ever to create manipulated content with minimal expertise.

Although some deepfakes are intended for entertainment or satire, many have been weaponized for:

- Misinformation and propaganda
- Political manipulation
- Defamation and cyberbullying
- Revenge porn and identity theft

For example, a fake video of a world leader announcing military action or a celebrity making offensive remarks can trigger mass panic, election interference, or irreversible reputational damage. As such, the detection of deepfakes has become a critical need in the digital information ecosystem.

The primary motivation for this project is to contribute toward safeguarding digital authenticity by providing a reliable, efficient, and scalable solution to detect deepfakes. By combining the computational efficiency of MobileNetV2 with the temporal sequence learning capabilities of LSTM networks, this system aims to accurately classify manipulated media.

Furthermore, the project aspires to empower individuals, media platforms, and government agencies with a tool that enhances public trust, mitigates digital misinformation, and upholds the ethical use of artificial intelligence.

4. Literature Survey

In recent years, numerous deep learning-based techniques have been proposed to detect and classify deepfake videos. These studies have enhanced our understanding of the vulnerabilities inherent in current deepfake generation techniques and have driven the development of increasingly sophisticated detection methods. However, many of these approaches face challenges in terms of scalability, generalization to real-time data, or robustness across varied datasets. Below is a concise survey of significant contributions in this field.

4.1 Face Warping Artifacts Detection

The study on Face Warping Artifacts [15] introduced a technique to detect deepfakes by analyzing inconsistencies between synthesized facial regions and their neighboring areas using a Convolutional Neural Network (CNN). This method exploited the common limitation in deepfake generation, where synthetic faces are created at a limited resolution and then upscaled or warped to align with target facial structures. While effective against basic deepfakes, this method did not account for temporal dynamics within video sequences, leaving it vulnerable to high-quality, temporally consistent manipulations.

4.2 Detection Using Eye Blinking Patterns

The Eye Blinking Detection approach [16] leveraged a Long-term Recurrent Convolutional Network (LRCN) to analyze temporal eye-blinking behavior, a subtle physiological cue often missing in early deepfake videos. The absence of natural blinking patterns provided a basis for classification. However, with the advent of more sophisticated deepfake generators that simulate blinking and other facial movements, relying solely on such isolated features has become insufficient. Effective detection now requires a broader range of temporal and spatial cues.

4.3 Capsule Networks for Deepfake Detection

This method explored the use of Capsule Networks [17] to detect manipulated content by modeling spatial hierarchies and pose relationships. Capsules aim to preserve part-whole relationships in images, which can help in identifying structural inconsistencies in deepfakes. However, the study introduced artificial noise to improve generalization, which may degrade model performance when applied to clean, real-world data. Additionally, the lack of explicit temporal modeling limits its effectiveness in analyzing video sequences.

4.4 Recurrent Neural Networks for Frame Analysis

Another approach utilized Recurrent Neural Networks (RNNs) [18] for temporal sequence modeling based on frame-level features extracted using an ImageNet-pretrained CNN. While the idea of sequential processing was promising, the model was trained on the HOHO dataset [19], which contained only 600 visually similar video clips. The limited dataset resulted in poor generalization and scalability, particularly when tested against diverse or high-resolution deepfake samples from modern datasets.

4.5 Summary and Proposed Improvement

The reviewed methods often suffer from one or more limitations—such as reliance on narrow feature cues, limited training datasets, or the absence of robust temporal modeling. To overcome these challenges, our project introduces a hybrid approach combining MobileNetV2 (a lightweight yet effective CNN) for efficient spatial feature extraction for temporal sequence analysis. This architecture is trained on a diverse dataset comprising real and fake videos from FaceForensics++ and Celeb-DF, resulting in improved generalization and real-world applicability. Furthermore, our system is integrated into a user-friendly application interface, enabling practical use in real-time scenarios.

5. Problem Definition and Scope

5.1 Problem Statement

With the rapid advancements in deep learning, particularly in generative models such as Generative Adversarial Networks (GANs) and Autoencoders, the creation of hyper-realistic fake videos—commonly known as deepfakes—has become increasingly accessible and sophisticated. These AI-driven technologies allow virtually anyone to fabricate synthetic media that is nearly indistinguishable from authentic content, posing significant ethical, legal, and security concerns.

While video manipulation and visual effects have existed for decades, modern deepfake techniques are powered by AI algorithms that make forgery not only faster and cheaper but also more difficult to detect. The democratization of these tools has led to their use in harmful contexts including political misinformation, celebrity impersonation, cyberbullying, identity theft, financial scams, and revenge porn. The viral nature of social media platforms amplifies the spread and impact of such manipulated media, making timely detection critically important.

Despite the simplicity of generating deepfakes, accurate and reliable detection remains a complex challenge. Traditional detection methods that rely on human visual cues or basic digital forensics are insufficient due to the subtlety and precision of modern forgeries. This calls for the development of intelligent, automated systems capable of identifying manipulation at scale.

This project directly addresses this challenge by proposing a robust, AI-powered deepfake detection framework. We leverage MobileNetV2 as a lightweight yet effective deep feature extractor. This hybrid approach enables the system to not only analyze individual frames but also model the continuity and contextual coherence across them—significantly enhancing detection performance.

5.1.1 Goals and Objectives

The primary goals and objectives of this project are:

- To accurately detect deepfake videos using a deep learning approach involving MobileNetV2 for spatial features.
- To help mitigate the spread of deceptive or harmful content across digital platforms, contributing to a more trustworthy online ecosystem.
- To distinguish between authentic (pristine) and AI-manipulated (deepfake) videos, enhancing digital content verification.
- To provide a user-friendly interface through which end-users can upload video files and receive instant classification results, supported by a confidence score that reflects the model's certainty.

The overall mission of this project is to combat AI-generated media threats by leveraging advanced AI tools responsibly and ethically to protect users and digital platforms.

