International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-31

https://doi.org/10.70917/ijcisim-2026-1972
Article

ANALYZING VISUAL FEATURES FOR REAL VS. FAKE FACE DETECTION: PREPARING FEATURE-AWARE HYBRID CNN MODELS

Nitu Yadav[1], Savita Sheoran[2]

[1] Department of Computer Science and Engineering, Indira Gandhi University, Meerpur, Rewari, Haryana, India.
niturao.2810@gmail.com
[2] Department of Computer Science and Engineering, Indira Gandhi University, Meerpur, Rewari, Haryana, India.
savita.sheroran@igu.ac.in

Abstract: Recent advances in generative adversarial networks (GANs) have led to the availability of high-quality synthetic facial images (deepfakes) at large scale. These include but are not limited to, such advances pose challenges in digital security, misinformation detection and biometric identification. The above methods are used in this work to combat the growing threat of the developing synthetic media by understanding the discriminative features between actual and fake face images. The primary goal is to discover and learn the most discriminative handcrafted feature, which we can use to build more efficient hybrid CNN-based classifiers in subsequent studies. We show a variety of methods of feature extraction and visualization by leveraging public dataset "140k Real and Fake Faces". Color histograms, Local Binary Patterns (LBP), Sobel, Canny edge detectors are some of the traditional image descriptors for the representation of color, texture, edge and frequency data. We have applied Discrete Cosine Transform (DCT). These manually created features have been converted into 2-D spatial models by PCA, tSNE and UMAP to describe their ability to separate classes. This work further pinpoints significant cues for distinguishing real faces from computer-generated or printed ones, and lays the groundwork for a hybrid CNN model that integrates learned and hand-crafted features.
Keywords: Fake Face Detection; Handcrafted Features; Convolutional Neural Networks; Feature Visualization; Deep Learning.

1. Introduction

It has become a great challenge to have visual contents remain authentic in the digital age. Fast advancements in machine learning and generative modeling approaches, and specifically in Generative Adversarial Networks (GANs), have made it possible for people to generate hyper realistic synthetic images, which could not be less resemble a real photograph. One of the better known and most controversial outputs of GANs is the generation of synthetic human faces, aka "deepfakes." With the growing prevalence of deepfakes in online media and communication platforms, trust, digital security and the spread of misinformation become a new and serious concern. It has become an increasingly common technological necessity to distinguish real from fake faces, with synthetic media increasingly becoming refined and as a result, ensuring the integrity of visual information is increasingly difficult due to such. Though deepfake detection has generated significant academic interest, accurate identification of synthetic and genuine faces has been a challenging task. One of the main reasons is the high visual fidelity of modern GAN-generated images. Where earlier deepfakes were often visibly flawed or artifacts, modern models like StyleGAN2 and StyleGAN3 generate images that retain natural lighting, realistic skin textures, facial symmetry and finer features like blemishes, glasses, hair strands, etc. Detecting such sophisticated synthetic content, however, is becoming an increasingly complex task. There is also a generalization problem with different datasets and deepfake generation algorithms, which inhibits the development of efficient detection systems. This study aims to provide clear insights on the differences and relationships between handcrafted features and deep learning-based ones in image classification. A handcrafted feature is a descriptor that is manually designed and needs domain

knowledge defined by an expert to capture characteristic visual information. Most of these traits target low- and mid-level image characteristics:

• Edges (coming from Sobel and Canny filters): to observe shifts in intensity and object boundaries.

• Color distributions are given through Color Histograms, on the basis of which we can see how RGB intensities spread.

• Texture patterns are analyzed by using Local Binary Patterns in order to measure spatial structure of grayscale images.

• Frequency characteristics (using Discrete Cosine Transform): to study periodic components and locate compression inconsistencies.

The significant merit of handcrafted features is interpretability, meaning a feature has a specific, understandable purpose. But these features are often not robust in representing generalizability on many datasets or capturing high-level semantic levels that the complexity of current synthetic images demand. In contrast, deep learning models (especially Convolutional Neural Networks [CNN]) learn hierarchical representation directly from raw pixels. When designing new architectures, some of the early layers of a CNN will inherit edges or corners, while the deeper layers learn a lot more properties (like facial structure or identity). Because of such an end-to-end model, it allows for a minimal amount of manual feature engineering and is well adapted for different data samples. As powerful as CNNs are, they are still subject to criticism for their “black box” behavior. Unlike handcrafted features, CNNs are trained on internal representations which are non-interpretable by themselves. This lack of transparency is an important roadblock to adopting such an approach for IoT, which requires transparent accountability. Many of the best features of handcrafted models also provide clarity and computational efficiency that can be combined with the representational power and adaptability of deep learning. Hybrid models have shown good robustness in areas like medical imaging, remote sensing and biometric recognition as well as generalizability. On the other hand, in the context of big data, in machine learning-based learning, it is still a matter of time before models are capable of generalizing from single computational samples to multiple datasets and to fit each dataset completely. For instance, mixing Histogram of Oriented Gradients (HOG) and CNN-based embeddings yields excellent performance in facial recognition. The iris detection system has also been promising to combine handcrafted texture descriptors and deep learning features to be able to capture changes in lighting, occlusion and pose. Nevertheless, for fake face recognition hybrid methods are still underexplored, particularly for systematic feature-level investigation. Lack of research on empirical data that:

