Enhancing Deepfake Detection Accuracy and Generalization Through a Hybrid LBP-CNN Approach

Dr. M. Swapna
Assistant Professor
Computer Science and Engineering Department
Matrusri Engineering College,
Saidabad, Hyderabad, Telangana, India

Abstract - **In response to the growing concerns regarding the spread of deepfake media generated by advanced deep learning techniques, ensuring digital content authenticity has become a critical challenge. Existing detection systems often depend solely on either spatial or temporal features, resulting in limited accuracy, poor generalization to unseen data, and difficulty in real-time application. To address these shortcomings, this study proposes a hybrid deepfake face detection framework that combines Local Binary Patterns (LBP) and Convolutional Neural Networks (CNNs). LBP captures fine-grained texture inconsistencies, whereas CNNs extract deep spatial features, thereby enhancing the detection performance. The system was deployed using a Flask-based web interface for real-time analysis and visualization. When trained on benchmark datasets such as FaceForensics++, DFDC, and Celeb-DF, the model achieves high accuracy, robustness, and generalizability. This approach offers a more reliable and efficient solution for safeguarding the integrity of digital media.**

Keywords - Deepfake detection, Local Binary Pattern (LBP), Convolutional Neural Networks (CNN), Flask, FaceForensics++, digital forensics.

I. INTRODUCTION

Deepfake technologies have rapidly evolved into a major concern regarding digital media authenticity. Utilizing advanced deep learning techniques, such as Generative Adversarial Networks (GANs), malicious actors can create highly convincing fake images and videos, especially of human faces. These synthetic media challenge the credibility of digital content and have serious implications in domains such as politics, journalism, cybersecurity, and social media. As these technologies become more accessible, the risk of misinformation and identify manipulation continues to grow

Existing deepfake detection systems primarily rely on deep learning models that extract spatial and temporal features. While effective in controlled environments, they face key limitations, such as poor generalization to novel deepfake techniques, sensitivity to image and video compression, and high computational requirements. Many of these systems are also not accessible for real-time use or lack a user-friendly interface, making them impractical for widespread forensic or public applications. To overcome these challenges, this study presents a hybrid deepfake detection framework that combines the strengths of Local Binary Patterns (LBP) and Convolutional Neural Networks (CNNs). LBP captures micro-texture inconsistencies that often go undetected by standard CNNs, whereas CNNs extract deep spatial features to effectively distinguish real content from manipulated content. The integration of these two techniques enhances the detection accuracy and robustness against diverse deepfake generation methods.

The system was deployed through a Flask-based web interface, enabling real-time face image input, analysis, and result visualization. Users can effortlessly upload facial images and receive immediate feedback on whether the content is real or manipulated, making the tool highly practical for applications in digital forensics, law enforcement, media verification, and cybersecurity investigations. The intuitive interface ensures usability for both technical and non-technical users, increasing accessibility across various sectors.

Trained on large-scale datasets such as FaceForensics++, DFDC, and Celeb-DF, the model ensures high precision and robustness across varying conditions, including lighting, orientation, and compression. By combining LBP for micro-texture analysis with CNNs for spatial feature extraction, subtle deepfakes can be effectively detected. Its lightweight architecture supports deployment on standard hardware and integration into forensic and public platforms. The system also allows continuous learning, ensuring adaptability to emerging deepfake techniques and promoting a trustworthy digital media environment.

I. Objectives

A robust deepfake detection system is presented, which combines Convolutional Neural Networks (CNNs) and Local Binary Patterns (LBP) to extract spatial and texture features from facial images. Integrated into a Flask-based web interface, it enables real-time and user-friendly verification. Trained on diverse benchmark datasets, the model ensures high accuracy, scalability, and generalization across various deepfake techniques.

II. Scope

This system focuses on classifying individual facial images, emphasizing the high accuracy detection of various manipulation techniques. Designed for

adaptability, it is well suited for real-world applications such as journalism verification, forensic analysis, and content moderation platforms.

II. LITERATURE REVIEW

In recent years, a wide range of methods have been proposed for detecting deepfakes, focusing on spatial, temporal, and frequency-based inconsistencies introduced during manipulation. Key approaches include :

[1] The study "Face X-ray for Deepfake Detection" used a two-stream ResNet-based model to identify manipulated regions in facial images. Although effective on known data, it showed poor generalization to deepfakes from unseen techniques or datasets. [2] "MesoNet" introduced a lightweight CNN model targeting low-level inconsistencies in facial videos. Despite its efficiency, it struggles with high-quality, compressed videos and often misses subtle deepfakes.

