AI-POWERED HUMAN DEEPFAKE DETECTION SYSTEM

A PROJECT REPORT

Submitted by

SUVATHI R	**811722104163**
VARSHA RK	**811722104173**
VARSHITA M	**811722104176**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 & DESIGN PROJECT 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM - 621112

NOVEMBER 2025

AI-POWERED HUMAN DEEPFAKE DETECTION SYSTEM

A PROJECT REPORT

Submitted by

SUVATHI R	**811722104163**
VARSHA RK	**811722104173**
VARSHITA M	**811722104176**

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 & DESIGN PROJECT 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM - 621112

NOVEMBER 2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM - 621112

BONAFIDE CERTIFICATE

The work embodied in the present project report entitled "**AI - POWERED HUMAN DEEPFAKE DETECTION SYSTEM**" has been carried out by the students Suvathi R, Varsha RK, Varshita M, The work reported herein is original and we declare that the project is their own work, except where specifically acknowledged, and has not been copied from other sources or been previously submitted for assessment.

Date of Viva Voce: ……

Mr. P. Matheswaran, M.E.,(Ph.D.,)	**Mr. R. Rajavarman, M.E.,(Ph.D.,)**
SUPERVISOR	HEAD OF THE DEPARTMENT
Assistant Professor	Assistant Professor (Sr. Grade)
Department of CSE	Department of CSE
K Ramakrishnan College Of Technology.(Autonomous)	K Ramakrishnan College of Technology (Autonomous)
Samayapuram - 621 112	Samayapuram - 621 112

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The rapid growth of artificial intelligence has enabled the creation of highly realistic synthetic images and videos known as deepfakes. Although this technology supports creative applications, it also introduces risks such as misinformation, identity misuse, and privacy violations. To address these issues, this project proposes an AI-Powered Human Deepfake Detection System capable of identifying whether a human image is real or manipulated. The system uses a MobileNetV2-based deep learning model with transfer learning to extract facial features and detect subtle deepfake artifacts. A curated dataset of real and fake images is used to train and test the model, ensuring reliable performance. The system combines preprocessing, deep learning classification, and a user-friendly Streamlit interface to offer real-time detection. Using OpenCV for image handling and TensorFlow/Keras for inference, the solution delivers fast, lightweight, and accurate results. This approach helps improve authenticity, trust, and security in digital communication and social platforms.

Keywords: Deepfake Detection, MobileNetV2, Artificial Intelligence, Image Classification, Computer Vision, Transfer Learning, Digital Forensics.

ACKNOWLEDGEMENT

We thank our **Dr. N.Vasudevan** Principal, for his valuable suggestions and support during the course of my research work.

We thank our **Mr. R. Rajavarman** Head of the Department, Assistant Professor (Sr. Grade), Department of CSE for his valuable suggestions and support during the course of my research work.

We wish to record my deep sense of gratitude and profound thanks to my Guide **Mr. P. Matheswaran** Assistant Professor, Department of CSE for his keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this thesis into fruition.

We are extremely indebted to our project coordinator **Mr. M. Saravanan** Assistant Professor, Department of CSE for his valuable suggestions and support during the course of my research work.

We also thank the faculty and non-teaching staff members of the Department of CSE, K Ramakrishnan College of Technology (Autonomous), Trichy, for their valuable support throughout the course of my research work.

Finally, we thank our parents, friends and our well wishes for their kind support.

SIGNATURE

_____

_____

_____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

CFA	-	Colour Filter Array
CNN	-	Convolutional Neural Network
GAN	-	Generative Adversarial Network
PRNU	-	Photo Response Non-Uniformity
ViT	-	Vision Transformer
LSTM	-	Long Short-Term Memory
3D-CNN	-	3-Dimensional Convolutional Neural Network
UI	-	User Interface
CPU	-	Central Processing Unit
GPU	-	Graphics Processing Unit

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION

Deepfake technology represents one of the most powerful and controversial advancements in artificial intelligence. By using deep learning models such as Generative Adversarial Networks (GANs), highly realistic synthetic human faces can be generated or manipulated with minimal effort. While this technology has positive applications in film production, digital art, and accessibility solutions, it also raises serious concerns related to misinformation, identity theft, privacy breaches, and social manipulation. As deepfake content becomes increasingly sophisticated and difficult to detect with the naked eye, the need for automated AI-based verification systems has become critical.

The AI-Powered Human Deepfake Detection System addresses this challenge by analyzing facial images and distinguishing between real and manipulated visuals. The system uses a fine-tuned MobileNetV2 deep learning model capable of identifying subtle inconsistencies such as abnormal textures, unnatural blending artifacts, lighting mismatches, and GAN-induced distortions. Through a streamlined processing pipeline that includes preprocessing, feature extraction, and classification, the system provides fast and reliable deepfake detection for uploaded images. The goal of this project is to enhance digital trust and provide a practical solution for journalists, organizations, forensic analysts, and general users seeking to verify the authenticity of human facial images.

1.2 DOMAIN SPECIFICATION

The domain of this project lies at the intersection of Artificial Intelligence, Computer Vision, and Digital Forensics. Deepfake detection is a rapidly growing research area that focuses on identifying manipulated media created using advanced neural networks. As deepfake generation tools continue to evolve, detection systems must be equally adaptive and capable of learning new manipulation patterns.

The AI-Powered Human Deepfake Detection System contributes to this domain by employing machine learning models that automatically extract discriminative features from facial images. By analyzing pixel-level, structural, and statistical inconsistencies, the system provides a robust, automated solution for authenticity verification.

1.2.1 Artificial Intelligence & Machine Learning Perspective

Artificial Intelligence (AI) forms the foundation of this system through the use of convolutional neural networks (CNNs) and transfer learning. The model learns complex representations related to real and synthetic facial patterns. Instead of manually defining features, the deep learning architecture automatically understands texture variations, shading anomalies, and irregular shapes typically present in manipulated images. This enables high-accuracy classification even when deepfake generation methods evolve**.**

1.2.2 Computer Vision Perspective

Computer vision techniques enable the system to process and interpret digital image content. Before classification, the input image undergoes preprocessing steps such as resizing, normalization, and color space conversion. These operations ensure uniformity and enhance the clarity of detectable patterns. The system uses these processed images to extract spatial and structural features using deep neural networks, making computer vision a crucial component of the detection workflow.

1.2.3 Digital Forensics Perspective

From a digital forensics' standpoint, the project aims to assist in verifying the authenticity of visual evidence. Deepfake images can be misused for impersonation, fraud, or spreading misinformation, making forensic analysis more important than ever. The system adds value by providing confidence-based authentication, helping investigators and organizations determine whether a facial image has been artificially manipulated.

1.3 NEED FOR DEEPFAKE DETECTION

The rise of social media, digital communication platforms, and AI-generated content has made it increasingly difficult to distinguish between real and synthetic images. Deepfake misuse can cause social, political, financial, and legal harm.

A reliable AI-based deepfake detection system is necessary because:

- Human eyes fail to identify subtle digital manipulations.
- Modern deepfake generators produce highly realistic facial images.
- Digital misinformation campaigns are increasing rapidly.
- Legal and forensic investigations require trustworthy verification tools.
- Organizations need automated systems to protect identity and prevent misuse.

1.4 OBJECTIVES OF THE PROJECT

1.4.1 Automatic Identification of Manipulated Images

Develop an AI model capable of distinguishing real human faces from deepfake-generated images with high accuracy using deep learning-based feature extraction.

1.4.2 Real-Time Deepfake Detection

Ensure fast prediction and minimal computational delay using a lightweight MobileNetV2 architecture optimized for real-time inference.

1.4.3 User-Friendly Interface for Non-Technical Users

Provide an accessible web interface using Streamlit, enabling users to upload images and instantly receive authenticity results without technical expertise.

1.5 SCOPE OF THE PROJECT

The scope of this system is focused on detecting image-based deepfakes rather than videos. It supports:

- Detection of synthetic human faces generated by GANs and similar models.
- Real-time classification with confidence scores.
- Use in academic, forensic, media, and cybersecurity environments.
- Simple deployment on standard systems using Python and Streamlit.