5.1.2 Statement of Scope

In today's digital landscape, the tools for creating deepfakes are becoming ubiquitous, yet the tools for detecting them are scarce and often ineffective. This project aims to bridge that gap by delivering a reliable, real-time, and scalable web-based platform capable of classifying videos as "real" or "deepfake" with high precision.

The system is built around a two-stage pipeline:

1. Frame extraction from input videos.
2. Deep feature extraction using a fine-tuned MobileNetV2 model.

The platform supports video uploads directly from the browser and returns a binary classification (Real or Deepfake) along with a confidence score. The lightweight architecture ensures efficient processing, even on moderate hardware. Moreover, the framework can be extended to:

- Browser extensions for auto-detection during video playback.
- Real-time verification for video conferencing tools (e.g., Zoom, Teams).
- API integrations for social media platforms like WhatsApp, Instagram, and Facebook to enable pre-upload verification.

5.1.3 Software Description and Functional Scope

Input Specifications:

- Input Type: Video files (e.g., .mp4)
- Size Constraints: Recommended maximum file size is ~20 MB for optimal performance.
- Input Validation: The system validates file format, size, and checks for file corruption before processing.
- Platform Requirements: Requires a supported web browser and stable internet connection (for web-hosted versions).

Output Specifications:

- Classification Result: Indicates whether the video is a Deepfake or Real.
- Confidence Score: Displays the model's certainty in the prediction (e.g., 92.7% confidence).
- Processing Time: Dependent on file size and server load, typically within 5–30 seconds.

5.1.3 System Expansion Possibilities:

- Integration of browser-based extensions for seamless on-the-fly detection.
- Addition of real-time detection modules in video calling apps.
- Development of REST APIs for automated verification in content moderation systems or messaging platforms.

5.1.4 Conceptual State Diagram / I/O Flow:

1. Input: User uploads a video file through the web interface.
2. Processing Pipeline:
    - Frame Extraction: Frames are sampled from the video.
    - Feature Extraction: Each frame is passed through the MobileNetV2 model.
    - Temporal Analysis & Classification.
3. Output:
    - Classification Label: Deepfake or Real.
    - Confidence Score: Percentage value indicating prediction certainty.

5.2 Major Constraints

- User Interaction:

  The system offers a simple upload interface for video submission. Once uploaded, deepfake detection begins automatically, using a pre-trained MobileNetV2 model.

- Prediction Display:

  The application processes each frame and highlights predictions directly on faces, showing "Fake" or "Real" with the associated confidence score for each detected face.

- User-Friendly Interface:

  The web interface, built using Flask, provides a clean and intuitive experience. Users can interact with the tool without needing any technical background, making the solution accessible to a wide audience.

- Cross-Platform Compatibility:

  Since it is built on standard web technologies, the application can be accessed on various platforms—Windows, macOS, Android, and iOS—using any modern browser.

6. Methodologies of Problem Solving

6.1 Analysis

1. **Solution Requirement:**

   The problem was analyzed extensively to assess the feasibility of developing a deepfake detection system. Existing methods were reviewed to identify best practices and limitations. Early experiments using unbalanced datasets (containing only real or only fake videos) led to biased results and poor model generalization. This insight emphasized the need for balanced datasets—an equal distribution of real and fake videos—to enhance model accuracy and reduce bias and variance.

2. **Solution Constraints:**

   Several constraints were considered during system design and development, including:

   - Cost-efficiency: Ensuring the system remains affordable for development and deployment.
   - Processing Speed & Latency: The system must process videos in a reasonable timeframe.
   - Hardware & Software Requirements: Compatibility with standard devices and platforms.
   - User Accessibility: Minimizing the need for technical expertise.
   - Resource Availability: Access to GPUs and scalable cloud resources for training and inference.

3. **Parameters Identified for DeepFake Detection**

   Key visual anomalies used to guide training and model focus included:

   1. Abnormal blinking patterns
   2. Irregularities in teeth shapes
   3. Unnatural eye distances
   4. Inconsistent or unrealistic moustaches
   5. Double edge effects around facial boundaries
   6. Artifacts in iris segmentation
   7. Lack of natural wrinkles
   8. Disoriented face angles
   9. Uneven skin tone
   10. Unnatural facial expressions
   11. Illogical lighting and shading

12. Awkward pose transitions
13. Inconsistent presence of double chins
14. Flickering or odd hairstyles
15. Exaggerated cheekbone structures
16. Inconsistent head poses

6.2 Design

The architecture was designed using MobileNetV2 due to its lightweight, efficient structure for image feature extraction. Instead of full-video analysis, the system analyzes individual frames to balance detection quality with computational efficiency. Temporal relationships were incorporated using sequential frame processing, allowing the model to capture changes across time while avoiding the overhead of full video modeling.

6.3 Development

Python was used for development. Although PyTorch was considered, TensorFlow/Keras was selected for better web integration. The MobileNetV2 model was trained on a balanced dataset of real and fake videos. Training was conducted on Google Cloud Platform (GCP), utilizing GPU acceleration for improved performance and faster training times.

6.4 Evaluation

The model was evaluated using a comprehensive and diverse dataset. The primary evaluation tool was the confusion matrix, which provided insights into:

- True Positives (TP): Real videos correctly identified
- True Negatives (TN): Fake videos correctly identified
- False Positives (FP): Fake videos misclassified as real
- False Negatives (FN): Real videos misclassified as fake

Additional evaluation metrics included:

- Accuracy
- Precision
- Recall
- F1-score

6.5 Outcome

The final result is a trained deepfake detection model integrated into a Flask-based web application. Users can upload video files, which are then processed by the backend. The system extracts frames, performs inference using MobileNetV2, and displays annotated frames with confidence scores indicating whether each face is real or fake.