[3] "Exposing Deep Fakes Using Inconsistent Head Poses" relied on facial landmarks and geometric analyses. While effective for unnatural poses, its accuracy decreases when deepfakes use corrected poses or stable angles. [4] "Detecting Deepfake Videos Using Biological Signals" leveraged PPG signals to spot heart rate inconsistencies. However, it requires high-resolution input and fails under low light, noise, or compression, which are common on social platforms.

[5] The paper "Multi-task Learning for Deepfake Detection via AudioVisual Inconsistencies" presented a method that jointly analyzed lipsync errors and visual cues using a multi-modal approach. Although effective in detecting audio-visual mismatches, it cannot be applied to images or silent videos, and its performance degrades in cases where audio is absent or intentionally well-synchronized. The proposed hybrid deepfake detection framework overcomes these limitations by integrating Local Binary Patterns (LBP) to capture subtle texture inconsistencies and Convolutional Neural Networks (CNNs) for deep spatial feature extraction. Unlike single-modality or shallow-feature methods, this system is robust across various resolutions, compression levels, and manipulation types. Additionally, the realtime, Flask-based interface ensures usability and accessibility, making it suitable for practical deployment in diverse digital forensic scenarios.

III. SYSTEM ARCHITECTURE

The architecture of the proposed deepfake detection system comprises a modular, layered design that ensures both functional robustness and operational efficiency. The system was developed to facilitate high-performance image classification and real-time user interaction while preserving data privacy and integrity through localized processing and a secure architecture.

A. Core Architectural Layers:

1. Preprocessing Module:

Before any analysis, all input images underwent a standardized preprocessing stage to enhance the quality and consistency of feature extraction. This module begins with face detection, utilizing established algorithms such as Haar Cascades or Dlib to accurately localize and crop facial regions from the input image. Once detected, the faces were aligned to a canonical orientation, which ensured consistency across the dataset and improved the robustness of the subsequent layers. This is followed by normalization, where the pixel intensity values are scaled, and the images are resized to a fixed resolution, making them compatible with the input specifications of the neural network. To promote generalization and reduce overfitting during training, data augmentation techniques, such as brightness variation, rotation, and flipping, may be applied.

2. Feature Extraction Unit:

This unit is responsible for extracting both spatial and texture-based features from facial regions using a dual-pathway approach. The first pathway involves the Local Binary Pattern (LBP) module, which captures local texture descriptors by thresholding the neighboring pixels. This results in an efficient encoding of micro-patterns, often indicative of synthetic content, such as uniform textures or unnatural transitions. Simultaneously, the second pathway employs Convolutional Neural Network (CNN) layers to hierarchically extract spatial features through the application of learnable filters and pooling operations. These layers are particularly adept at identifying deepfake-specific anomalies, such as blending inconsistencies and unnatural facial textures. The outputs from both the LBP and CNN pathways were then concatenated or fused to form a comprehensive feature representation, which was forwarded to the classification layer.

3. Classification Layer:

The classification layer serves as the decision-making engine for the system. The concatenated feature map from the previous unit is first flattened and passed through one or more fully connected layers, which distill high-dimensional features into a compact representation. The final classification is performed using an output activation layer sigmoid for binary classification (real vs. fake) or SoftMax for potential multi-class extension. This layer generates a probability score that indicates the likelihood that the input face is authentic or synthetically generated.

4. Web Interface Layer (Flask Framework)

To provide accessibility and real-time interaction, the trained model was deployed through a user-friendly web interface developed using the Flask micro-framework. This interface allows users to upload facial images via a

web browser and receive instant classification results, including the prediction and confidence score. Optionally, techniques such as Grad-CAM or attention visualization may be used to render localized heatmaps, offering insights into the regions that most influenced the model's decision. Flask handles HTTP requests, routes inputs to the backend model, and returns results efficiently, enabling a seamless and interactive user experience.