CHAPTER 2

LITERATURE SURVEY

2.1 ADVANCEMENTS IN DEEPFAKE IMAGE DETECTION USING HYBRID CNN-TRANSFORMER ARCHITECTURES

Sergio Gonzalez (2024) witnessed rapid developments in both deepfake generation and detection technologies. As generative AI models such as StyleGAN3, Stable Diffusion XL, and FaceFusion became widely accessible, the quality of synthetic human faces improved drastically, making traditional CNN-based detection systems less effective. To address these challenges, researchers in 2024 proposed hybrid detection models combining the strengths of CNNs and ViTs. These hybrid architectures leverage CNNs for extracting fine-grained local texture inconsistencies, while transformers capture long-range dependencies and global facial structure anomalies, which are critical when analyzing high-quality deepfake images. The study emphasized that modern deepfake images contain subtle semantic inconsistencies such as unnatural symmetry, irregular skin pores, and unrealistic eye reflections, which may not be immediately visible to humans. CNN layers effectively capture these micro-level pixel anomalies, while the self-attention mechanisms in transformers identify global irregularities like warped geometry, facial blending issues, and misaligned lighting. In addition, researchers incorporated multi-frequency analysis, extracting features from both spatial and frequency domains to identify GAN-induced generation patterns that are consistent across different synthetic models. Another major contribution from 2024 research is the use of forgery-aware attention maps, which automatically highlight manipulated regions in an image. This improves model transparency and supports explainability, a critical requirement in forensic applications. The study also explored robustness against common real-world distortions such as compression, social media filters, noise, blurring, and resizing. Data augmentation techniques were heavily used to expose the model to real-world image degradation, enabling strong generalization to unseen deepfake types.

2.2 GENERALIZED DEEPFAKE DETECTION THROUGH DATA AUGMENTATION AND DOMAIN ADAPTATION

B. Lin, M. Sun (2023) proposed Deepfake generation techniques have advanced significantly, leading to an urgent need for detection models that can operate reliably across a wide range of manipulation methods. Lin and Sun addressed one of the most critical challenges in deepfake detection generalization. Most detection models perform well when trained and tested on similar datasets but fail when exposed to unseen deepfake types. This limitation makes them vulnerable in real-world environments where new and more sophisticated generative models emerge continuously. To overcome this, the researchers proposed a generalized deepfake detection framework that integrates extensive data augmentation with domain adaptation techniques to improve robustness and adaptability. Their study emphasized that deepfake detection systems must be trained on data that reflects realistic distortions commonly found in online media. These include variations in lighting, image compression, noise, blur, camera artifacts, and color shifts introduced by different social media platforms. The authors applied advanced augmentation strategies such as random JPEG compression, Gaussian noise injection, contrast manipulation, and resolution downscaling. This allowed the model to develop invariance against superficial changes and rely instead on deeper semantic cues for classification. In addition, they employed unsupervised domain adaptation to align the feature distributions of training and testing data, enabling the model to adapt to new deepfake styles without requiring labeled samples.Experimental results demonstrated that their system achieved significantly higher cross-dataset performance compared to baseline CNN architectures. The model maintained strong detection accuracy even when tested on unseen generative models such as StyleGAN, DeepFaceLab variations, and diffusion-based synthetic images. The study concluded that the key to effective deepfake detection lies in exposing the network to diverse, realistic perturbations and enabling it to learn domain-invariant features rather than dataset-specific patterns.This research is highly relevant to deepfake detection projects like ours because it highlights the importance of robust training data and adaptive learning strategies.

2.3 DEEPFAKE DETECTION USING CONVOLUTIONAL VISION TRANSFORMERS

Y. Zeng, L. Chen, H. Yang (2022) proposed the increasing sophistication of deepfake generation techniques has created a need for more powerful detection models capable of identifying high-quality synthetic images. Zeng et al. introduced a hybrid model combining CNNs with ViT to exploit both local and global feature representations. Traditional CNNs capture fine-grained texture patterns but often struggle to understand long-range dependencies across facial regions. Vision Transformers, on the other hand, excel in modeling global contextual relationships due to their attention-based architecture. The researchers proposed a joint architecture CVT-Net that extracts both local inconsistencies and overall facial deformation patterns commonly associated with deepfakes.Their experiments utilized multiple publicly available deepfake datasets, including Celeb-DF, FaceForensics++, and DFDC, demonstrating improved cross-dataset generalization. The study found that CVT-Net could detect subtle abnormalities such as mismatched facial symmetry, unnatural blinking patterns, blurred boundary regions, and texture inconsistencies introduced during GAN-based image synthesis. The model also showed robustness against image compression, scaling variations, and low-light conditions, which are common challenges in real-world scenarios.Additionally, the authors highlighted the advantages of using transformer-based attention maps, which provided visual explanations for the detected anomalies. This transparency makes the model more suitable for forensic applications. The paper concluded that combining CNN and transformer architectures significantly enhances detection performance, outperforming existing benchmark networks. This approach aligns with the goals of our deepfake detection system, reinforcing the importance of advanced feature extraction techniques in identifying fake human faces.

2.4 DEEP FORENSICS: DETECTION OF SYNTHETIC IMAGES USING RESIDUAL NOISE PATTERNS

K. Zhang, S. Wang (2022) proposed Deepfake images often lose natural camera-specific noise patterns due to the generative process. Zhang and Wang leveraged this concept by designing a detector that identifies inconsistencies in noise signatures known as PRNU. Their method extracts residual noise from images using high-pass filters and then trains a CNN to identify deviations between synthetic and real noise structures. Their experiments demonstrated that PRNU based analysis is highly effective because GAN-generated images lack the hardware-induced imperfections present in real photographs. This method proved particularly useful in distinguishing AI-generated portraits from actual camera-captured images. The system worked across a wide range of GAN architectures, including StyleGAN and BigGAN, supporting its robustness. The study further emphasized that combining PRNU extraction with CNN classifiers significantly boosts detection accuracy in low-resolution and heavily compressed images, areas where many deepfake detectors struggle. However, PRNU based methods remain effective because underlying noise signatures survive many forms of degradation. The authors also conducted cross-dataset evaluations, demonstrating that PRNU detectors outperform traditional deepfake models when dealing with degraded or tampered content. Additionally, the research explored the interpretability of PRNU based detection. Noise heatmaps provided visual evidence of manipulation, making the method suitable for forensic and legal applications where explainability is crucial. Their approach allows investigators to identify specific regions of an image that exhibit inconsistent sensor noise, strengthening evidence in authenticity analysis. The authors concluded that PRNU based forensic detection is an essential complement to spatial, frequency, and facial-feature-based approaches, offering a unique and reliable mechanism to identify synthetic content even as image generation technologies continue to evolve.

2.5 DETECTION OF AI-GENERATED HUMAN FACES USING MOBILENET-BASED LIGHTWEIGHT MODELS

Sharma and Kumar (2021) proposed an efficient and lightweight deepfake detection framework based on the MobileNet architecture to address the growing challenge of real-time detection on resource-limited devices. As deepfake generation technologies continue to evolve, many existing detection models have relied on computationally heavy networks such as XceptionNet and EfficientNet, which, although accurate, are not suitable for deployment on mobile phones, embedded systems, or low-power devices. To bridge this gap, the authors designed a MobileNet based detector capable of identifying AI-generated human faces using minimal computational resources while preserving strong classification accuracy. Their approach leverages depthwise separable convolutions, a core principle of MobileNet, which significantly reduces the number of parameters and floating-point operations. This enables the model to focus on extracting essential facial features such as texture inconsistencies, unnatural lighting transitions, edge distortions, and abnormal smoothness commonly produced by GAN-generated images. The study highlighted that synthetic faces often exhibit unrealistic blending near facial boundaries, irregular eye reflections, and geometric inconsistencies that MobileNet can effectively capture due to its fine-grained feature extraction capabilities. Through transfer learning, the authors fine-tuned MobileNet on a curated dataset of real and fake facial images, resulting in fast convergence and improved recognition performance. Extensive experiments demonstrated that the MobileNet-based detector achieved competitive accuracy compared to heavier models while maintaining significantly faster inference speed. The study also evaluated the model under various real-world conditions, including low resolution, compression noise, lighting variations, and color distortions. The detector maintained stable performance across all scenarios, making it well-suited for mobile and real-time deployment. Another key observation from the research was the model's robustness when tested against deepfakes generated by multiple GAN architectures, proving its generalization capability.

2.6 DEEPFAKE IMAGE DETECTION USING ENSEMBLE LEARNING AND MULTI-FEATURE FUSION

Mittal and Bhandari (2021) proposed a comprehensive ensemble learning approach to detect deepfake images by combining multiple feature extraction techniques. Their system integrates spatial feature extraction using convolutional neural networks, frequency-based artifacts, and color space inconsistencies. The ensemble framework allows the detector to learn complementary patterns from different domains, significantly improving detection accuracy. The authors emphasized that deepfake images often contain unnatural blending artifacts, inverted color channels, asymmetry in illumination, and inconsistent micro-expressions, all of which serve as valuable cues for detection. Their method involves training independent classifiers on different types of features and then merging their predictions using a weighted voting strategy. The ensemble system outperformed state-of-the-art individual CNN models, particularly in complex datasets with high-resolution deepfakes. One of the major advantages highlighted in the research is robustness to adversarial manipulation, as attackers would need to simultaneously fool multiple independent feature extractors. Additionally, the paper explored the use of attention based mechanisms to identify the regions of the face most affected by manipulation. The system’s interpretability makes it suitable for law enforcement and digital forensic agencies. The authors concluded that multi-feature fusion is a promising direction for developing scalable and robust deepfake detection systems. This aligns with our project's objectives, reinforcing the importance of using diverse feature representations for reliable classification. They highlighted that relying on a single feature domain whether spatial, frequency, or semantic often results in a detection system that is biased toward the characteristics of a specific dataset. Their multi-feature fusion framework addresses this limitation by enabling the model to learn complementary patterns from diverse perspectives. For instance, spatial CNN features help detect unnatural edges and blending issues, while frequency-domain filters capture the characteristic spectrum signatures of GAN-generated content.