• Discover which handcrafted features are the most discriminative.

• Compare these features with CNN-learned embeddings.

• Visualize the feature space, and interpret the feature space helps shape your model.

In addition, there is limited research that has used dimensionality reduction methods (such as PCA, t-SNE, and UMAP) to capture the level of similarity of feature space for distinguishing between real and fake classes. Without this understanding, the formulation of a hybrid model is often based on heuristics or guesswork of fusion rather than relying on data-driven choices. These limitations have led to the need for systematic extraction as well as evaluation for handcrafted and CNN-generated features on the basis of their potential to segregate the real and fake face by dimensionality reduction and visualization. This work aims at extracting and examining the visual characteristics – either manually or learned – that differentiate real from fake faces. By doing so, we hope to contribute to interpretable and efficient hybrid CNN architectures for future works.

In contrast to the current deepfake detection research, which is largely concerned either with model creation or with performance finalization, this work provides a comparative framework based on features in a systematic manner, combining handwritten descriptors and CNN-based embeddings. The novelty of the study consists in (i) holistic assessment of color, texture, edge, and frequency attributes, (ii) holistic visualization of feature space separability analysis using PCA, t-SNE, and UMAP, and (iii) deriving data-driven information on hybrid CNN model design instead of heuristic feature fusion. This analytical method is structured by bringing together interpretability and performance to offer a baseline framework of next-generation explainable deepfake detection systems.

The main objectives of this study are as follows:

- To methodically sample a varied class of handcrafted features from facial images across four dominant visual aspects:
- Color Histograms – color distribution across RGB channels.

o Edge Features: using Sobel and Canny detectors to extract spatial boundaries

o Texture Descriptors: denote details of the micro-texture features by Using Local Binary Patterns (LBP)

o Frequency Features: capture spatial frequency patterns applying Discrete Cosine Transform (DCT)

To construct a lightweight Convolutional Neural Network (CNN) for binary classification (real vs. fake). We derive dense-layer embeddings of high-level, learned features well-suited for classification from this model. To assess and visualize the discriminative performance of handcrafted and CNN-generated features using three cutting-edge dimensionality reduction approaches:

o PCA: For using linear feature projection.

o t-Distributed Stochastic Neighbor Embedding (t-SNE): To recover local structure in non-linear projections.

o Uniform Manifold Approximation and Projection (UMAP): For high-dimensional manifold preservation.

• For the purpose of comparison of handcrafted features against CNN-learned embeddings in terms of:

o Class separability in 2D plots.

o Interpretability and potential for hybrid integration

o Workability for live or limited resource use cases.

Indeed, this research resolves one of the most pressing problems in the contemporary, AI-intensive digital space, i.e., finding synthetic facial images through rigorous quality testing. Through a structured approach to extract, analyze, and compare both handcrafted and CNN derived visual features.

In the rest of the paper, the order of the work is set by:

Section 2, we provided a comprehensive survey addressing related research works in the area of Fake Image Detection considering hand-induced as well as deep learning-based methods to focus on the major contributions, limitations, as well as key areas of study that can be advanced in next.

Section 3 presents a rich methodology-preparation of data, handcrafted and custom-designed CNN based feature extraction techniques and dimensionality reduction techniques for visual representation of feature separability.

Section 4 presents and analyses experimental results presented based on a set of feature detection and visualization experiments.

Section 5, the paper ends by summarizing its main findings, exploring the implications of interpretable feature analysis in fake-image detection, future avenues, research, including hybrid fusion technique with hybrid model and real-time.