6.6 Applications

The project has been deployed as a web-based application, enabling cross-platform usage. Its user-friendly interface allows easy video uploads and returns detection results in an accessible format. Potential applications include:

- Fact-checking tools
- Social media content moderation
- Law enforcement digital forensics
- Media and journalism verification pipelines

6.7 Hardware Resources Required

Server-Side Requirements:

- GPU-enabled environment (Google Colab Pro, GCP, or local machine with GPU)
- Sufficient storage for video datasets and trained models

Client-Side Requirements:

- Any modern web browser (Chrome, Firefox, Safari, Edge)
- Desktop or laptop capable of uploading video files

6.8 Software Resources Required

- Operating System: Windows 7 or later
- Programming Language: Python 3.0+
- Frameworks: TensorFlow, Flask
- Cloud Platform: Google Colab
- Libraries:
  - OpenCV (for frame processing)
  - Face-recognition (for face detection and alignment)

7. Software Requirements Specification (SRS)

7.1.1 Purpose and Scope of Document

This Software Requirements Specification (SRS) outlines the development of a Web-Based Deepfake Video Detection System using deep learning models, primarily convolutional and recurrent neural networks. The system aims to assist users in determining the authenticity of video content by analyzing frames and detecting deepfake manipulations.

This document covers:

- Functional and non-functional requirements
- Use case modeling and data flow diagrams
- System architecture and component interactions
- Identified risks and mitigation strategies
- Evaluation metrics for performance and accuracy

7.1.2 Use Case View

The system provides an interactive interface through which users can upload videos and receive deepfake analysis results. Below is the summarized flow of interactions:

- User uploads a video via the web application
- System extracts frame's and processes them
- Deep learning model classifies each frame as real or fake
- Predictions with confidence scores are overlaid on the video and displayed

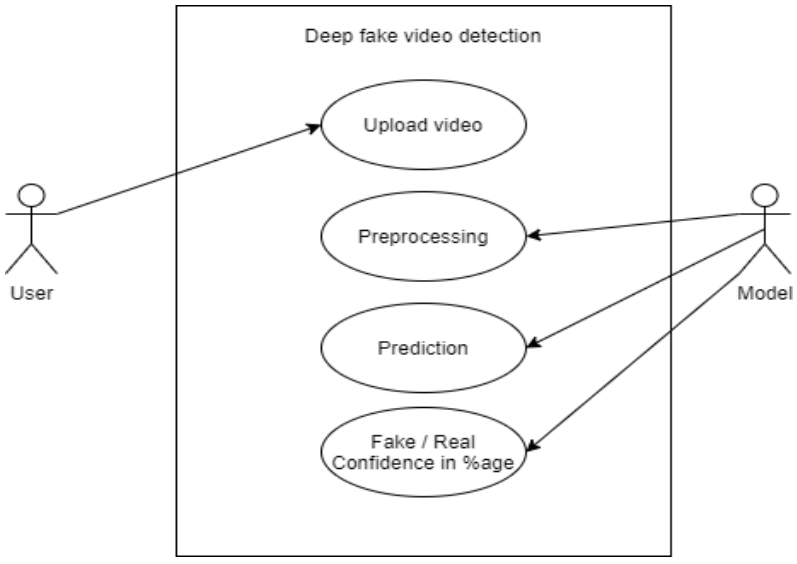

Figure 7.1 Use Case Diagram

7.2 Functional Model and Description

7.2.1 Data Flow Diagrams

DFD Level 0 – System Overview

- Input: User uploads a video
- Process: System performs frame extraction, face detection, and classification
- Output: Final output includes prediction (Real/Fake) with confidence scores

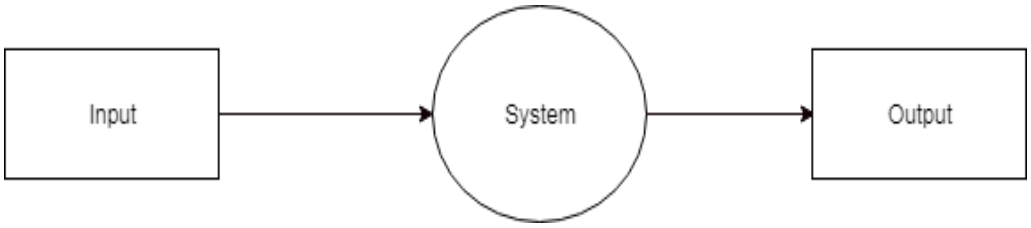

Figure 7.2 DFD Level 0

DFD Level 1 – Process Breakdown

Detailed description of each subprocess including:

- Frame extraction
- Preprocessing and face detection
- Deepfake classification
- Result rendering and display

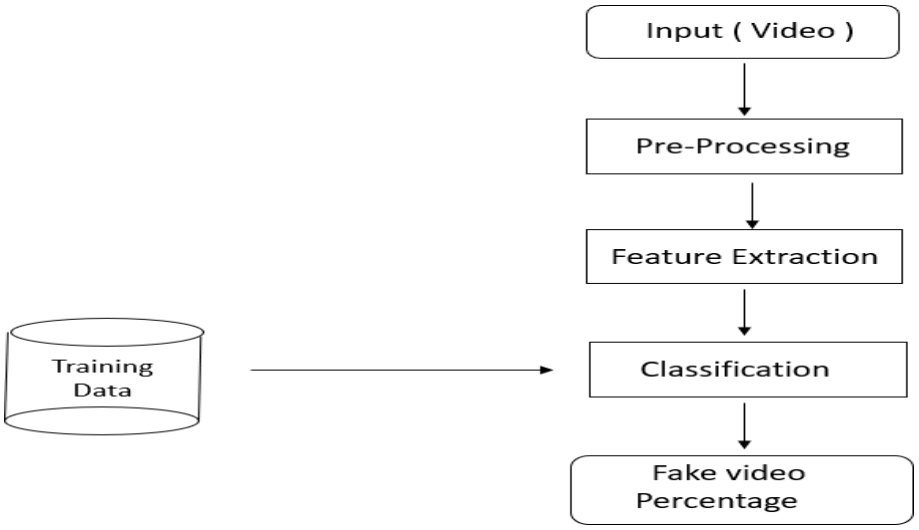

Figure 7.3 DFD Level 1

7.2.2 Activity Diagrams

Training Workflow

Describes the steps followed to train the model:

- Load dataset
- Extract frames
- Detect and crop faces
- Train deep learning model (MobileNetV2)
- Save the model for inference

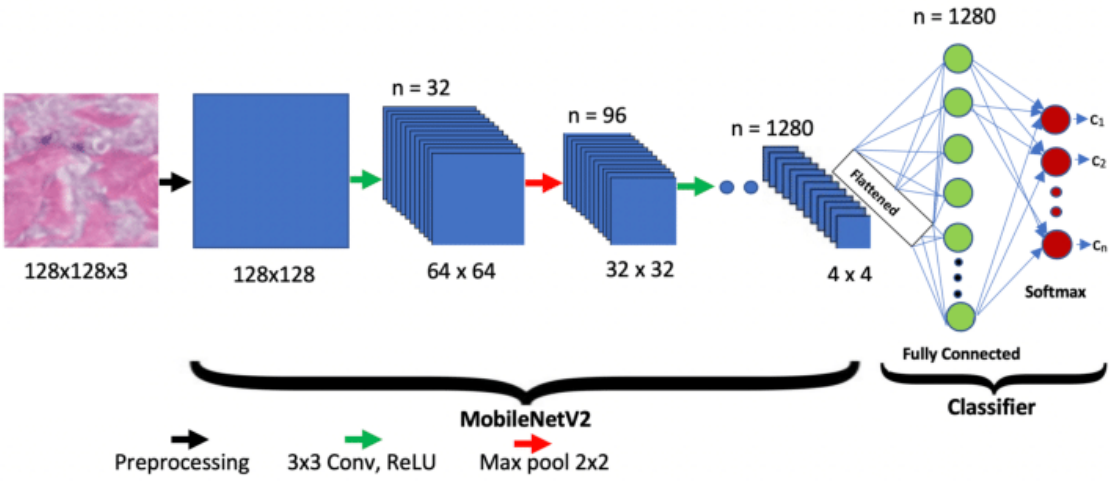

Figure 7.4 Training Workflow

Testing Workflow

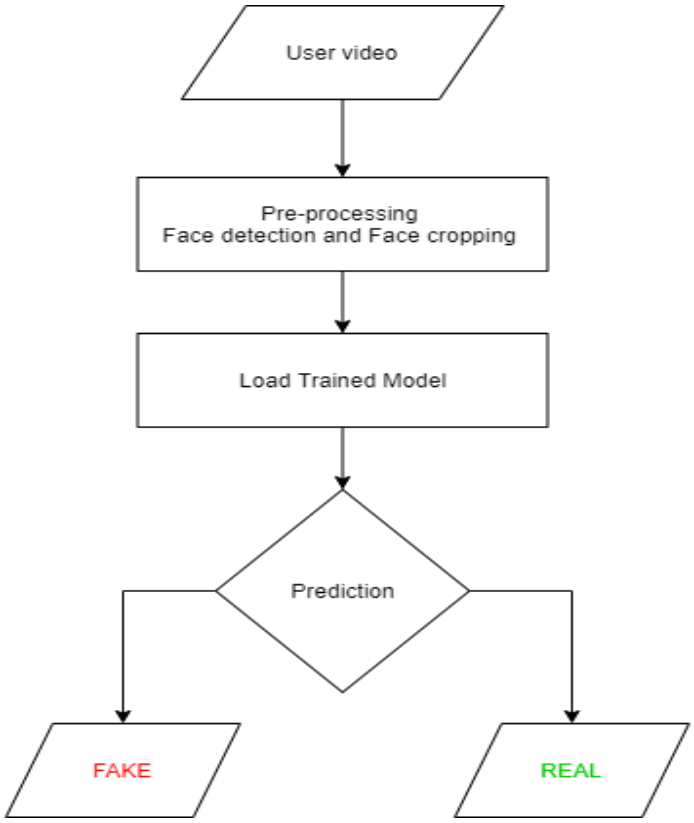

Figure 7.5 Testing Workflow

Describes the real-time testing pipeline:

- User uploads video
- Frame extraction and preprocessing
- Pass frames to trained model
- Get predictions and confidence
- Display results on web interface

7.2.3 Non-Functional Requirements

Safety Requirements:

- Uploaded videos are not stored permanently
- Temporary files are automatically deleted after 30 minutes
- Entire process is automated without human interaction

Security Requirements:

- Encryption of video files during upload and processing
- SSL protocols enabled for secure communication
- Internal-only decryption during processing
- Ensures user privacy and content integrity

7.2.4 Sequence Diagram

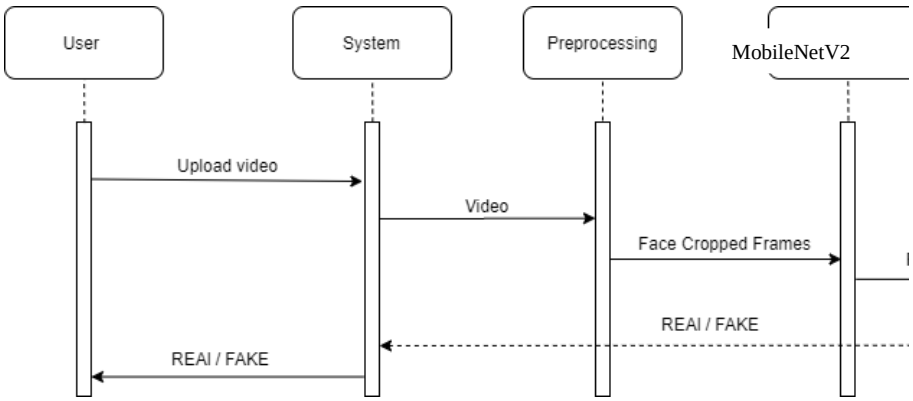

Figure 7.6 Sequence Diagram

The sequence diagram illustrates the complete interaction flow between user and system components:

Actors and Actions:

- User: Uploads video and views results
- Frontend: Sends video to backend API
- Backend Server: Handles extraction, preprocessing, and deepfake classification
- Renderer: Displays the processed video with predictions

8. Detailed Design Document

8.1 System Architecture

The Deepfake Detection System follows a multi-stage deep learning pipeline, built for effective video-based classification. The architecture is modular, enabling easy updates and scalability. It includes data preprocessing, model training, and real-time prediction via a web-based interface.