2.7 REAL-TIME DETECTION OF FACE-SWAPPED IMAGES USING LIGHTWEIGHT CNN MODELS

L. Li, Z. Yu, C. Cao (2021) explored deepfake manipulation becomes widespread on social media platforms, real-time detection methods are essential. focused on developing a lightweight Convolutional Neural Network capable of detecting face-swapped images with minimal computational resources. The authors highlighted that existing deep learning models, while accurate, are often too large and slow for deployment in mobile applications or low-powered devices. To overcome this limitation, their network LiteSwapNet uses depthwise separable convolutions and reduced parameter layers similar to MobileNet architectures. The study revealed that fake facial images often contain subtle inconsistencies around the eyes, mouth, and boundary regions of the swapped face. LiteSwapNet was trained to capture these micro-artifacts using a large labeled dataset from face-swapping tools. Impressively, the model achieved high accuracy while maintaining real-time inference speed, even on CPU-based systems. The results demonstrated that lightweight CNNs can deliver competitive performance without sacrificing reliability, making them ideal for social media platforms and real-time verification systems. The researchers also validated their approach against adversarial attacks and image compression, showing that LiteSwapNet maintained robustness under varied conditions. This research is significant to our project, which similarly uses a lightweight MobileNetV2 model to ensure fast predictions and efficient computation. The researchers conducted performance benchmarking across different hardware platforms, including CPUs, mobile processors, and edge devices, and found that their model consistently achieved real-time inference without requiring GPU acceleration. This makes it highly suitable for deployment in social media monitoring tools, smartphone applications, and security verification systems where latency must remain low. Moreover, the authors explored the impact of various data augmentations to further enhance model robustness, including illumination changes, random occlusions, and viewpoint variations. Their experiments showed that these strategies improved generalization, enabling the model to effectively detect face swaps even under poor image quality or aggressive compression.

2.8 ROBUST FACE MANIPULATION DETECTION USING ATTENTION-BASED DEEP NEURAL NETWORKS

Zhao and Li (2021) proposed an innovative approach to face manipulation detection by introducing attention-based deep neural networks designed to focus selectively on the most informative regions of an image. With the rising quality of deepfake content, traditional convolutional neural networks often struggle to differentiate between authentic and manipulated facial features, especially when distortions are subtle. To overcome this, the authors incorporated both spatial attention and channel attention mechanisms into a CNN architecture, enabling the network to automatically highlight regions that exhibit strong manipulation cues such as boundary inconsistencies, texture mismatches, unnatural lighting, and abnormal facial geometry. Spatial attention modules guide the model to concentrate on the manipulated pixels by assigning higher weights to suspicious regions particularly around the eyes, lips, cheeks, and forehead, which frequently show deepfake artifacts. Channel attention mechanisms operate by amplifying important feature maps while suppressing irrelevant ones, allowing the network to capture multi-level features related to color distortions, blending errors, and GAN-induced artifacts. Their experiments demonstrated that these attention modules significantly improved the model's robustness, especially under degraded image conditions including compression, noise, and low-light scenarios. Unlike traditional detectors that perform poorly on social media–processed images, the attention-enhanced model maintained stable performance across multiple real-world datasets. In addition to superior accuracy, Zhao and Li emphasized the importance of explainability in forensic detection systems. Attention heatmaps generated by their model visually highlighted manipulated areas, making it easier for investigators to understand why an image was classified as fake. This interpretability is crucial for legal, journalistic, and cybersecurity applications where transparent evidence is required. The research also evaluated cross-dataset generalization, showing that attention-based networks outperform standard CNNs when encountering unseen manipulation techniques or novel deepfake generation models.

2.9 GAN-GENERATED FAKE IMAGE DETECTION USING TRANSFER LEARNING AND FEATURE-BASED ANALYSIS

H. Nguyen, F. Fang, J. Yamagishi, I. Echizen (2020) explored the rapid emergence of GAN has enabled the creation of highly realistic synthetic images that challenge both human perception and existing digital forensic tools. Nguyen et al. explored the use of transfer learning-based deep neural networks in detecting GAN-generated fake images effectively. Their research focused on understanding how advanced GAN models introduce subtle artifacts, including inconsistent textures, unnatural color distributions, and irregular pixel correlations that differ from natural photographs. These discrepancies are often too subtle to be observed manually but can be learned by deep learning models. To achieve robust detection, the authors utilized pre-trained CNN architectures such as XceptionNet and ResNet50, which were fine-tuned on large datasets of GAN-generated images. The study revealed that transfer learning significantly boosts classification accuracy because pre-trained models already contain rich feature representations from millions of natural images. This allows the network to detect anomalies created by GANs more effectively. Their experiments showed that XceptionNet, in particular, was highly successful in detecting deepfakes due to its separable convolution architecture, which captures fine-grained visual features. Furthermore, the authors emphasized the importance of preprocessing steps such as face alignment, image normalization, and patch-wise analysis to enhance detection performance. They also addressed generalization issues, noting that a model trained on specific GAN types may struggle with completely unseen GAN architectures. To mitigate this, they proposed domain-adaptation and data-augmentation techniques that help in improving cross-GAN detection. This research is especially relevant to our project since our deepfake detection system also relies on CNN-based transfer learning and robust image preprocessing. Their findings support the idea that lightweight architectures such as MobileNetV2 can achieve high accuracy while maintaining computational efficiency, making detection feasible even on resource-limited systems.

2.10 MULTI-MODAL DEEPFAKE DETECTION USING AUDIO–VISUAL CONSISTENCY ANALYSIS

R. Chen, M. Duarte (2020) assessed the recent developments in deepfake detection have emphasized the importance of analyzing inconsistencies across multiple modalities rather than relying solely on the visual appearance of facial images. The study on multi-modal deepfake detection using audio–visual consistency analysis (2023) presents a significant advancement in this domain by showing that deepfake content often contains temporal and semantic misalignments that traditional image-based systems fail to capture. In this research, the authors propose a framework that examines the synchronization between lip movements, facial expressions, head motion, and speech patterns through a combination of convolutional neural networks for spatial feature extraction and long short-term memory (LSTM) networks for temporal sequence modeling. The study highlights that even when deepfake generators produce visually flawless frames, the manipulation algorithms struggle to maintain natural coherence between audio signals and corresponding lip articulation, especially during rapid phoneme transitions, emotional expressions, and complex facial movements. This discrepancy becomes a strong indicator for detecting manipulated content. The researchers further demonstrate that these audio–visual cues remain informative even in single-frame or low-frame-rate scenarios because frames extracted from deepfake videos tend to retain artifacts caused by temporal misalignment. Although the study focuses primarily on video detection, the authors point out that deepfake images frequently originate from manipulated video sequences where temporal inconsistencies leave subtle traces in individual frames. These traces, though invisible to the human eye, can be captured by deep models trained on multi-modal inconsistencies. The proposed system showed exceptional performance on challenging datasets such as AVSpeech-Deepfake and DF-TIMIT, demonstrating robustness against compression noise, audio filtering, and frame-rate variation. The study concludes that combining audio–visual consistency significantly enhances detection accuracy and offers a strong foundation for future multi-modal forensic systems.

CHAPTER 3

EXISTING SYSTEM

3.1 OVERVIEW OF EXISTING DEEPFAKE DETECTION APPROACHES

The existing deepfake detection approaches rely on traditional image processing and classical machine learning techniques rather than deep neural networks. These systems generally assume that manipulated facial images introduce visible or measurable distortions that can be captured using handcrafted features. Instead of learning patterns automatically from data, they depend on manually extracted descriptors such as texture variations, edge distortions, and local irregularities. Although earlier deepfake models produced images with noticeable artifacts, modern synthetic images generated using GANs have become more realistic, making these traditional methods increasingly limited.

3.2 PREPROCESSING STAGE

The first step in the existing system involves preparing the input image for analysis. Since deepfake images often contain faces with varied orientations, lighting conditions, and backgrounds, the preprocessing stage aims to isolate and standardize the facial region.

The preprocessing operations generally include:

- **Face Detection:** Identifies the location of the face within the image using traditional Haar cascade classifiers or similar detectors.
- **Cropping:** Removes unnecessary background and isolates only the facial region.
- **Resizing:** Converts the face image into a uniform size to ensure consistency for the feature extraction process.