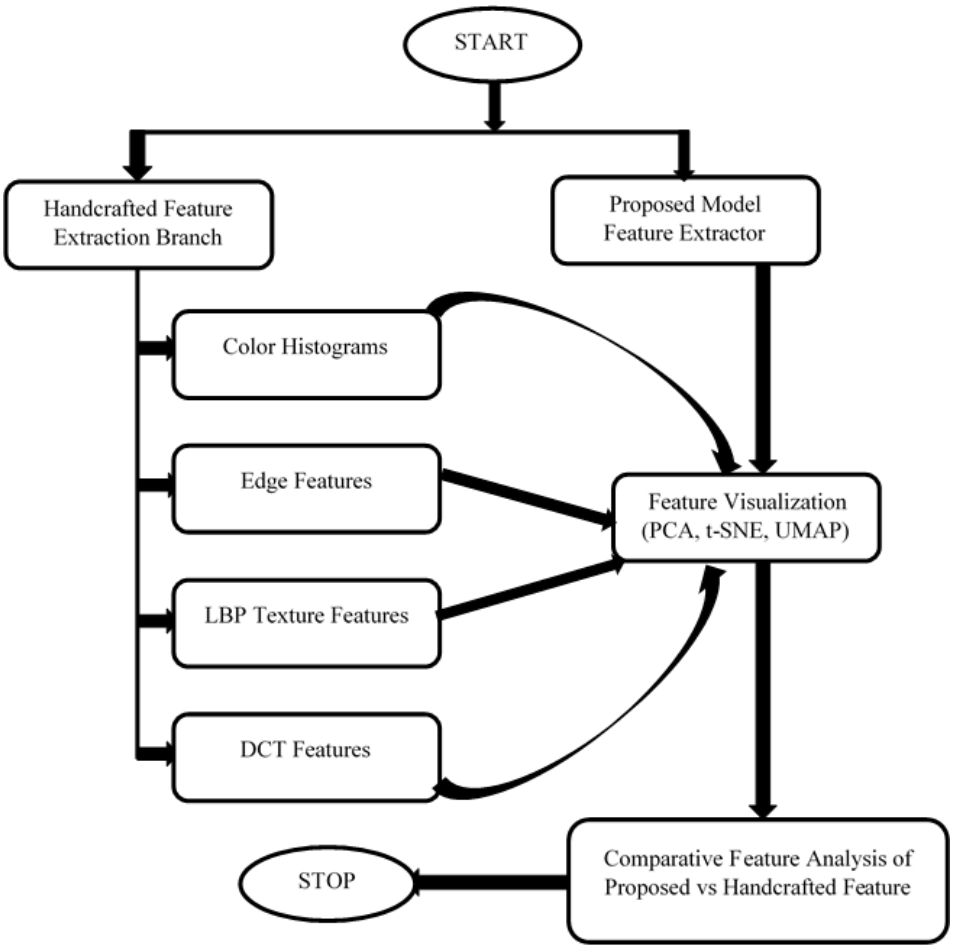

Figure 1 Proposed Work flow of feature analysis

2. Related Work

The rise of deepfake technologies has led to a surge in research to better detect and mitigate the risk of synthetic facial media. It summarizes main contributions from 2019 to the present, addressing novel methodologies, benchmark results and the present state-of-the-art of fakes detection technologies.

2.1 Approaches based on Deep Learning:

Deep learning has been at the center of deepfake detection, making the most out of neural networks to detect subtle artifacts in synthetic media. Passos et al. (2022) [1] presented a systematic literature review for the detection on deep learning-based methods and pointed out the performance of CNN models, transformer ones, etc for deepfake detection. They also focused on how datasets that are of good diversity in content and size benefit the model to make the model generalize to different types of fakes. Zhao et al. (2021) [2] proposed a multi-attentional deepfake detection network that deals with fine-grained classification. Their multi-spatial attention head and textural feature enlargement block technology extracts the more implicit cues of the image and exploits subtle discrepancies between the true and the fake image and state-of-the-art performance on several benchmarks in different scales. Xu et al. (2023) [3] provided a new Deepfake detection approach Multi-Channel Xception Attention Pairwise Interaction (MCX-API). By using pairwise learning and multi-channel color space features, this method improves detection accuracy and generalization for a broader variety of Deepfake generation methods. By performing well on the different open-set and closed-set cases of multiple public datasets, achieving impressive performance such as 98.48% BOSC accuracy upon FF++ inference and 90.87% accuracy on CelebDF in the end, this model is quite robust and can be easily adopted with real-world application. Lin et al. (2024) [4] introduced Curricular Dynamic Forgery Augmentation (CDFA) as a new Deepfake detection method. This approach improves generalization by jointly training a detector based on a dynamic forgery augmentation policy, which integrates with augmentations in a