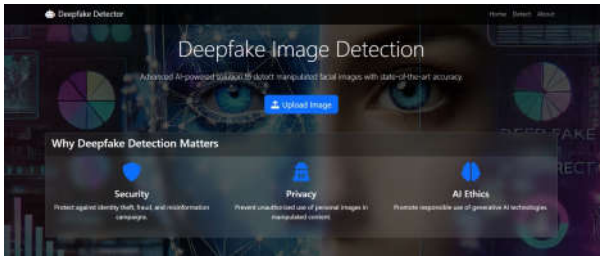

Figure 1: Input Stage: Facial Image Selection

B. Security and Data Integrity

To protect user privacy, maintain trust, and ensure system resilience against misuse, several security and data integrity measures are embedded in the system design.

1. Access Control:

The system integrates authentication mechanisms to ensure that only authorized and registered users can interact with the model. For institutional or forensic deployments, these controls can be extended with role-based access control (RBAC), which provides granular privileges based on user roles.

2. Local Image Processing

In the interest of privacy, all uploaded images are processed in memory during the user session and are not persisted on the server or stored in any database. This ensures that sensitive biometric data are not retained after classification and that users maintain full control over their media inputs.

3. Data Anonymity

The system is designed to operate independently of any personally identifiable information (PII). Uploaded images are not linked to any user data, ensuring compliance with data protection standards, such as the General Data Protection Regulation (GDPR).

4. Tamper-Resistant Operation

The system integrity is preserved through a read-only operational mode during inference. No modifications to the user data or model were made during the runtime. Although logs may be maintained for debugging purposes, they are explicitly designed to exclude any user media. For future versions, cryptographic hash verification can be employed to ensure the integrity of the model and code.

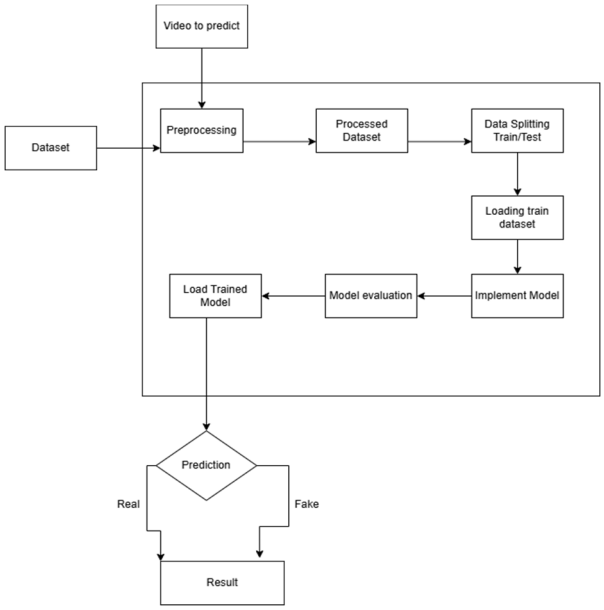

Figure 2: System Architecture

5. Scalability and Modularity

The entire system architecture is modular and scalable, allowing individual components such as the CNN backbone, preprocessing pipeline, or web interface to be upgraded or replaced without disrupting the system's functionality. This ensures adaptability to emerging threats, new deepfake techniques, and evolving detection algorithms.

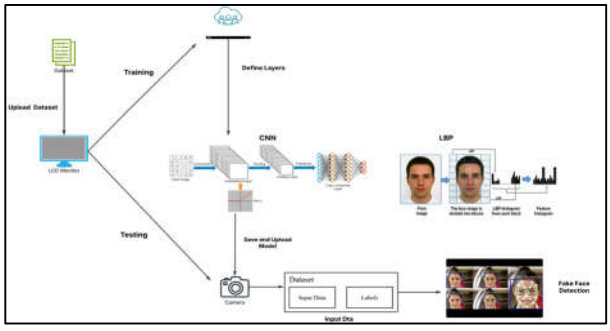

Figure 3: System Architecture of Deep Fake Face Detection

IV. PROPOSED METHODOLOGY

The proposed system presents a hybrid image-based approach for detecting deepfake facial forgeries by leveraging the strengths of **Local Binary Patterns (LBP)** for texture encoding and **Convolutional Neural Networks (CNNs)** for hierarchical feature learning. The core objective of this methodology is to extract both low-level and high-level visual inconsistencies commonly introduced during face synthesis using deep learning methods such as GANs. The system was designed for static image analysis and deployed via a lightweight Flask-based web interface for practical usability.