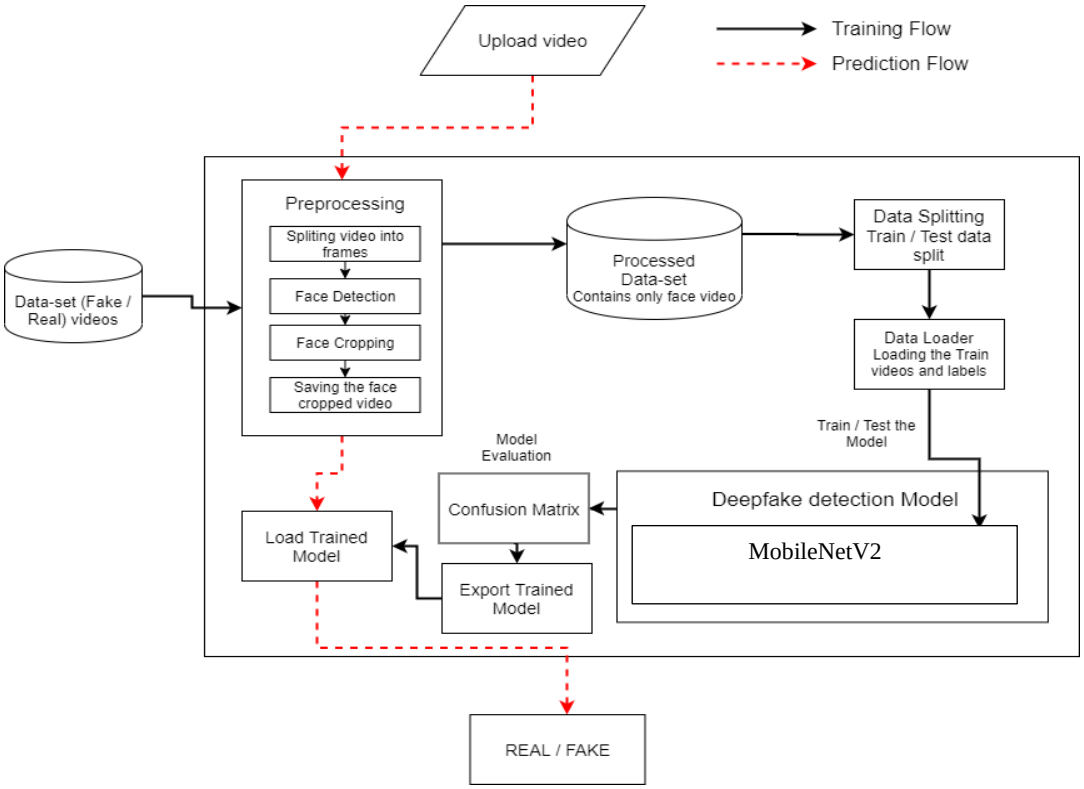

Figure 8.1 System Architecture

Key Stages of the System:

- Dataset Collection & Preprocessing

  Facial regions are detected and cropped from video frames to focus on relevant areas, improving model efficiency and accuracy.

- Model Training

  A lightweight and efficient convolutional neural network, MobileNetV2, is used for binary classification of real vs fake videos.

- Web Application Deployment

  The trained model is integrated into a Flask-based web interface to allow users to upload videos and receive deepfake predictions in real time.

Creating deepfake videos

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

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

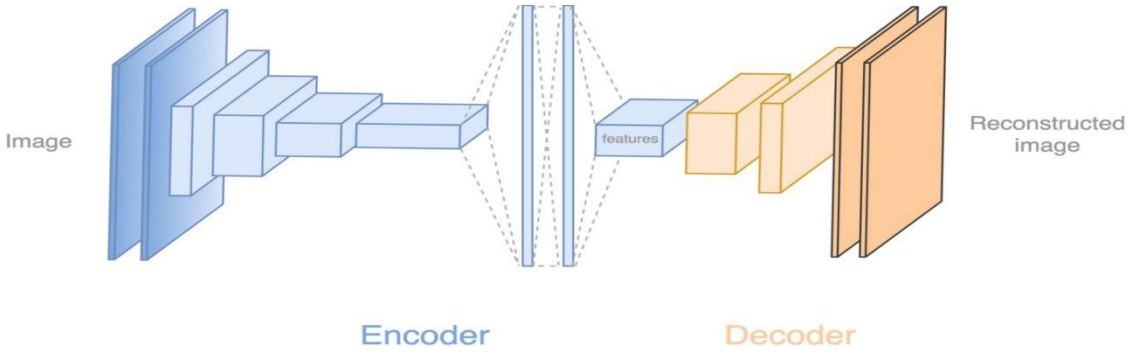

Figure 8.2: Deepfake generation

Figure 8.3: Face Swapped deepfake generation

Tools for deep fake creation.