progressive curriculum. It adopts the new augmentation feature – self-shifted blending that mimics the temporal inconsistency of Deepfakes. CDFA is remarkably superior across datasets and manipulation strategies with better performance than general methods in most of the benchmark. Chen et al. (2022) [5] developed a generalizable Deepfake detection based on improving forgery variety and the model sensitivity. Their method synthesizes most of the augmented forgeries in different configurations and trains a model to predict such configurations, enhancing the model's capability to detect varied manipulation. They also employ adversarial training for generating the most difficult forgeries directly. Based on their experimental results, in terms of generalization their method significantly outperforms the current state-of-the-art methods. Lai et al. (2024) [6] proposed Generalized Multi-Scenario Deepfake Detection (GM-DF) model for enhancement of generalization on various datasets and attack types. They found that direct training on combined datasets leads to performance degradation due to domain difference. This led to the implementation of a hybrid model that uses a domain expert for the domain-related features, leveraging CLIP for common feature alignment, masked image reconstruction to capture forgery detail and applying a domain-aware meta-learning protocol with new alignment loss. And their framework greatly increases generalization across many different real-world environments**.**

Table 1 Deep Learning Methods Comparison

Citation	Methodology	Research Gap Addressed	Limitations
Passos et al. (2022) [1]	Surveyed deep learning-based methods (CNNs, Transformers) for deepfake detection	Lack of a unified overview of effective architectures and dataset diversity needs	Review-based; does not propose or evaluate a novel detection method
Zhao et al. (2021) [2]	Developed a multi-attentional network using spatial attention and textural features	Insufficient fine-grained detection in existing models	May overfit to specific dataset artifacts; limited open-set evaluation
Xu et al. (2023) [3]	Proposed MCX-API using pairwise learning and multi-channel color representations	Weak generalization to unseen DeepFake types in open-set scenarios	Evaluated on limited datasets; real-world adaptability not fully explored
Lin et al. (2024) [4]	Introduced CDFA with curricular forgery augmentation and self-shifted blending	Poor cross-dataset and cross-manipulation generalization	Complexity may hinder real-time or low-resource deployment
Chen et al. (2022) [5]	Used forgery configuration prediction and adversarial training for diverse forgery simulation	Limited diversity and sensitivity in training data	High training cost due to dynamic adversarial synthesis
Lai et al. (2024) [6]	Proposed GM-DF with domain-aware meta-learning, CLIP alignment, and hybrid expert modeling	Performance drop in multi-dataset training due to domain shifts	Complex training pipeline requiring significant computational resources

2.2 Hybrid CNN-RNN Models:

Hybrid models that combine CNNs with RNN (Recurrent Neural Network), such as Long Short-Term Memory (LSTM) networks, show great promise in capturing spatial aspects as well as temporal features in video content. Saikia et al. (2022) [7] introduced a CNN-LSTM model that exploits optical flow features for detecting deepfakes in videos. They were able to achieve high accuracy on several datasets such as DFDC, FF++, Celeb-DF. Zhang et al. (2021) [8] presented a deepfake detection method, Temporal Dropout 3-Dimensional Convolutional Neural Network (TD-3DCNN). In this method, temporal variability between video frames is captured using a 3D

CNN plus a temporal dropout method, which samples video frames at random during training. Because of this approach, the model generalization is stronger and overfitting behavior is much reduced thus improving the video-level deepfake detection accuracy. This method proved to be effective and robust by experiment using benchmark datasets. With this proposed method, Al-Dhabi and Zhang (2021) [9] reported a deepfake detection system that provides a convergence of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to enhance the detection accuracy. They had a ResNeXt50 pre-trained model to extract features and an LSTM in addition to that, to capture the intra-frame and inter-frame temporal features of the videos. Using a CNN-RNN hybrid method leads to effective deepfake detection. This results in competitive performance on a wide variety of video data while still keeping its basic architecture simple. Elhassan et al. (2022) [10] proposed a deepfake detection method concentrated on teeth and mouth movements--features which are difficult to faithfully imitate for fake videos. This method leverages multi-transfer learning to pick up biological signals from facial regions using a diverse list of pre-trained models (for example, DenseNet121, EfficientNetB7, Xception). This method enhances the recognition of deepfakes with higher accuracy than its predecessor model. Khalil et al. (2021) [11] proposed a new generation deepfake detection algorithm known as iCaps-Dfake which addressed these problems. The proposed method employs LBP for texture analysis and HRNet with modifications, as well as CapsNets with a routing mechanism to improve classification performance. Ismail et al. (2021) [12] presented a YOLO-CNN-XGBoost-based deepfake detection process. The method integrates YOLO face detector for face region extraction, InceptionResNetV2 for deep feature extraction, and Extreme Gradient Boosting (XGBoost) classification using deepfake classification methods. The model achieved high-performance (AUC 90.62% and accuracy 90.73%) when tested on the merged CelebDF-Face Forensics++ (c23) dataset and outperformed a number of state-of-the-art techniques in the detection of face-swapped deepfakes.