The methodology is divided into several key components:

A. Local Binary Pattern (LBP)

The Local Binary Pattern (LBP) is a widely used textue descriptor that effectively captures local structural information within an image. It operates by comparing the intensity of each pixel with that of its surrounding neighbors within a defined radius. Based on these comparisons, binary values are assigned typically '1' if a neighbor's intensity is greater than or equal to the central pixel, and '0' otherwise. The resulting binary patterns are then transformed into decimal representations, forming a compact yet powerful encoding of the local texture. These descriptors are particularly useful for identifying fine-grained variations, such as irregular textures, unnatural edges, and noise, which are often indicative of tampering or image manipulation.

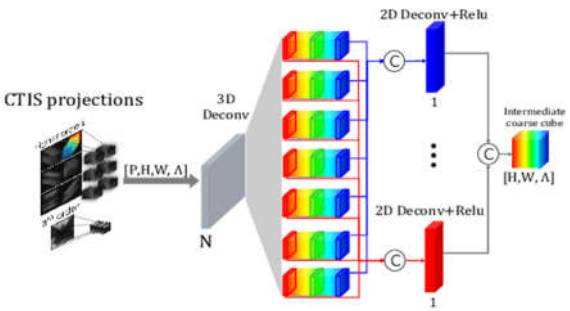

Figure 4: Local Binary Pattern Architecture

In the proposed deepfake detection system, the LBP operator is applied to grayscale facial images after the alignment and normalization stages during preprocessing. This ensures consistency in the input and maximizes the reliability of the extracted texture features. The resulting LBP codes were aggregated into histograms that served as interpretable and discriminative texture descriptors. These histograms were then used as inputs to the classification module, either individually or in conjunction with the CNN-derived spatial features. Notably, the LBP enhances the system's sensitivity to pixel-level anomalies, especially in critical facial regions, such as the eyes, mouth, and skin areas, which deepfake generation algorithms frequently fail to synthesize accurately. Thus, the LBP significantly contributes to the system's ability to detect subtle artifacts and improves the overall classification performance.

B. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are integral to the proposed system for learning complex spatial features from facial images in a supervised learning framework. The CNN architecture was designed to hierarchically extract visual representations that highlight both low- and high-level attributes relevant to deepfake detection. The network begins with multiple convolutional layers that apply learnable filters to the input image to detect fundamental visual elements, such as edges, corners, and textures. This is followed by pooling layers, which progressively reduce the spatial resolution while preserving the most salient features, thereby improving the computational efficiency and reducing the risk of overfitting.

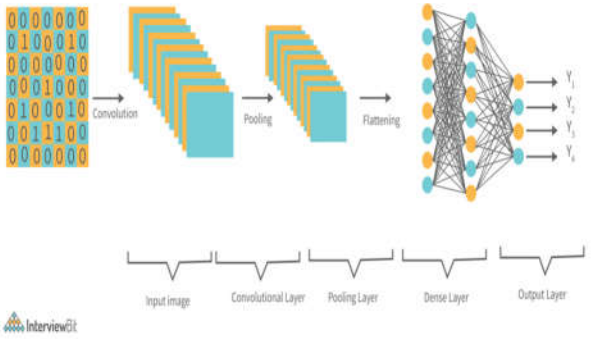

Figure 5: Convolutional Neural Network (CNN)

To further enhance the model's robustness and generalization capability, dropout layers were incorporated to prevent the coadaptation of neurons, whereas batch normalization layers stabilized and accelerated the training process by normalizing the output of the preceding layers. The final fully connected layers act as a high-level classifier, integrating the extracted spatial features to perform binary classification and distinguishing between real and fake faces.

This CNN-based feature extraction process complements the Local Binary Pattern (LBP) module by focusing on global and abstract features, such as facial symmetry distortions, unnatural blending artifacts, and inconsistencies in lighting characteristics, which are commonly present in synthetically generated images. Together, the CNN and LBP components provide a robust dual-stream architecture capable of capturing both localized texture anomalies and broader structural inconsistencies for effective deepfake detection.

C. Feature Fusion and Classification Process

To enhance the discriminative power of the proposed deepfake detection system, features extracted from the Local Binary Pattern (LBP) and Convolutional Neural Network (CNN) modules are fused at the feature level. This fusion is typically achieved through concatenation or ensemble-based logic, enabling the integration of both fine-grained texture descriptors and high-level semantic spatial features. By combining the strengths of both local and global feature representations, the system can detect a wide range of deepfake artifacts, from subtle pixel-level inconsistencies to broader structural anomalies.