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

8.2 Architectural Design

8.2.1 Module 1: Dataset Gathering

To ensure high generalization across diverse content, a hybrid dataset was created using:

- FaceForensics++: 200 real, 200 fake videos
- Celeb-DF: 500 real, 500 fake videos

8.2.2 Module 2: Preprocessing

Each video goes through the following preprocessing steps:

- Frame Extraction: Convert video into individual frames
- Face Detection & Cropping: Detect and crop facial regions from each frame
- Video Reconstruction: Rebuild videos using only the cropped facial regions
- Frame resolution: 224x224 pixels
- Frames without faces are discarded for accuracy

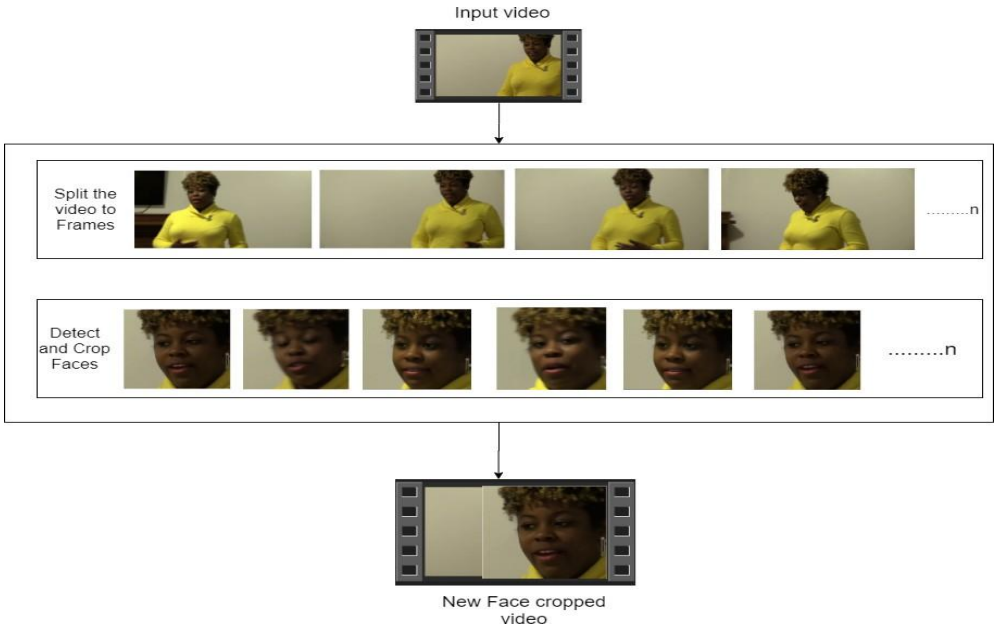

Figure 8.4 Preprocessing Workflow

8.2.3 Module 3: Dataset Splitting

The dataset is split into training, testing and validation sets:

- Training Set
- Testing Set
- Validation Set

8.2.4 Module 4: Model Architecture

The deepfake classifier consists of:

- Base Model: MobileNetV2, selected for its low memory footprint and high accuracy
- Transfer Learning Strategy:
    - Initially, the base layers are frozen, and only the top layers are trained
    - Later, the top 30 layers are unfrozen for fine-tuning, enhancing feature learning from facial artifacts

8.2.5 Module 5: Hyperparameter Tuning

The following hyperparameters were optimized after multiple trials:

- Optimizer: Adam
- Learning Rate: 1e-5
- Loss Function: Binary Cross-entropy
- Batch Size: 32

8.3 Web Interface

The system includes a Flask-based web application that provides a user-friendly interface for real-time deepfake prediction.

Key Components:

- Index Page (index.html)
  - Allows the user to upload a video
  - Upload form is styled for clarity and usability

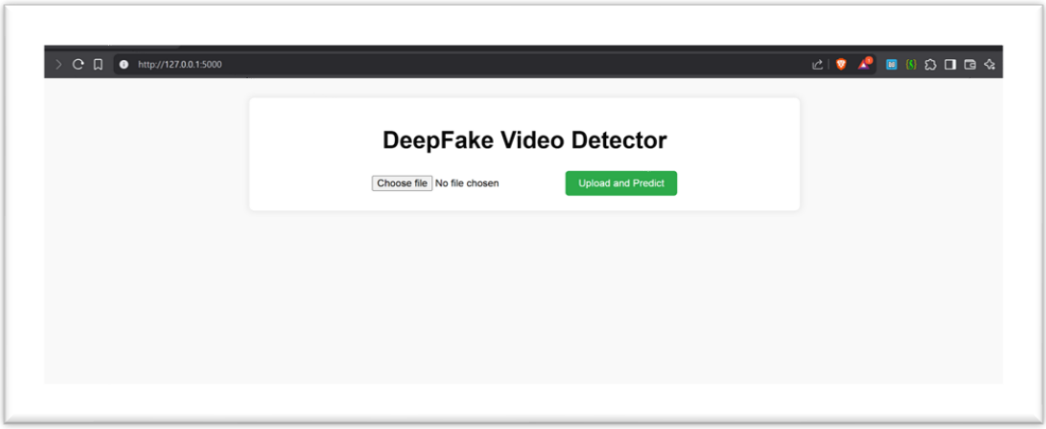

Figure 8.5 Index Page of Web App

- Prediction Page (predict.html)
  - Displays whether the uploaded video is Real or Fake
  - Shows the confidence score.