Table 2 Hybrid Models Comparison

Citation	Methodology	Research Gap Addressed	Limitations
Saikia et al. (2022) [7]	CNN-LSTM using optical flow to capture spatial and temporal features in videos	Improved detection by leveraging motion dynamics and deep learning integration	May not generalize well to highly compressed or low-resolution videos
Zhang et al. (2021) [8]	Temporal Dropout with 3D CNN (TD-3DCNN) for video-level detection	Addressed temporal inconsistencies and overfitting in video-based detection	Limited scalability to long video sequences or real-time detection
Al-Dhabi & Zhang (2021) [9]	ResNeXt50 CNN for feature extraction + LSTM for temporal modeling	Combines intra-frame and inter-frame features for better temporal analysis	Simple architecture might miss complex manipulations or context cues
Elhassan et al. (2022) [10]	Multi-transfer learning focusing on teeth and mouth movements as biological signals	Identified rarely explored facial regions hard to forge accurately	May struggle with videos where the mouth is obscured or poorly visible
Khalil et al. (2021) [11]	iCaps-Dfake: LBP + modified HRNet + Capsule Networks with concurrent routing	Tackled poor generalization of previous models across datasets	Computationally complex; may be resource-intensive for deployment
Ismail et al. (2021) [12]	YOLO face detection + InceptionResNetV2 CNN + XGBoost classifier	Introduced a hybrid ensemble for improved classification accuracy	Potentially limited by the fixed feature extractor and merging strategy

2.3 Detection Techniques that are Characteristic and Light Weighted:

For computational efficiency, designers also have used lightweight models that use handcrafted features. Yasir and Kim (2025) [13] proposed a method that fuses texture-based features (Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP) and KAZE) into a machine learning model. The approach was very accurate on the Face Forensics++ and Celeb-DF datasets while keeping their computational cost low - and hence was applicable for real-time use. Du et al. (2019) [14] proposed a Locality-Aware Autoencoder (LAE) to learn intrinsic representations from forgery regions. The generalization on unseen manipulations was enhanced by incorporating pixel-wise masks at training and their model achieved better accuracy compared to multiple state-of-the-art methods. Shao et al. (2025) [15] offered Deepfake-Adapter, a recent, parameter efficient paradigm for boosting deepfake detection using the generalization of large pre-trained ViTs. The approach avoids using only low-level forgery models and exploits high-level semantic information via lightweight dual-level adapters: Globally-aware Bottleneck Adapters and Locally-aware Spatial Adapters. These modules are embedded in the frozen ViT backbone to optimize global and local forgery cues without overfitting. From experiments on different benchmarks, we can see that the model does better in cross-dataset as well as cross-manipulation cases. Yan et al. (2023) [16] introduced a new disentanglement-based mechanism to improve the generalization of detections for deepfake based on preventing overfitting in the presence of forgery-irrelevant and method-specific features. This decomposes image data into 3 primary parts called: forgery-irrelevant features, method-specific features and common forgery features. Using multi-task learning and exploiting binary and multi-class classification methods, as well as introducing contrastive regularization, their method isolates robust common forgery cues for detection. Research has suggested that basing only such features on these findings greatly increases generalizability to novel types of forgery than any existing model. Guan et al. (2022) [17] proposed the Local- & Temporal-aware Transformer-based Deepfake Detection (LTTD) framework to improve post-processing resistance in deepfake videos. They combine local low-level features with temporal dynamics by applying Local Sequence Transformer (LST), that learns temporal constancy over a limited region. This local detail is subsequently clustered at a global level with contrastive learning for the last classification. The framework surpasses on most previous methods for detecting the weak forgeries and also to great generalization in the multi-dataset. Shuai et al. (2023) [18] has introduced a secure deepfake detection framework, which enhances generalization ability and alleviates some of the problems of overfitting by expanding the range of forgery cues the model learns. This approach significantly enlarges potential forgery regions with a two-stream network model but utilizes multi-stream, multi-scale feature analysis across three collaborative learning modules designed to implement multi-stream, multi-scale feature handling. To address their lack of grounding with a ground truth forgery annotation, they proposed a Semi-supervised Patch Similarity Learning method to derive forged locations. Their approach outperformed their competitors on the six benchmark datasets and brought the performance standard for frame-level detection and video-level detection performances to new levels**.**