This hybrid feature representation is then passed to the classification head, which consists of fully connected layers culminating in a sigmoid-activated output layer

for binary classification purposes. The sigmoid function outputs a probability score that quantifies the likelihood of a face being fake. A predefined decision threshold was applied to this probability to determine the final classification label. Importantly, this threshold can be fine-tuned depending on the deployment context; for instance, can be optimizeding for higher recall in forensic investigations where a deepfake is critical, or higher precision in real-time applications to reduce false alarms. This flexible classification framework contributes significantly to the overall robustness and adaptability of the system across diverse use cases and data sets.

D. Dataset Overview

The dataset used for training and evaluating the proposed system consisted of over 18,000 high-quality facial images extracted from benchmark deepfake datasets, such as FaceForensics++, DFDC, and Celeb-DF. These images encompass a diverse range of genuine and manipulated faces, covering various ages, ethnicities, lighting conditions, facial expressions, and image qualities.

By utilizing a large volume of individual images rather than entire videos, the system focuses on extracting detailed spatial and texturebased features from each frame. This approach allows for the precise analysis of subtle local texture inconsistencies, captured effectively by Local Binary Patterns (LBP), alongside deep spatial features extracted by Convolutional Neural Networks (CNNs).

The extensive size and diversity of the dataset ensured that the model generalized well across different deepfake generation methods and real-world variations, improving its robustness against compression artifacts, noise, and various manipulation styles. This image-based training strategy supports efficient real-time detection while maintaining high accuracy and reliability in identifying deepfake facial content.

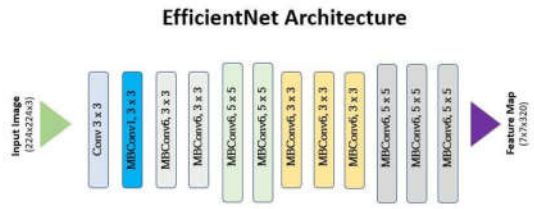

Figure 6: EfficientNet Architecture

E. Web Deployment Using Flask

To bridge the gap between research and real-world usability, the proposed deepfake detection system was deployed through a webbased interface utilizing Flask, a lightweight Python micro web framework. This deployment enables interactive and user-friendly access to the detection capabilities of the underlying model. The web interface facilitates image upload functionality, allowing users to submit facial images directly through a browser. Once an image is uploaded, the system performs real-time classification, typically within 1–2 s, providing immediate feedback on the authenticity of the face. The interface prominently displays the classification result, indicating whether the input face is real or fake, along with a confidence score (e.g., 91.3% confidence fake), enhancing the transparency and interpretability of the model's output. For added security and data privacy, all uploaded images are handled in memory and are not stored on disk or retained post-processing, ensuring session isolation and safeguarding sensitive biometric information

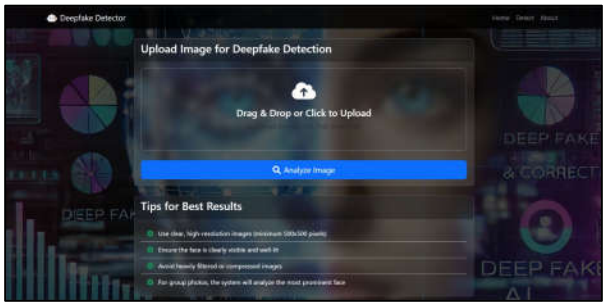

Figure 7: Input Stage: Facial Image Selection.

The lightweight and modular nature of the Flask framework ensures seamless integration with the backend model and supports flexible deployment across various platforms, including local systems, cloud-based environments, and edge devices. This implementation demonstrates the practicality, responsiveness, and adaptability of the system for real-time applications in forensic, institutional, and consumer-facing settings.

V. IMPLEMENTATION AND RESULTS

The implementation of the proposed deepfake detection system involved the careful integration of preprocessing techniques, model training, dataset preparation, and interface deployment. The goal was to create a lightweight, high-accuracy solution capable of efficiently detecting manipulated facial images efficiently in real time.