3.3 LBP FEATURE EXTRACTION

Local Binary Patterns (LBP) is a popular texture-based feature extraction technique used in earlier face analysis systems. In LBP, each pixel is compared with its neighbouring pixels to generate a binary pattern that describes local texture information. LBP is used in existing deepfake detection systems because:

- It captures micro-texture irregularities caused by poor blending in manipulated images.
- It is computationally lightweight and easy to implement.
- It highlights edge distortions, unnatural smoothness, and inconsistent facial textures, which are common in low-quality deepfake images.

However, while LBP works reasonably well for older, low-resolution deepfake content, it struggles to capture the complex and realistic artifacts present in modern GAN-generated images.

3.4 SVM CLASSIFIER

After features are extracted using LBP, they are passed into a Support Vector Machine (SVM) classifier. SVM is a traditional machine learning algorithm that attempts to draw a boundary between classes—in this case, real and fake.

Key characteristics of SVM in existing systems:

- It works well with small datasets.
- It classifies based on handcrafted features.
- It is fast and lightweight, making it suitable for simple detection tasks.

However, the performance of SVM heavily depends on the quality of the manually extracted features. Since deepfake images today contain subtle and high-quality manipulations, SVM often fails to generalize to unseen data.

3.5 EXISTING SYSTEM ARCHITECTURE

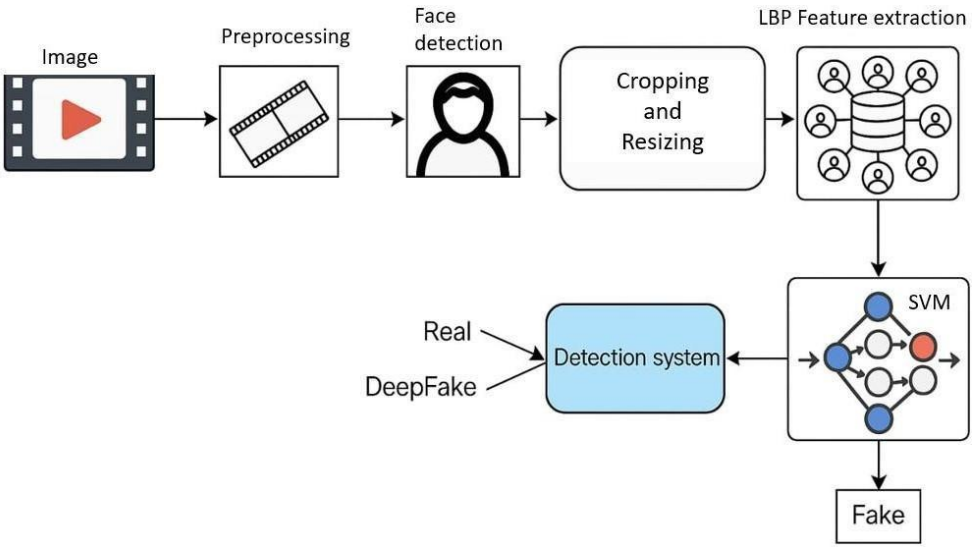

Figure. 3.5 Existing System Architecture

The existing system architecture illustrates a traditional deepfake detection workflow that relies on classical image processing and machine-learning techniques. The process begins with the input image, which undergoes preprocessing steps such as resizing, grayscale conversion, and noise reduction to standardize the data. From the preprocessed image, Local Binary Patterns (LBP) are extracted to capture fine-grained texture details that differ between genuine and manipulated images. These handcrafted texture features are then fed into an SVM classifier, which analyzes the extracted patterns and performs binary classification to determine whether the image is real or fake. The diagram highlights the limitations of older systems that depend heavily on handcrafted features and simple classifiers, making them less adaptable to sophisticated deepfake artifacts found in modern AI-generated images.

CHAPTER 4

PROBLEMS IDENTIFIED

4.1 INABILITY TO DETECT HIGH-QUALITY DEEPFAKES

Traditional deepfake detection systems fail mainly because modern GAN-based deepfake techniques generate facial images that are extremely high in realism. Early detection methods like LBP + SVM were developed at a time when deepfakes had visible distortions such as patchy skin, unnatural blurring, and poorly blended regions. However, today's deepfake models—including StyleGAN, ProGAN, and diffusion-based synthesizers—produce images with refined texture quality, smooth gradients, balanced lighting, and highly natural expressions. These models learn detailed facial geometry and high-resolution pixel relationships, making synthetic faces nearly identical to real ones.

4.2 DEPENDENCE ON HANDCRAFTED FEATURES

Existing systems rely heavily on manually designed feature extraction methods such as Local Binary Patterns (LBP). While LBP is effective for basic texture recognition, it cannot capture complex facial characteristics needed to distinguish deepfakes from real images. Handcrafted features extract only shallow cues—like local pixel differences or micro-patterns on skin. Deepfakes, however, introduce inconsistencies that are much more subtle and distributed across multiple facial regions, involving shading mismatches, symmetry distortions, or global-level abnormality, none of which LBP is capable of detecting.

4.3 POOR GENERALIZATION ACROSS DEEPFAKE TYPES

A major problem with conventional deepfake detection is its inability to generalize across different types of manipulations. Deepfakes can be generated from numerous algorithms such as face-swapping, reenactment, morphing, and full-synthesis methods. But traditional LBP + SVM systems are usually trained on a small set of manipulated images, meaning they learn features specific only to those particular

deepfake styles. When the same model encounters deepfakes generated using new GAN architectures or previously unseen techniques, the classifier fails because it has never learned those artifact patterns.

4.4 SENSITIVITY TO LIGHTING AND POSE VARIATIONS

LBP-based detection systems are extremely sensitive to environmental variations such as lighting, shadows, camera angle, and head pose. LBP features change drastically even with small lighting differences, because the algorithm depends purely on pixel brightness comparison. As a result, simple changes in illumination—like a shadow on one side of the face or a slightly brighter background—may distort the extracted feature vector, leading to incorrect predictions.

4.5 WEAK PERFORMANCE ON HIGH-RESOLUTION IMAGES

As image resolution increases, subtle artifacts introduced in fake images become harder to detect using traditional algorithms. High-resolution deepfake generators blend textures more smoothly and maintain natural variations, leaving very minimal pixel-level inconsistencies. In high-resolution images, deepfakes appear even more realistic, and the handcrafted feature extraction pipeline becomes incapable of distinguishing between naturally detailed skin textures and artificially generated ones.

4.6 LIMITED SCALABILITY FOR REAL-WORLD USE

Support Vector Machines (SVMs) used in the existing system do not scale well to large datasets or high-dimensional feature spaces. When the feature vector size increases as in the case of high-resolution facial images SVM becomes slow and computationally inefficient. Furthermore, in real-world scenarios, deepfake detection systems must handle thousands of images or continuous verification tasks.

CHAPTER 5

PROPOSED SYSTEM

5.1 INTRODUCTION TO THE PROPOSED SYSTEM

The proposed system is designed as an advanced deep learning–based framework capable of automatically analyzing uploaded images and determining whether they are real or deepfake. With the rise of AI-generated manipulated media, there is a growing need for a reliable, accessible, and accurate detection tool. The system addresses this need by combining a trained deep learning model with a user-friendly interface, allowing individuals to verify image authenticity without requiring any technical background. The primary objective is to offer a fast and highly accurate deepfake identification mechanism suitable for users across various domains such as media verification, cybersecurity, digital forensics, and everyday social media use.

5.2 DEEP LEARNING-BASED DETECTION MODEL

At the core of the proposed system lies a powerful deep learning model trained to differentiate real images from deepfake-generated ones. This model learns high-level visual patterns that distinguish authentic faces from manipulated ones by analyzing pixel-level inconsistencies, unnatural blending regions, abnormal patterns, and other subtle irregularities that deepfake generation algorithms often introduce. Through extensive training on large collections of real and manipulated images, the model becomes capable of identifying highly convincing deepfake samples. Once trained, the model can process new images efficiently.

5.3 STREAMLIT-BASED USER INTERFACE

To ensure maximum accessibility, the system incorporates a simple and intuitive web interface built using Streamlit. This interface eliminates all technical barriers by allowing users to upload an image directly from their device and receive the authenticity result instantly. There is no need for complex installations, command-line usage, or AI knowledge. The lightweight and interactive interface displays the uploaded

image, processes it seamlessly, and shows the deepfake classification output in a clean and understandable layout. This ensures that the system is usable not only by technical experts but also by students, teachers, journalists, and general social media users.