Table 3 Feature-Based and Lightweight Detection Methods Comparison

Citation	Methodology	Research Gap Addressed	Limitations
Yasir and Kim (2025) [13]	Combined handcrafted texture features (HOG, LBP, KAZE) with machine learning for efficient detection	Computational efficiency and real-time applicability in deepfake detection	May be less effective against highly sophisticated or evolving deepfake generation methods
Du et al. (2019) [14]	Locality-Aware AutoEncoder with pixel-wise masks to focus on forgery regions	Generalization to unseen manipulations by learning intrinsic forgery representations	Dependence on accurate pixel-wise masks; may struggle with complex forgery artifacts
Shao et al. (2025) [15]	Parameter-efficient tuning of pre-trained Vision Transformers using dual-level adapters (global/local)	Lack of high-level semantic understanding and overfitting to low-level forgery cues	Requires large pre-trained ViTs; may have higher computational demand than lightweight models
Yan et al.	Disentanglement framework	Overfitting to forgery-	Complexity of multi-

(2023) [16]	decomposing image features into forgery-irrelevant, method-specific, common	irrelevant and method-specific features; enhancing generalizability	task learning and contrastive regularization; potential training instability
Guan et al. (2022) [17]	Local Sequence Transformer capturing temporal consistency in spatially restricted regions with contrastive learning	Robustness against post-processing and temporal forgery detection	Potential sensitivity to video quality degradation; model complexity might limit real-time use
Shuai et al. (2023) [18]	Two-stream network with multi-scale collaborative modules and semi-supervised patch similarity learning	Overfitting to dominant forgery regions and lack of annotated forgery localization	Requires semi-supervised annotation strategy; may increase training complexity and runtime

2.4 Transformer-Based and Multi-Modal Techniques:

As they can represent long-range dependencies, transformers have been extended for large-scale and local deepfake detection. Wang et al. [19] presented the Multi-modal Multi-scale Transformer (M2TR), a model to process image patches with different sizes to detect local inconsistencies. By combining frequency domain analysis and RGB, M2TR was superior to current approaches. Waseem et al. (2023) [20] proposed a new attention-based multi-task deepfake detection approach that improves classification and forgery localization. Their proposed approach combines the attention-based decoder with an encoder to create local feature maps that identify manipulated regions. Combined with frequency domain information, these attention-refined features create a powerful and discriminative model to identify face and expression swaps. The system addresses the performance decrements that can be common to cross-dataset explorations by enhancing generalizability across datasets. The experimental results also validated the competitive nature of their approach with existing state-of-the-art approaches in in-domain and cross-domain settings. Guan et al. (2022) [21] presented the LTTD (Local- & Temporal-aware Transformer-based Deepfake Detection), which focuses on local spatial and temporal features to improve detection robustness to post-processing distortions. The core of such approach is the Local Sequence Transformer (LST) that captures the temporal consistency over small spatial scales using shallow 3D filters to record temporal consistency in small areas. In particular, their approach achieved state-of-the-art performance on multiple benchmark datasets and proved to be state-of-the-art in terms of their approach, leading to better detection accuracy and generalization performance. Wang et al. (2022) [22] had proposed the Localization Invariance Siamese Network (LiSiam) in order to improve the deepfake detection robustness with varying image qualities and cross-database conditions. They achieve this through a Siamese analysis which compares original and corrupted images, enforcing consistency in the localization of forged regions through an entirely novel form of localization invariance loss. Further Mask-guided Transformer is added to model the contextual relationship of the manipulated areas and surrounding environment. We propose the multi-task learning framework with segmentation, classification and localization as tasks to make the target performance in Face Forensics++ and Celeb-DF datasets superior. Pu et al. (2022) [23] proposed a dual-level collaborative approach for deepfake detection to overcome performance degradation when training with imbalances in real-world datasets. Their approach also detects forgeries simultaneously at the frame and video level via a joint loss function that minimizes AUC and error rates. By combining a novel AUC-based loss and multitask learning, the model overcomes the limitation of focal loss when dealing with data imbalance. The cooperative design allows for mutual reinforcement between frame-level and video-level detection mechanism, increasing robustness to the varying video quality and the generalization of cross-dataset evaluations. Ganguly et al. [24] introduced the ViXNet model, which is a combined deepfake detector that can generalize to new datasets, by capturing subtle manipulation artifacts on the whole face.