A. Datasets

The proposed deepfake detection system was implemented using three benchmark datasets, each chosen for its unique contribution to the system's robustness and generalization. The FaceForensics++ dataset served as the primary source for extracting both real and forged facial images from compressed video samples. This dataset is particularly useful because of its inclusion of multiple forgery techniques, such as Face Swap and Deep Fakes, under controlled yet varied scenarios. To introduce broader diversity, the Deep Fake Detection Challenge (DFDC) dataset created by Facebook AI was utilized. It provided high-resolution facial images spanning different demographic groups, lighting conditions, backgrounds, and facial poses, enabling the model to generalize beyond narrow data

distributions. The third dataset, Celeb-DF (v2), contains high-quality manipulated images with minimal visual artifacts. This dataset was instrumental in assessing the model's ability to detect subtle manipulations and verify its robustness against sophisticated deepfakes. Notably, all datasets were processed to extract static facial images for classification, explicitly excluding video-based temporal information.

B. Preprocessing Pipeline

A consistent preprocessing pipeline was applied across all datasets to ensure high-quality and uniform input data. The images were first extracted and resized to 224×224 pixels using OpenCV. Face regions were detected and cropped using Haar Cascade or Dlib's HOG-based detector. The pixel values were normalized to the [0, 1] range for training stability. To enhance generalization and reduce overfitting, data augmentation techniques such as horizontal flips, random rotations, brightness/contrast adjustments, zooms, and spatial shifts were applied. This process ensured a diverse, consistent, and robust dataset for the model training.

C. Model Training

The core detection model was built using TensorFlow/Keras in Python and trained on preprocessed datasets via supervised learning. The architecture features multiple convolutional and max-pooling layers to extract hierarchical spatial features, along with dropout and batch normalization to prevent overfitting and enhance training stability. The final classification was achieved using fully connected layers with a sigmoid activation function for binary output. The model used the Adam optimizer with a dynamic learning rate scheduler and a binary cross-entropy loss. Training was performed with a batch size of 32 for 25–50 epochs, employing early stopping based on the validation loss to maintain generalization.

D. Results and Performance

The performance of the trained model was rigorously evaluated on a reserved test set using several standard classification metrics. The system demonstrated an accuracy of approximately 93–95% across unseen images, indicating a strong generalization. The F1-score exceeded 0.92, suggesting a balanced performance between precision and recall. The ROC-AUC score, which measures the model's confidence across various thresholds, reached 0.96, affirming the system's ability to reliably distinguish between real and fake facial images. Beyond the technical metrics, the model was integrated into a Flask-based web interface, enabling real-time detection with an average response time of less than 2 s per image. The interface provides instant visualization of the classification results and detection probabilities, offering a smooth user experience and demonstrating the practical viability of the system. Overall, the results validate the effectiveness and deployability of the proposed solution in real-world scenarios.