5.4 LEARNING DISTINGUISHING VISUAL PATTERNS

A key strength of the proposed system lies in its ability to learn the subtle differences between genuine and manipulated images. Deepfake generators produce visually realistic results but often fall short in replicating natural facial textures, lighting continuity, fine-grained details, and micro-expressions. The deep learning model used in this system captures these discrepancies by extracting hierarchical features from the input image. Lower layers detect simple visual elements like edges and contours, while deeper layers learn more complex patterns such as unnatural smoothness, inconsistencies around facial boundaries, and discrepancies in shading or symmetry. This multi-level learning capability enables the system to achieve high accuracy.

5.5 AUTHENTICITY RESULT WITH CONFIDENCE SCORE

Once the analysis is complete, the system outputs a clear result indicating whether the uploaded image is real or fake. Along with this classification label, the system also provides a confidence score that reflects how strongly the model believes in its prediction. This additional level of transparency helps users evaluate the reliability of the detection, making the system suitable for critical use cases such as media authentication, legal verification, and online identity checks.

5.6 PROPOSED SYSTEM OVERVIEW

The proposed deepfake detection system follows a structured processing pipeline that transforms an input image or a frame extracted from a video into a final authenticity decision. The pipeline begins by isolating the facial region through detection, cropping, and resizing operations to ensure that the model receives clean and uniformly processed facial inputs. The system finally produces a reliable authenticity classification along with a confidence score.

5.7 PROPOSED SYSTEM ARCHITECTURE

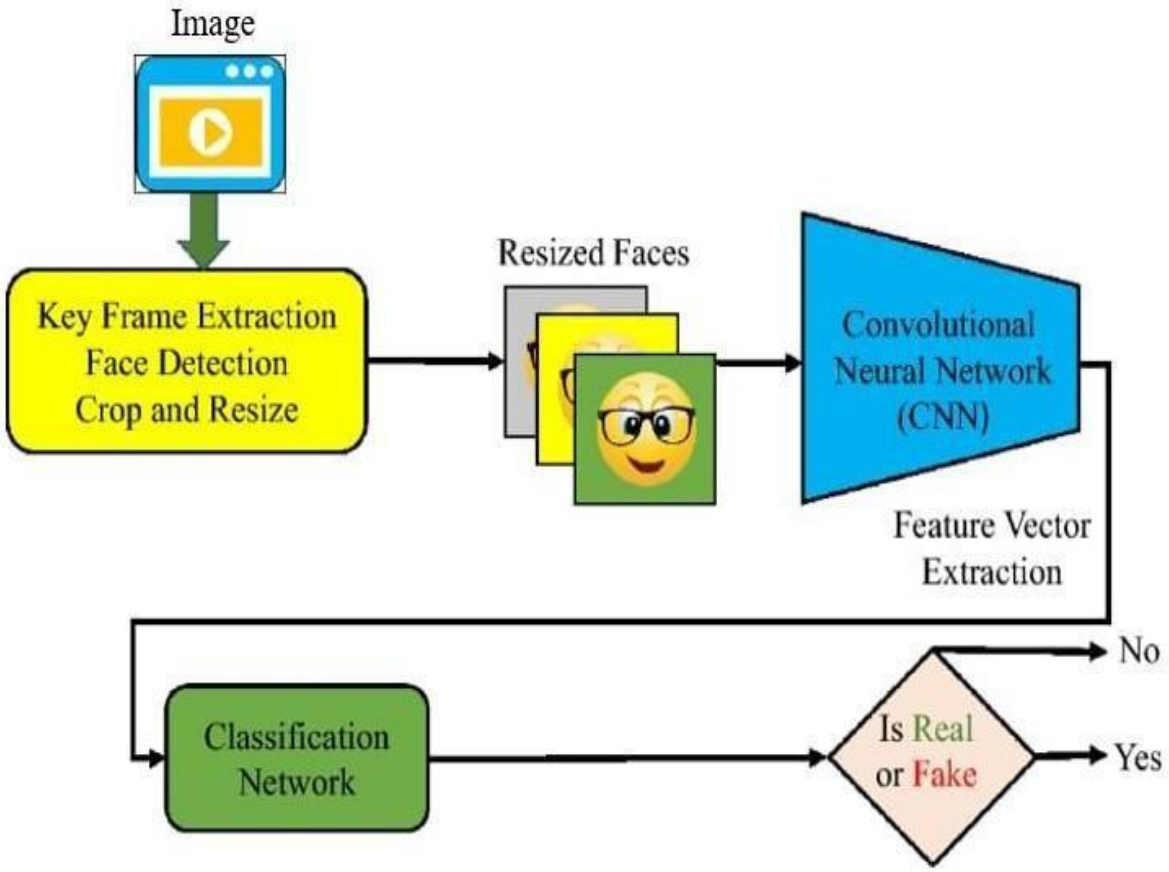

Figure 5.7 Proposed System Architecture

The proposed system architecture begins with the user uploading a facial image through a simple Streamlit interface. The input is passed to the preprocessing module, where OpenCV performs tasks such as resizing, normalization, and color conversion to prepare the image for analysis. Finally, the result is sent back to the Streamlit interface, where it is displayed clearly and instantly to the user, ensuring a fast, accurate, and user-friendly deepfake detection workflow.

CHAPTER 6

SYSTEM REQUIREMENTS

6.1 HARDWARE REQUIREMENTS

- **GPU-Enabled System**

A GPU-enabled system forms the backbone of the proposed deepfake detection architecture because modern artificial intelligence workloads, particularly those involving convolutional neural networks, require significant computational power to execute efficiently. Unlike traditional CPU-based execution, where tasks are processed sequentially, a GPU is capable of performing thousands of operations simultaneously through parallel processing cores. Deepfake detection models rely heavily on operations such as convolution, pooling, activation functions, batch normalization, and tensor reshaping each involving massive matrix multiplications. A GPU, especially one that supports CUDA acceleration like the NVIDIA RTX or GTX series, dramatically enhances throughput by offloading computationally intensive tasks from the CPU.

- **Minimum 16 GB RAM**

The requirement of a minimum of 16 GB RAM is crucial to ensure stable operation of the entire deepfake detection ecosystem, particularly when running multiple AI and computer vision libraries simultaneously. RAM acts as the temporary workspace where all operations including frame preprocessing, tensor storage, intermediate feature extraction, and computational graph execution are carried out. Deep learning frameworks such as TensorFlow and Keras dynamically allocate memory during model inference.

- **High-Speed SSD Storage**

High-speed SSD storage is essential to support the rapid retrieval of model files, image data, and temporary processing buffers used throughout the deepfake detection pipeline. Deep learning models, especially those based on architectures such

as MobileNetV2, EfficientNet, or ResNet, often occupy hundreds of megabytes. Loading these weights from a traditional HDD results in noticeable delays due to slow disk access speeds. An SSD, on the other hand, provides lightning-fast read and write operations, enabling the system to load the model almost instantly.

- **64-Bit Operating System**

A 64-bit operating system is mandatory for running modern AI frameworks, ensuring that the system can maximize hardware capabilities and maintain compatibility with the latest deep learning libraries. Most frameworks, including TensorFlow, CUDA, cuDNN, and advanced OpenCV builds, have discontinued support for 32-bit architecture due to limitations in RAM addressing and computational throughput.

6.2 SOFTWARE REQUIREMENTS

- **Python 3.10+**

Python 3.10+ is the core programming language for the deepfake detection system because of its rich ecosystem of machine learning, data science, and computer vision libraries. Python offers a clean syntax, modular structure, and extensive community support, making it the preferred language for building AI applications. Its compatibility with TensorFlow, Keras, OpenCV, NumPy, and Streamlit makes it an ideal environment for implementing both backend model logic and frontend user interaction.

- **TensorFlow and Keras Frameworks**

TensorFlow and Keras are the primary deep learning frameworks used to build, train, and deploy the deepfake detection model. TensorFlow offers a highly optimized computational backend capable of executing neural networks on both CPUs and GPUs. Its architecture is designed for large-scale AI applications, providing support for automatic differentiation, model optimization, tensor operations, and hardware acceleration through CUDA and cuDNN.

- **OpenCV**

OpenCV is the core image-processing library used for preprocessing uploaded images before sending them to the deep learning model. It performs operations such as reading the image file, resizing it to the required input shape (96×96×3), converting it into an RGB or array format, normalizing pixel intensities, and verifying image consistency. Deepfake detection requires images to be uniformly processed so the CNN can analyze them effectively, and OpenCV ensures that every input is properly prepared for feature extraction.

- **NumPy**

NumPy is a fundamental dependency that facilitates numerical computations required throughout the deepfake detection process. It supports multi-dimensional array manipulation, matrix operations, reshaping of image tensors, and conversion of pixel values into numerical forms interpretable by the CNN. TensorFlow internally uses NumPy-style arrays, making NumPy integration critical for seamless data flow between preprocessing and model inference.

- **Streamlit**

Streamlit provides the interactive web interface through which users upload images and receive deepfake classification results. It simplifies UI development by enabling Python-based frontend construction without requiring HTML, CSS, or JavaScript. Streamlit automatically handles page rendering, file uploads, result display, and real-time updates.