Table 4 Transformer-Based and Multi-Modal Method Comparison

Citation	Methodology	Research Gap Addressed	Limitations
Wang et al. (2021)	Introduced Multi-modal Multi-scale Transformer	Lack of multi-scale and modality fusion in	May require high computational resources

[19]	(M2TR) using RGB and frequency domain features across patch scales.	transformer-based deepfake detection.	due to multi-scale input processing.
Waseem et al. (2023) [20]	Proposed attention-based multi-task network combining localized attention features and frequency domain signals.	Weak generalization in cross-dataset and expression swap detection.	Localization quality relies on attention accuracy; may struggle with subtle manipulations.
Guan et al. (2022) [21]	Developed LTTD framework with Local Sequence Transformer and contrastive learning for spatial-temporal cue modeling.	Poor robustness to post-processing and limited use of local temporal features.	Dependence on temporal consistency may reduce effectiveness on static image datasets.
Wang et al. (2022) [22]	Proposed LiSiam using Siamese network with localization invariance loss and Mask-guided Transformer.	Degradation in detection performance under varying image quality and cross-dataset shifts.	Requires image pairs (original and degraded), increasing input complexity.
Pu et al. (2022) [23]	Designed dual-level collaborative framework optimizing AUC with joint frame- and video-level detection.	Limited performance under imbalanced and real-world data distributions.	May not capture fine-grained forgery details due to video-level generalization.
Ganguly et al. (2022) [24]	Introduced ViXNet with Vision Transformer and Xception in dual-branch setup for learning local/global inconsistencies.	Poor generalization across datasets due to reliance on dataset-specific features.	Transformer-based architecture may be resource-intensive for real-time use.

2.5 Physiological and Biological Signal-Based Detection:

Chakraborty and Naskar (2024) [25] provided a survey on the integration of human physiological information and facial biomechanics used to achieve reliable deepfake detection. Their work is urgent because it solves a problem and fills the current hole for detectors that are able to bypass such demographic, social and cultural biases, which are often difficult for the traditional techniques to deal with. Through the approach of recent work leveraging physiological signals (micro-expressions and involuntary facial dynamics), they show that such methods provide a more accurate and less biased way to detect synthetic media. Akhtar et al. (2024) [26] conducted an extensive review of existing image, video, and audio deepfake datasets and detection methods to further progress the detection of next generation detectors. Their work points to the dual-use of deepfake technologies — positive and negative — and notes how tools that make deepfake creation easy to automate the process for average users are becoming an increasingly popular and widely available tool, if not a necessity. The paper introduces current detection techniques as the top-busting problem with regard to generalization, robustness and explainability and provides a solution to some pressing issues. Le et al. (2023) [27] dealt with the growing problems of finding very realistic deepfake media, discussing the drawback of existing detection techniques Shree et al. (2024) [28] has demonstrated a comprehensive work on deepfake detection for generative AI and large language models in general. Their paper highlights the technical comparison between traditional AI and modern generative models, demonstrating a robust execution of a deepfake image detection model built on RESNET-50 and MTCNN. Through hypothesis testing, they test the utility of the approach to a wide range of real-world deepfake scenarios in order to supplement current AI-based content validation by utilizing next-generation neural network architectures. Bendiab et al. (2025) [29] performed a thorough study of the growing problems that deepfakes present to digital media forensics. Their work highlights the growing realism and access of the deepfake technology that prevents both human and algorithmic detection. In this paper we discuss existing detection techniques, limitation of the methods, and the best practices that will aid in creating better AI applications, expanding datasets and establishing appropriate regulations so as to cope with

serious deepfake threats. Ren et al. (2025) [30] studied the gap that is growing between academic deepfake detection research and its practical real-world application. The analysis by Tekam et al. (2025) [31] describes a hybrid CNN-Transformer system of deepfake image detection because it is an efficient system that incorporates both the spatial and global context analysis in feature detection. The method is shown to be more accurate in detecting, but the lack of discussion on diversity of data sets, their ability to generalize, and transparency of the experiment slightly limit their further application.