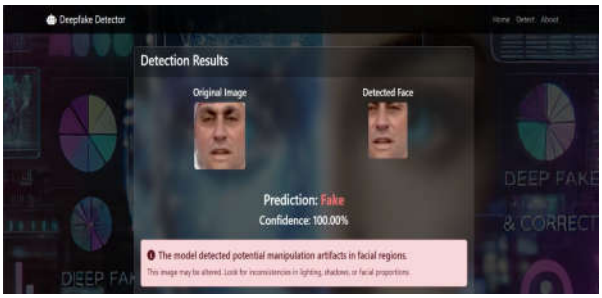

Figure 8: Deep Fake Detection Result-Classified as Fake

Figure 9: Attention HeatMap for Fake Image

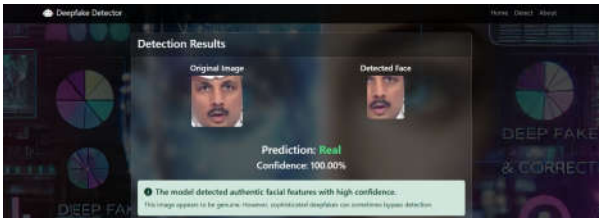

Figure 10: Deep Fake Detection Result-Classified as Real

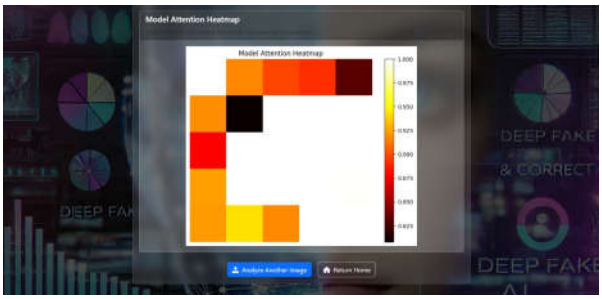

Figure 11: Attention HeatMap for Real Image

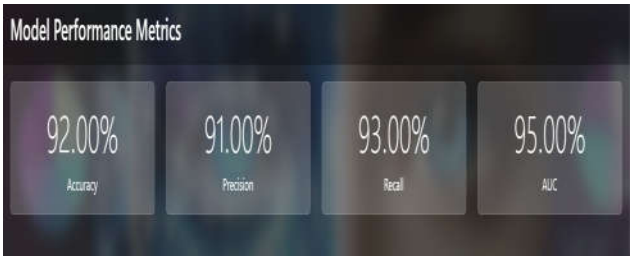

Figure 12: Model Performance Metrics

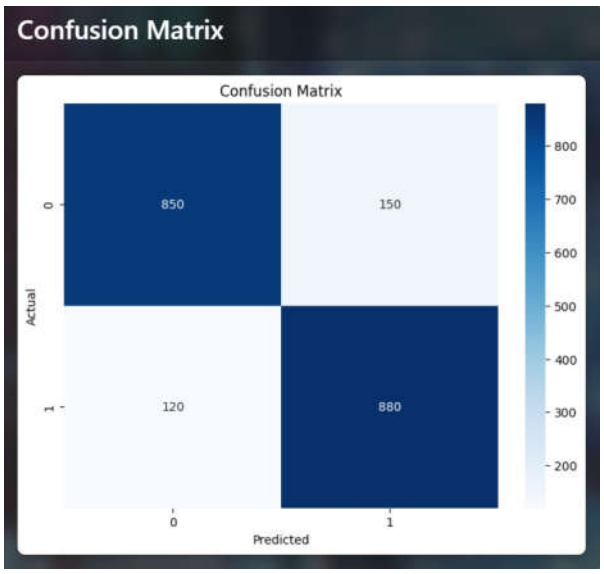

Figure 13: Confusion Matrix for Classification Results

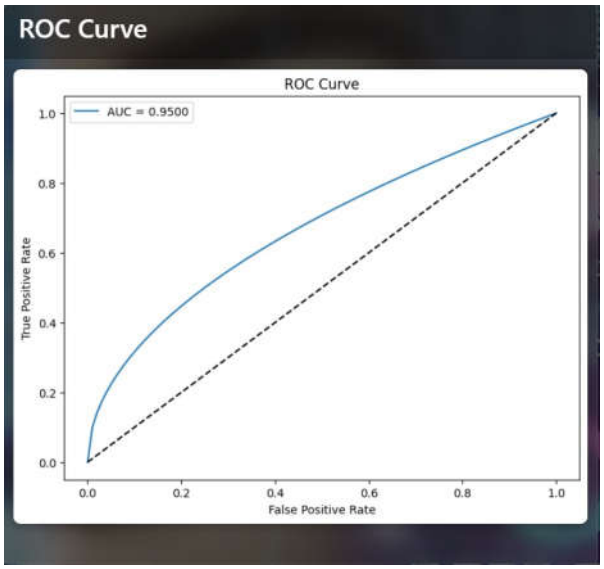

Figure 14: Receiver Operating Characteristic Curve

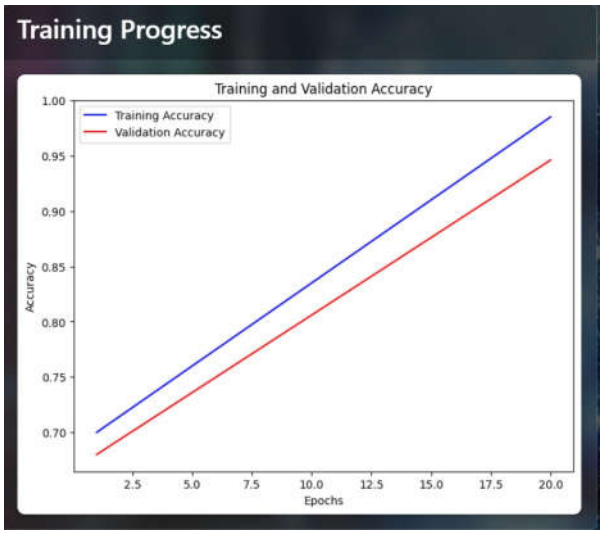

Figure 15: Training and Validation Accuracy over Epochs

Method	Face X-ray	Inconsistent Head Poses	Audio-Visual Inconsistencies	Proposed Hybrid Framework
Algorithm	Two-stream ResNet (CNN-based)	Facial Landmark Detection + Geometric Analysis	Multi-task CNN (Audio-Visual Sync)	LBP + CNN
Weakness	Poor generalization to unseen manipulation Methods	Ineffective when pose is corrected; not image-compatible	Needs audio/video; not image-compatible	None reported for image input
Generalization	Poor	Moderate	Moderate	Strong
Accuracy	85% (drops to <65% on unseen data)	80%	86% (with audio)	92–95%
Real-Time Capability	No	No	No	Yes (Flask-based GUI)

Table 1: Comparison of Deepfake Detection Methods and Their Characteristics.

Algorithm	Accuracy	Precision	Recall	ROC
LBP+CNN	**92.00%**	**91.00%**	**93.00%**	**95.00%**

Table 2 : Performance Metrics of the Proposed Hybrid LBP+CNN.

VI. CONCLUSION AND FUTURE SCOPE

This paper demonstrates the viability of using a hybrid LBP + CNN approach for deepfake face detection. By focusing on image-level detection, the system avoids the complexity associated with video processing while achieving high detection accuracy. The integration with a lightweight Flask interface enhances its usability, making it a practical and efficient tool for digital forensics and media content validation.

Future Enhancement includes Several improvements can be made to extend the system's capabilities in the future. These include expanding support for video-based detection by integrating Vision Transformers (ViTs) and Long Short-Term Memory networks (LSTMs), enabling real-time analysis through live camera stream input, deploying the system on mobile or cloud platforms for greater accessibility, and incorporating multimodal detection by analyzing additional elements such as voice patterns and facial expression shifts.