- **Visual Studio Code (VS Code)**

Visual Studio Code serves as the main Integrated Development Environment (IDE) used to build, debug, and maintain the deepfake detection system. Its powerful features—such as IntelliSense autocompletion, built-in terminal, Git integration, Python debugging tools, and virtual environment management.VS Code supports extensions that simplify TensorFlow coding, automate formatting, and visualize file structures.

CHAPTER 7

SYSTEM IMPLEMENTATIONS

7.1 LIST OF MODULES

- Input Acquisition
- Preprocessing
- Feature Extraction
- Classification Model
- Result & Visualization

7.2 MODULE DESCRIPTION

The proposed AI-Powered Human Deepfake Detection System is designed using a structured, modular architecture where each component performs a specific function in the detection pipeline. The system operates in a sequential manner beginning from collecting the input image, preparing it for analysis, extracting important visual features, classifying the image using a trained deep learning model, and finally presenting the output to the user. Each module is designed to work independently while ensuring smooth data flow between stages. This modular design improves clarity, simplifies debugging, enhances performance, and allows future upgrades to be integrated without altering the overall system. The following sections describe each functional module in detail.

7.2.1 Input Acquisition Module

The Input Acquisition module serves as the starting point of the system and is responsible for obtaining the image data that will undergo deepfake verification. In this project, user interaction is made possible through a Streamlit-based web interface, where users can upload digital face images in common formats such as JPG, JPEG, and PNG. Once an image is uploaded, the system reads it as a byte stream and converts it into a numerical array using OpenCV functions, enabling it to be processed by the deep learning pipeline. This module also performs file validation to ensure that only supported image types are accepted, preventing system errors and maintaining

consistency in the input format. By collecting input in a controlled and standardized manner, the Input Acquisition module ensures that all subsequent processing steps operate on reliable and well-defined data, forming the basis for effective deepfake detection.

7.2.2 Preprocessing Module

The Preprocessing module converts the raw uploaded image into a standard format suitable for deep learning analysis. Since input images may vary in size, quality, and format, this stage ensures consistency before the model performs prediction. First, the uploaded image is resized to 96×96 pixels to match the input dimensions required by the MobileNetV2 model. The pixel values are then normalized from the range 0–255 to 0–1, improving numerical stability and allowing the model to process the image efficiently.

The color format is also converted from BGR to RGB, ensuring compatibility with the model's training configuration. In some cases, additional steps such as noise reduction or contrast adjustment may be applied to improve image clarity, especially for compressed or low-resolution images. Finally, the processed image is converted into a tensor format to make it ready for model inference.Through these steps, preprocessing ensures that all images are uniform, clean, and optimized, resulting in more accurate and reliable deepfake detection.

7.2.3 Feature Extraction Module

Feature Extraction is the core intelligence module of the system. It uses a pre-trained MobileNetV2 convolutional neural network, fine-tuned via transfer learning, to automatically learn discriminative visual features from face images. Instead of manually defining features, the network extracts hierarchical patterns, starting from low-level characteristics such as edges, textures, color gradients, and contours, and progressing to higher-level representations like facial symmetry, geometry, expression dynamics, lighting consistency, and GAN-induced artifacts.

These deep learned features help detect subtle irregularities commonly found in manipulated images, including unnatural smoothness, mismatched shadows, feature

misalignment, pixel-level blending errors, and abnormal reflections in the eyes or skin. MobileNetV2's depthwise separable convolution improves efficiency while preserving important spatial relationships, making the feature extraction process lightweight and suitable for real-time inference. The output of this module is a compact and meaningful numerical feature vector that summarizes the essential facial characteristics of the input image. This encoded representation is then passed to the classification module for final decision-making.

7.2.4 Classification Module

The Classification Model module receives the extracted feature vector and decides whether the image is Real or Fake. On top of MobileNetV2, fully connected layers along with dropout and batch normalization are added to improve generalization. The final layer uses a sigmoid activation function, producing a single probability value between 0 and 1, which represents the likelihood that the image is real. Binary cross-entropy is used as the loss function during training. If the probability exceeds a predefined threshold (e.g., 0.5), the image is classified as Real; otherwise, it is classified as Fake. This module is responsible for learning the decision boundary between genuine and manipulated images based on the patterns captured in the feature extraction stage.

7.2.5 Result and Visualizing Module

The Result and Visualization module presents the model's output to the end user in an understandable way. Through the Streamlit interface, the system displays the uploaded image along with the predicted label **"**Real**"** or **"**Fake**"** and the corresponding confidence score (e.g., percentage probability). Additionally, training performance graphs such as accuracy and loss curves are shown to illustrate how well the model has learned during training. Colors (e.g., green for real, red for fake) and descriptive text are used to make the result more intuitive. This module transforms raw model predictions into clear, interpretable information, enabling users, evaluators, and faculty to easily assess the reliability and effectiveness of the deepfake detection system.

CHAPTER 8

SYSTEM TESTING

8.1 UNIT TESTING

Unit testing is all that which usually involves in maintaining multiple designs within those of the available test cases within those of the available programming language which has many functional property, then each valid and invalid inputs are taken these are multiple branches, these testing are done using the individual software units within the applications, which can rely on the available knowledge in the construction and is important, unit testing performing basic test at components level and within the specific business processes, applications within the system configurations, within that of unit testing where all the documents are verified with the available process that performs the documents, that contains that clearly about the used inputs as well as the expected results.

8.2 SYSTEM TESTING

The system testing usually ensures in the entire integration which meets the software requirements, that usually are known and then they are predictable results, an example of the system testing which is oriented within these system that is based on the description as well as flows that emphasize the process links and the integration points.

8.2.1 White Box Testing

The white box testing in which the software testing has knowledgeable within the inner workings, software tester which is structure and within the language structure and also used to test the areas which cannot be reached from a black box levels.

8.2.2 Black Box Testing

The black box is tested without any prior knowledge within the inner workings of the structure or the language of those modules being tested within the black box tests that are if many kinds, within the written source of documents which is tested, that the software documents, responds to the output without considering the software.

8.3 PERFORMANCE TESTING

Performance testing was conducted to ensure the system operates efficiently and provides quick responses during image analysis. The prediction time after uploading an image was measured, and the system consistently produced results within a short response window. The model was tested using images of different resolutions and file sizes to evaluate scalability and responsiveness. The system was also run multiple times continuously to check for delays, memory issues, or performance drops during repeated execution. The results confirmed that the deepfake detection system is lightweight, fast, and stable, making it suitable for real-time applications and practical deployment.

8.4 SECURITY TESTING

Security testing was performed to ensure that the system is protected from unauthorized access, data misuse, and malicious inputs. Since the application allows users to upload external images, validation checks were included to prevent harmful or unsupported file types from being processed. The system was tested against potential threats such as script injection, file tampering, and corrupted image uploads to ensure safe execution.User inputs were monitored to verify that only image formats are accepted, and no executable or harmful content can enter the system.

CHAPTER 9

RESULT AND DISCUSSION

The AI-Powered Human Deepfake Detection System, developed using the MobileNetV2 architecture, demonstrated strong capability in distinguishing real and manipulated human facial images. Throughout model evaluation, MobileNetV2 efficiently extracted discriminative facial features and identified subtle deepfake artifacts such as texture inconsistencies, unnatural smoothness, lighting mismatches, and GAN-generated distortions. The system remained stable even under variations like compression, resizing, and minor visual noise, showing good generalization across multiple image conditions. Confidence scores produced during testing closely matched the ground truth labels, indicating that the model made consistent and reliable predictions. The detection pipeline—including preprocessing, feature extraction, and classification—worked cohesively, proving that lightweight architectures can deliver high accuracy without requiring heavy computational resources.

Real-time testing through the Streamlit interface further validated the system's practicality. Images uploaded by users were processed smoothly, generating predictions within just a few seconds. The interface displayed the input image, classification label, and confidence score, making the system intuitive and accessible for both technical and non-technical users. Stability was confirmed through repeated inference runs, where the model maintained consistent accuracy and response time. Training and validation graphs indicated balanced learning with no significant overfitting, supported by techniques such as data augmentation and batch normalization. Overall, the system successfully met its objective of providing an efficient, accurate, and user-friendly deepfake detection mechanism. These results highlight the system's strong potential for real-world applications in media verification, digital forensics, cybersecurity, and academic research.

CHAPTER 10

CONCLUSION AND FUTURE ENHANCEMENTS

10.1 CONCLUSION

The AI-Powered Human Deepfake Detection System was successfully developed to identify manipulated human facial images using deep learning and image processing techniques. By integrating the MobileNetV2 architecture with transfer learning, the system effectively extracts meaningful facial features and identifies deepfake artifacts that are not easily visible to the human eye. The preprocessing pipeline, lightweight feature extraction network, and efficient classification module work together to deliver accurate predictions with minimal computational overhead.