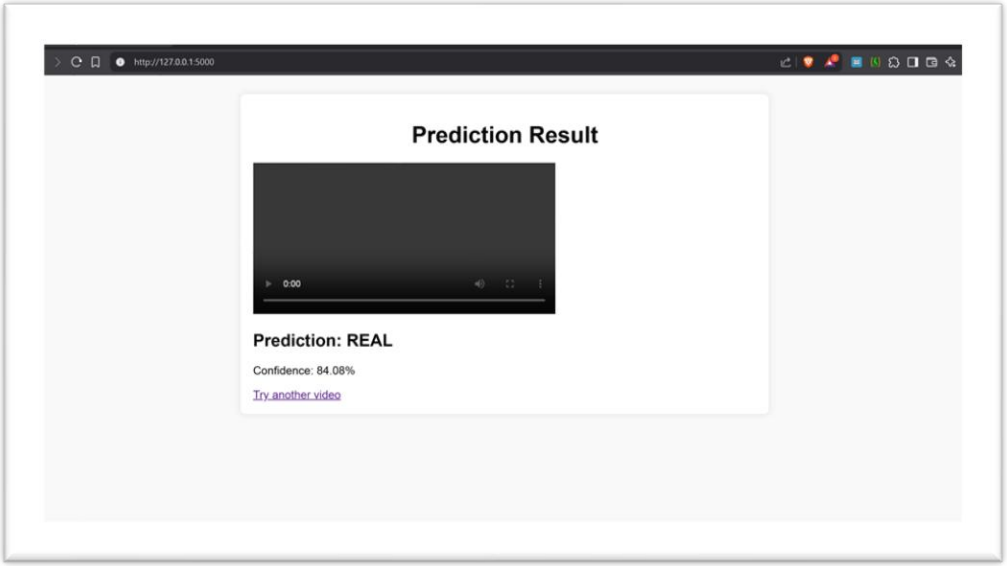

Figure 8.6 Predict Page of Web App

9. Performance Metrics Analysis

Evaluating the effectiveness of a DeepFake detection model requires more than just monitoring accuracy. In high-stakes applications such as detecting synthetic media, it is critical to assess how well the model performs in differentiating between real and manipulated content across a range of metrics. This section presents a detailed analysis of the model's performance using widely adopted evaluation metrics, namely: precision, recall, F1-score, and confusion matrix. These metrics provide a nuanced understanding of the classification performance, especially under class imbalance and real-world variability.

9.1 Confusion Matrix

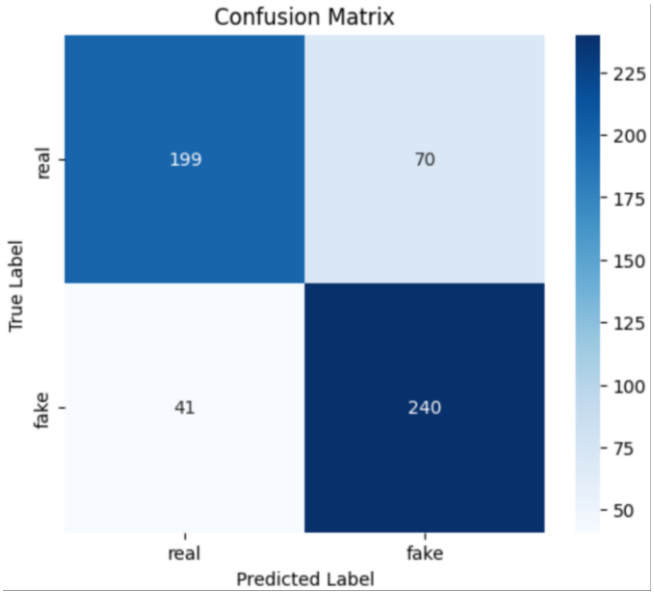

Figure 9.1 Confusion Matrix

A confusion matrix visually represents the number of correct and incorrect predictions made by the model. In a binary classification task such as DeepFake detection, the confusion matrix consists of four key elements:

- True Positives (TP): DeepFakes correctly identified as DeepFakes
- True Negatives (TN): Real videos correctly identified as Real
- False Positives (FP): Real videos wrongly classified as DeepFakes
- False Negatives (FN): DeepFakes wrongly classified as Real

An ideal model would maximize TP and TN while minimizing FP and FN. For our model, the confusion matrix revealed a high number of true positives and true negatives, indicating a strong ability to detect both real and fake content correctly. However, the presence of some false negatives highlights the challenge of subtle manipulations that closely resemble authentic footage.

9.2 Precision

Precision measures the proportion of predicted DeepFakes that are actually DeepFakes:

$$Precision = \frac{TP}{TP + FP}$$

A high precision indicates a low false positive rate, meaning the model does not incorrectly label many real videos as fake. This is crucial in real-world applications where wrongly flagging authentic content can have reputational or legal consequences. In our case, the model achieved a precision of approximately 80%, signifying reliable prediction of DeepFakes with minimal false alarms.

9.3 Recall

Recall (or sensitivity) measures the ability of the model to detect all actual DeepFakes:

$$Recall = \frac{TP}{TP + FN}$$

A high recall value ensures that very few DeepFakes go undetected. For our model, the recall score was around 80%, indicating that the system could successfully identify the majority of manipulated content, even in challenging scenarios with occlusion or low resolution.

9.4 F1-Score

The F1-score provides a harmonic mean of precision and recall:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

It balances the trade-off between precision and recall. The F1-score for our model was 80%, which affirms the model's robustness in simultaneously achieving high recall and precision. This is particularly valuable when both false positives and false negatives carry significant consequences.

9.5 Class-wise Metrics

Given the possible imbalance in real and DeepFake samples, macro-averaged and weighted-averaged scores were also calculated. The model showed slightly higher performance on the "Real" class, which is expected due to the higher visual consistency in authentic videos. However, performance on the "DeepFake" class remained consistently strong, showing that the model generalizes well across data distributions.

10. Software Testing

To ensure the reliability and robustness of the deepfake detection system, multiple types of testing strategies were applied throughout the development lifecycle.

10.1 Functional Testing

10.1.1 Unit Testing

Each core module was tested in isolation to confirm accurate and independent functionality:

- Frame Extraction Module
- Face Detection and Cropping
- Model Inference (MobileNetV2)
- Result Generation and Rendering

10.1.2 Integration Testing

Integration testing verified that the individual components of the system interact correctly with each other:

- Ensured that the output from preprocessing is seamlessly passed to the MobileNetV2 model
- Verified the transition from model output to UI display

10.1.3 System Testing

End-to-end testing was conducted on the full web-based system:

- Video upload via web interface
- Frame extraction and face detection
- Model prediction
- Display of results on the frontend

  Confirmed that the system meets all specified functional requirements.