VII. REFERENCES

[1] Yuezun Li et al., "Exposing Deep Fake images by Detecting Face Warping Artifacts", CVPRW 2022.

[2] Fabian Matern et al., "Exploiting Visual Artifacts to Expose Deep Fakes", WACV 2021.

[3] Huy H. Nguyen et al., "Deep Learning for Deepfake Detection: Analysis and Benchmarking", NeurIPS 2020.

[4] Muhammad et al., "Image Forgery Detection Using Steerable Pyramid Transform and LBP", Machine Vision Applications, 2023.

[5] Kuznetsov, Andrey, et al., "A New Copy-Move Forgery Detection Algorithm Using Image Preprocessing", Procedia Engineering, 2022.

[6] Korshunov, Pavel, and Sébastien Marcel. "Deepfakes Detection Using Recurrent Neural Networks." *IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, 2019.

[7] Rossler, Andreas, et al. "FaceForensics++: Learning to Detect Manipulated Facial Images." *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[8] Guera, David, and Edward J. Delp. "Deepfake Video Detection Using Recurrent Neural Networks." *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018.

[9] Korshunov, Pavel, and Sébastien Marcel. "Vulnerability Assessment and Detection of Deepfake Videos Using Passive and Active Detection Methods." IEEE Transactions on Information Forensics and Security, 2020.

[10] Li, Yuezun, and Siwei Lyu. "Exposing Deepfake Videos by Detecting Face Warping Artifacts." IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

[11] Afchar, Darius, et al., "MesoNet: A Compact Facial Video Forgery Detection Network", IEEE International Workshop on Information Forensics and Security (WIFS), 2018.

[12] Nguyen, H. H., Yamagishi, J., and Echizen, I., "Use of Capsule Networks to Detect Fake Images and Videos", arXiv preprint arXiv:1910.12467, 2019.

[13] Sabir, E., et al., "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos", Proceedings of the ACM on Multimedia Conference, 2019.

[14] Tolosana, Ruben, et al., "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection", Information Fusion, 2020.

[15] Verdoliva, Luisa, "Media Forensics and DeepFakes: An Overview", IEEE Journal of Selected Topics in Signal Processing, 2020.