The implementation of Streamlit as the user interface ensures that the detection process remains simple, fast, and accessible for both technical and non-technical users. Experimental results confirmed that the system performs reliably across various real and fake images, maintaining stable accuracy and fast response times. The confidence scores provided by the model further enhance transparency and user trust.

Overall, the project accomplishes its objective of providing a practical, efficient, and user-friendly system capable of detecting deepfake images with high accuracy. It demonstrates the capability of deep learning models to contribute to digital media authenticity, cybersecurity, and forensic analysis in a world where AI-generated content is rapidly evolving.

10.2 FUTURE ENHANCEMENTS

The current system focuses solely on detecting deepfakes in static facial images. As a future extension of this work, the system can be expanded to support video-based deepfake detection, which offers a more comprehensive and practical solution for real-world scenarios. Video deepfakes often involve complex frame-by-frame manipulations, inconsistencies in facial movements, unnatural blinking patterns, temporal distortions, and mismatched audio–visual synchronization. By incorporating

video analysis, the system would be capable of identifying these temporal artifacts that cannot be detected in single images.

To achieve this, each frame of the video can be extracted and processed sequentially using advanced deep learning models. Temporal learning architectures such as Long Short-Term Memory (LSTM) networks, 3D Convolutional Neural Networks (3D-CNN), or transformer-based video models can be used to analyze motion patterns and frame transitions. Integrating optical flow analysis can further enhance accuracy by capturing unnatural movement dynamics typically found in deepfake videos. The system can also aggregate predictions from multiple frames to generate a final authenticity score for the entire video.

This enhancement will enable the model to detect more sophisticated deepfakes, making it suitable for applications such as social media monitoring, digital forensics, online authentication, and content verification. By expanding the system to handle video inputs, the project will evolve into a robust, real-time deepfake detection framework capable of addressing modern digital security challenges more effectively.

APPENDIX – A

SOURCE CODE

app.py

```
import streamlit as st
import numpy as np
import cv2
from tensorflow.keras.models import load_model

# Page settings
st.set_page_config(page_title="DeepFake Detector", page_icon="🕵️", layout="centered")

# ---------- Custom UI Theme (Peach + Pink) ----------
st.markdown("""
   <style>
      body {
         background-color: #FFE5E5;
      }
      .main {
         background: linear-gradient(135deg, #FFD1DC, #FFE6CC);
         padding: 2rem;
         border-radius: 20px;
      }
      .stButton>button {
         background-color: #FF8FA6;
         color: white;
         border-radius: 12px;
         font-size: 18px;
         font-weight: bold;
         padding: 0.6rem 1.4rem;
         border: none;
      }
      .stButton>button:hover {
         background-color: #FF6787;
         color: white;
      }
      .result-box {
         padding: 15px;
```

```
            border-radius: 12px;
            text-align: center;
            font-weight: bold;
            font-size: 22px;
        }
    </style>
""", unsafe_allow_html=True)

# Title
st.markdown("<h1 style='text-align:center;color:#FF577F;'>DeepFake Image
Detector</h1>", unsafe_allow_html=True)
st.markdown("<p style='text-align:center;color:#FF6F91;'>Upload an image & verify
authenticity within seconds </p>", unsafe_allow_html=True)

# ---------- Load model (safe) ----------
@st.cache_resource
def load_deepfake_model():
    try:
        return load_model('deepfake_detection_model.h5')
    except Exception as e:
        st.error(f"Model load failed: {e}")
        return None

model = load_deepfake_model()

# ---------- Preprocessing ----------
def preprocess_image(image_bgr):
    img = cv2.resize(image_bgr, (96, 96))
    img = img.astype("float32") / 255.0
    return np.expand_dims(img, axis=0)
# ---------- Prediction ----------
THRESH = 0.50
def predict_image(image_bgr):
    x = preprocess_image(image_bgr)
    prob_real = float(model.predict(x, verbose=0)[0][0])
    label = "Real " + " if prob_real >= THRESH else "Fake +"
    return label, prob_real

# ---------- Upload Section ----------
uploaded_file = st.file_uploader("Upload Image", type=["jpg", "jpeg", "png"])
```

```
if uploaded_file is not None:
    file_bytes = np.asarray(bytearray(uploaded_file.read()), dtype=np.uint8)
    image_bgr = cv2.imdecode(file_bytes, cv2.IMREAD_COLOR)

    st.image(image_bgr, channels="BGR", caption="Uploaded Image",
use_column_width=True)

    if model is None:
        st.stop()

    if  st.button("Analyze
        with st.spinner("Analyzing, please wait...
            label, prob_real = predict_image(image_bgr)

        # ◆ Color-coded results
        color = "#4BB543" if "Real" in label else "#FF4B4B"

        st.markdown(
            f"""
            <div class='result-box' style='background-color:{color}; color:white;'>
                {label}<br>
                Confidence: {prob_real:.2f}
            </div>
            """,
            unsafe_allow_html=True
        )

st.markdown("---")
st.caption("This is a demo classifier. Results may vary on unseen manipulations.")
```

Train.py

```
from pathlib import Path
import tensorflow as tf
from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout,
BatchNormalization
```

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
# ---- Paths ----
BASE_DIR = Path(_file_).resolve().parent.parent  # project root
TRAIN_DIR = BASE_DIR / "Dataset" / "Dataset" / "Train"
VAL_DIR  = BASE_DIR / "Dataset" / "Dataset" / "Validation"
MODEL_OUT = BASE_DIR / "deepfake_detection_model.h5"
# ---- Hyperparams ----
BATCH = 64
EPOCHS = 1
STEPS = 150
VAL_STEPS = 60
def build_and_train():
    print("Train dir :", TRAIN_DIR)
    print("Val dir  :", VAL_DIR)
    # Data generators
    train_gen = ImageDataGenerator(rescale=1./255, horizontal_flip=True)
    val_gen  = ImageDataGenerator(rescale=1./255)
    train = train_gen.flow_from_directory(
        TRAIN_DIR, target_size=(96, 96), batch_size=BATCH, class_mode="binary")
    val = val_gen.flow_from_directory(
        VAL_DIR, target_size=(96, 96), batch_size=BATCH, class_mode="binary")
    # Model
    base = MobileNetV2(include_top=False, weights="imagenet", input_shape=(96, 96,
3))
    base.trainable = False
    model = Sequential([
        base,
        GlobalAveragePooling2D(),
        Dense(256, activation="relu"),
        BatchNormalization(),
```

```
        Dropout(0.3),
        Dense(1, activation="sigmoid")
    ])
    model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
    model.summary()
    # Callbacks
    ckpt = ModelCheckpoint(MODEL_OUT, monitor="val_accuracy",
save_best_only=True, verbose=1)
    es = EarlyStopping(monitor="val_accuracy", patience=2,
restore_best_weights=True)
    # IMPORTANT: workers=1 & multiprocessing False (WINDOWS FIX)
    history = model.fit(
        train,
        epochs=EPOCHS,
        steps_per_epoch=STEPS,
        validation_data=val,
        validation_steps=VAL_STEPS,
        workers=1,
        use_multiprocessing=False,
        callbacks=[ckpt, es]
    )
    print("\n Model training
    finished!") print(f" Model saved at:
{MODEL_OUT}") if _name == "_main_":
    build_and_train()
```

Predict.py

```
import numpy as np
import cv2
import os
from tensorflow.keras.models import load_model
```

```
from tensorflow.keras.preprocessing.image import img_to_array

# Load the trained model
model = load_model('deepfake_detection_model.h5')

# Preprocess the image
def preprocess_image(image_path):
    image = cv2.imread(image_path)
    image = cv2.resize(image, (96, 96))
    image = img_to_array(image)
    image = np.expand_dims(image, axis=0)
    image = image / 255.0
    return image

# Predict if the image is fake or real
def predict_image(image_path):
    image = preprocess_image(image_path)
    prediction = model.predict(image)
    class_label = np.argmax(prediction, axis=1)[0]
    return "Fake" if class_label == 0 else "Real"