10.1.4 Validation Testing

This type of testing verifies that uploaded video files are correctly processed and classified as Real or Fake, accompanied by an appropriate confidence score.

The system's predictions were validated against labeled test data to ensure correct classification behaviour.

10.2 Non-functional Testing

10.2.1 Performance Testing

Measured the average processing time per video using varying frame counts:

- Tested with 10, 20, 40, and 100 frames
- Ensured timely predictions for real-time usability

10.2.2 Load Testing

Simulated multiple concurrent users uploading and processing videos to observe system stability:

- Evaluated resource utilization and model responsiveness under stress
- System maintained consistent performance under simultaneous video uploads

10.2.3 Compatibility Testing

Ensured cross-platform and cross-browser compatibility:

- Browsers Tested: Google Chrome, Mozilla Firefox
- Operating Systems: Windows 10/11, Ubuntu Linux
- Verified consistent layout rendering, upload success, and prediction results across all tested environments

11. Conclusion.

In this project, we proposed and implemented a deep learning-based system for detecting deepfake videos using the MobileNetV2 architecture. Deepfakes, created through advanced AI and generative models, pose a growing threat to digital integrity, individual privacy, and public trust. Their ability to produce hyper-realistic synthetic content has made it increasingly difficult to distinguish between real and fake media, especially on social media platforms where misinformation can spread rapidly.

To address this challenge, our approach utilized MobileNetV2, a lightweight yet powerful Convolutional Neural Network (CNN), to extract meaningful spatial features from facial regions of video frames. Compared to more computationally intensive models like ResNet or ResNeXt, MobileNetV2 offers a balanced trade-off between accuracy and efficiency, making it suitable for deployment in real-time or resource-constrained environments.

The model was trained on a balanced dataset sourced from FaceForensics++ and Celeb-DF, comprising both real and fake videos. Each video underwent a comprehensive preprocessing pipeline including:

- Frame extraction
- Face detection and cropping
- Resizing to 224x224 pixels
- A maximum of 150 frames per video was used for training and inference.

Our trained model delivered high accuracy in classifying videos as Real or Fake, supported by softmax-based confidence scores. These scores provide insight into the reliability of each prediction. The detection system was wrapped in a Flask-based web application, offering a user-friendly interface for uploading videos and receiving real-time classification results.

In addition, the system was thoroughly validated through:

- Unit Testing
- Integration Testing
- Performance and Load Testing

These testing phases ensured that each module and the entire system worked reliably and efficiently across different usage scenarios. One of the key strengths of our system lies in its modular and scalable architecture, making it suitable for future enhancements, such as:

- Full-body deepfake detection
- Audio-based manipulation analysis
- Live-stream verification tools

Moreover, the efficiency of MobileNetV2 enables deployment on mid-range hardware, including devices with limited GPU capabilities or cloud-based platforms, increasing its practical applicability.

In summary this project successfully demonstrates a robust, efficient, and accessible deepfake detection system. It lays a foundation for real-world deployment in applications like:

- Social media moderation
- News/media verification tools
- Browser security extensions
- Law enforcement digital forensics

As the threat of synthetic media continues to evolve, expanding the dataset and integrating multi-modal detection (video, audio, text) will further enhance the system's ability to safeguard digital content.

12. Future Scope

The fight against deepfake technology is ongoing and ever-evolving. While our project provides a solid foundation for detecting facial deepfakes using the MobileNetV2 architecture, there are several directions in which this system can be improved and extended in the future.

1. The current system is limited to detecting deepfakes focused on faces. However, with advancements in generative AI, full-body deepfakes and voice-based manipulations are also becoming common. In the future, the system can be extended to detect such manipulations by integrating pose estimation models and audio analysis pipelines.

2. our solution is currently a web-based platform where users upload videos for analysis. For real-world usability, the system can be enhanced into a browser plugin or a mobile app that can run deepfake detection in real-time. This would allow users to verify videos instantly while browsing or chatting, making the tool more accessible and practical.

3. To further improve accuracy and robustness, the model can be trained on a much larger and more diverse dataset including multi-lingual, multi-ethnic, and varying lighting/background conditions. This would help in improving generalization and reducing bias.

Another important aspect is model optimization for real-time processing. Although MobileNetV2 is lightweight, techniques like quantization, pruning, or converting the model to TensorFlow Lite or ONNX format could make it faster and more efficient for mobile and embedded deployment.

Lastly, collaborations with social media companies could lead to large-scale integration of this technology for automatic screening and flagging of suspicious videos before they are publicly posted. This can significantly help in reducing the spread of misinformation.

Overall, the future scope of this project is vast, and with continuous improvements, it can become a powerful tool for safeguarding digital truth in today’s AI-driven world.

13. References

1. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). *FaceForensics++: Learning to Detect Manipulated Facial Images*. Proceedings of the IEEE International Conference on Computer Vision (ICCV). https://arxiv.org/abs/1901.08971

2. Li, Y., Chang, M., & Lyu, S. (2019). *Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/1909.12962

3. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/1801.04381

4. Chollet, F. (2015). *Keras: The Python Deep Learning Library*. https://keras.io/

5. OpenCV Documentation. *Open Source Computer Vision Library*. https://opencv.org/

6. Flask Documentation. *Flask: A lightweight WSGI web application framework*. https://flask.palletsprojects.com/

7. Face Recognition Library by Adam Geitgey (2018). https://github.com/ageitgey/face_recognition

8. Google Colaboratory. *Cloud-based Jupyter Notebook Environment*. https://colab.research.google.com/

9. Google Cloud Platform (GCP). https://cloud.google.com/