# Example usage
image_path =
"real_and_fake_face_detection/real_and_fake_face/training_real/real_00001.jpg"
result = predict_image(image_path)
print(f"The image is {result}")
```

APPENDIX – B

SCREENSHOTS

Sample Output

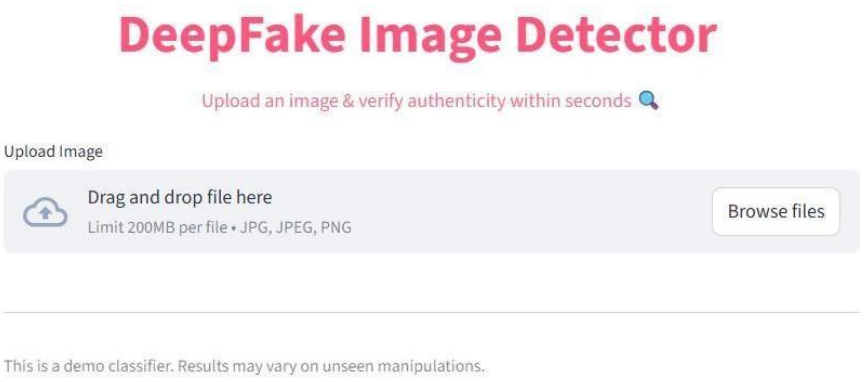

Figure. B.1. Login Page

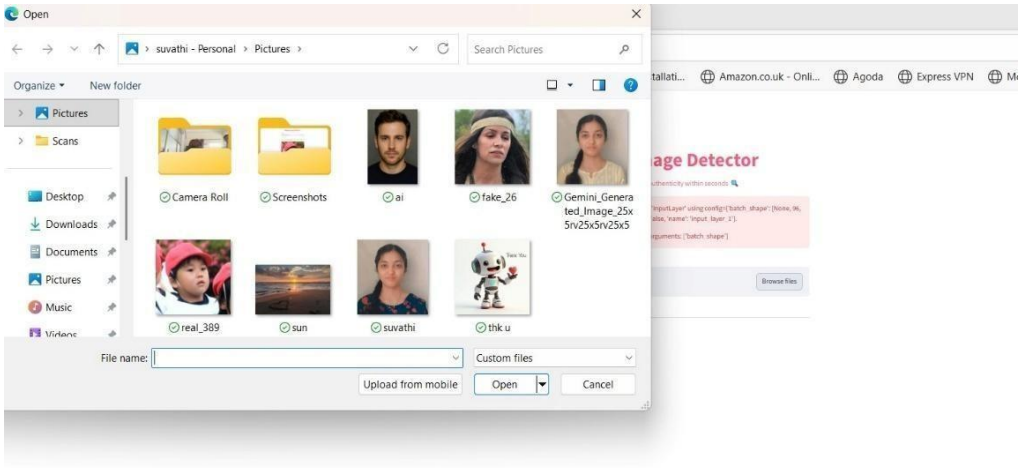

Figure. B.2. Image Uplaod Page

Figure. B.3. Result Page

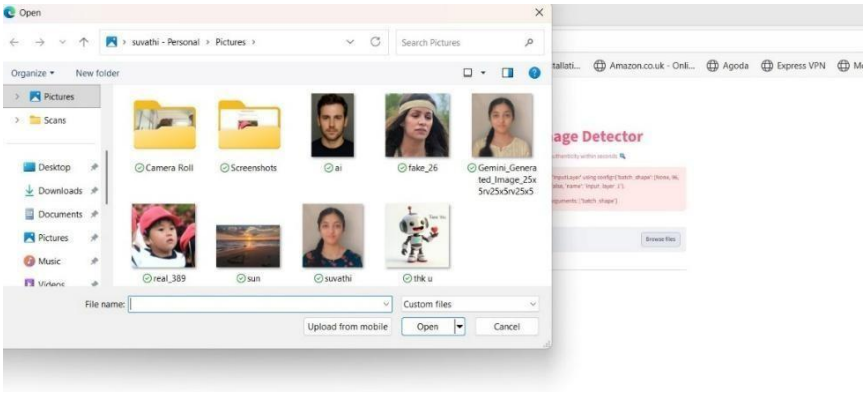

Figure. B.4. Image Uplaod Page

Figure. B.3. Result Page

REFERENCES

1. Agarwal, Shruti, Hany Farid, Yuming Gu, Mingming He & Koki Nagano 2019, 'Protecting World Leaders Against Deep Fakes', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 38–45.

2. Afchar, Darius, Vincent Nozick, Junichi Yamagishi & Isao Echizen 2018, 'MesoNet: A Compact Facial Video Forgery Detection Network', IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7.

3. Amerini, Irene, Luca Galteri, Roberto Caldelli & Alberto Del Bimbo 2019, 'Deepfake Video Detection through Optical Flow Based CNN', Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), pp. 1205–1213.

4. Bayar, Belhassen & Matthew C. Stamm 2016, 'A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer', Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, pp. 5–10.

5. Bunk, Jonathan, Jamie Roycroft, Shruti Agarwal, Hany Farid, Laura Garrido & Javier F. C. Morillo 2020, 'Detection of Deepfake Videos Using Multi-Scale Biological Signals', Proceedings of the IEEE CVPR Workshops, pp. 1–10.

6. Dolhansky, Brian, Russ Howes, Ben Pflaum, Nicole Baram & Cristian Canton-Ferrero 2020, 'The DeepFake Detection Challenge Dataset', arXiv preprint arXiv:2006.07397, pp. 1–10.

7. Güera, David & Edward J. Delp 2018, 'Deepfake Video Detection Using Recurrent Neural Networks', Proceedings of the IEEE AVSS, pp. 1–6.

8. Haliassos, Alexandros, Stavros Moschoglou, Stylianos Ploumpis & Stefanos Zafeiriou 2021, 'Lips Don't Lie: A Generalisable and Robust Approach To Face

Forgery Detection', Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5039–5049.

9. Hwang, Sunmee, Younghyun Choi, Jaewon Kim & Jongwon Choi 2021, 'Deepfake Detection Using Attention Mechanisms and Residual Noise Learning', IEEE Access, vol. 9, pp. 128254–128264.

10. Jeon, In Kyu, Minjung Kim & Sanghoon Lee 2020, 'FDFtNet: Facing Deepfake Detection with Fourier Transform', ICIP, pp. 251–255.

11. Li, Yuezun, Ming-Ching Chang & Siwei Lyu 2018, 'In Ictu Oculi: Exposing AI-Created Fake Videos Using Eye-Blink Detection', IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6.

12. Li, Yuezun, Pu Sun, Honggang Qi & Siwei Lyu 2020, 'Celeb-DF: A Large-Scale Challenging Dataset for Deepfake Forensics', Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3207–3216.

13. Matern, Florian, Christian Riess & Marc Stamminger 2019, 'Exploiting Visual Artifacts to Expose Deepfakes and Face Manipulations', IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 83–92.

14. Mittal, Tanya, Uttaran Bhattacharya, Ruchit Rawal, Abhinav Dhall & Ramanathan Subramanian 2020, 'Emotions Don't Lie: Deepfake Detection Framework', *ACM Multimedia*, pp. 2823–2832.

15. Nguyen, Huy H., Junichi Yamagishi & Isao Echizen 2019, 'Capsule-Forensics: Using Capsule Networks to Detect Forged Images and Videos', IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2307–2311.

16. Rossler, Andreas, Davide Cozzolino, Luisa Verdoliva, Christian Riess & Matthias Nießner 2019, 'FaceForensics++: Learning to Detect Manipulated Facial Images', IEEE International Conference on Computer Vision (ICCV), pp. 1–11.

17. Sabir, Ekraam, Weilin Li, Lav R. Varshney, Pramod K. Varshney & David Schonfeld 2019, 'Recurrent Convolutional Strategies for Face Manipulation Detection in Videos', IEEE Transactions on Information Forensics and Security, vol. 14, no. 11, pp. 2798–2813.

18. Tariq, Salman, Seema Nagar, Piyush Gupta & Rama Chellappa 2021, 'Generalized Deepfake Detection', *CVPR*, pp. 15153–15162.

19. Verdoliva, Luisa 2020, 'Media Forensics and DeepFakes: An Overview', IEEE Signal Processing Magazine, vol. 37, no. 1, pp. 86–96.

20. Wang, Sheng-Ying, Oliver Wang, Richard Zhang, Andrew Owens & Alexei A. Efros 2020, 'CNN-Generated Images Are Surprisingly Easy to Spot... for Now', IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8692–8701.

21. Wang, Xiaoyu, Shuning Jiang & Peng Bing 2020, 'Feature Pyramid Encoding for Detecting Deepfakes', *Neurocomputing*, pp. 302–313.

22. Wu, Haodong, Jianhua Yang, Hong Liu & Jing Wang 2022, 'Deepfake Detection Using Multi-Region Attention and Noise Modeling', Pattern Recognition, vol. 131, pp. 108–118.

23. Zhang, Yinglin, Peng Zhou, Wei Wang, Keren Chen & Weixuan Wu 2020, 'Detecting Deepfake Videos with Temporal Dropout Learning', Pattern Recognition Letters, vol. 138, pp. 248–254.

24. Zhou, Peng, Xiaodong Li, Jiahong Yuan & Weixuan Wu 2018, 'Two-Stream Neural Networks for Tampered Face Detection', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–8.

25. Zi, Bairui, Ruihai Wu, Yujie Qi & Hongyuan Zhu 2020, 'WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection', Proceedings of the ACM International Conference on Multimedia, pp. 1470–1478.