Table 5 Physiological and Biological Signal-Based Detection Approach Comparison

Citation	Methodology	Research Gap Addressed	Limitations
Chakraborty and Naskar (2024) [25]	Survey of deepfake detection using human physiological signals and facial biomechanics	Lack of bias-resilient detection across demographic, social, and cultural lines	Relies heavily on emerging techniques; lacks implementation of unified detection framework
Akhtar et al. (2024) [26]	Review of image, video, and audio deepfake datasets and detection tools; identification of research challenges	Weak generalization, explainability and accessibility of detection models	Primarily theoretical; does not propose a specific detection method
Le et al. (2023) [27]	Critical analysis of detection pipeline limitations and threat from novel deepfake generators	Preprocessing artifacts and unseen generator types not addressed by existing models	Lacks a proposed new model; emphasizes problem more than solution
Shree et al. (2024) [28]	Implementation of deepfake image detection using RESNET-50 and MTCNN, with hypothesis testing	Difficulty in detecting highly realistic deepfakes using traditional techniques	Evaluates only one detection framework; generalizability not tested across datasets
Bendiab et al. (2025) [29]	Analytical review of deepfake challenges and current detection limitations; proposal of legal and technical frameworks	Lack of robust forensics techniques and legal infrastructure to support detection	Suggests improvements without experimental validation or new dataset
Ren et al. (2025) [30]	Introduction of real-world faceswap dataset; evaluation of SOTA detectors under post-processing effects	Poor real-world performance of detectors due to overlooked post-processing like super-resolution	Sheds light on the problem but does not propose a new detection model

3. Methodology

Nevertheless, most of the current solutions are not interpretable and strong enough to be utilized in real-world scenarios and at high stakes in society, such as misinformation detection and cybersecurity surveillance. This section outlines the systematic approach employed for extracting and evaluating discriminative features from real and fake facial images. The methodology integrates handcrafted image processing techniques and a custom Convolutional Neural Network (CNN) to analyze and compare features, followed by visualization using dimensionality reduction techniques. The suggested framework will be developed keeping in mind software engineering principles and

considerations such as the modular pipeline design, the reusability of the feature extraction modules, and its ability to integrate with scalable machine learning deployment platforms like cloud and edge computing systems.

3.1 Dataset Description

The research utilizes the Kaggle publicly available dataset "140k Real and Fake Faces". The dataset is designed for identifying real and fake faces and the total number of images is about 140,000, where each image is divided into real (face image) and AI-generated (fake) face images. The dataset can be divided into three main directories:

Training data: This is used for training the model and getting to know the hidden key attributes of realistic/fake face.

Validation set: Used to control hyperparameters and check for model performance while training, avoiding overfitting.

Test Set: Used only for testing the model's performance after training.

There are two subfolders per each of these directories:

Fake — a type of synthetic face images created (like StyleGAN) with GAN.

This hierarchical structure can be implemented in a TensorFlow/Keras file directly with image data generators for both training and evaluation. Every image is either real (label=1) or fake (label=0), depending on where it belongs.

The following preprocessing steps have been realized to standardize input dimensions and promote effective learning:

Resizing: All such images are resized to 150 × 150 pixels.

Normalization: Pixel values were normalized to [0, 1] using rescale=1./255 to normalize for training the neural network.

Shuffling: Shuffle images in the training data to avoid order bias was done by the generators of the datasets.

Augmentation: The dataset is large and balanced, however and have omitted more augmentation such as other features for handcrafted feature extraction as no additional augmentation was performed, since no extra rotation, flipping or zooming was applied to keep the focus on the original image content while handcrafted feature extraction was being performed.

3.2 Handcrafted Feature Extraction

This paper examines the deployment of a number of manually hand-crafted feature descriptors to detect visual patterns that differentiate real from fake facial images. These features can all be chosen for their interpretability and they can be employed for various aspects of visual information.

3.2.1 Color Histogram Features

Color histograms are a mainstay in image creation that describe the pixel intensity distribution of the various color channels. Histograms were computed in RGB color space in this study due to keeping the natural appearance and in accordance with the original camera capture and storage. Images were resized to 150 × 150 pixels and separated into their three pixels (R, G, B). With respect to each channel, the total areas in each bin covering its full intensity for pixels were identical at 32 bin level under histograms in each channel [0,255]. Bin counts for each channel c∈{R, G, B} are stored with H c(i) and i=1,2,...,32 is the bin index. To ensure uniformity and comparability, each channel histogram was L2-normalized as follows:

$$\widehat{H}_c(i) = \frac{H_c(i)}{\sqrt{\sum_{j=1}^{32} \quad H_c(j)^2}} \qquad for\ i = 1,2,\dots,3 \qquad \dots(1)$$

The normalized histograms from all three channels were concatenated to form a 96-dimensional feature vector:

$$F_{hist} = \left[\widehat{H}_R, \widehat{H}_G, \widehat{H}_B\right] \in R^{96} \qquad \dots(2)$$

This feature vector describes (in a compact and interpretable way) the color composition of the image. Color histograms are especially relevant for fake face detection, since images generated by GANs often display abnormal color distributions or have unrealistic smoothness, because of their imperfect modeling of lighting and texture. These deviations can be subtle but are often captured by the statistical structure of the histograms. The extracted histogram features were subsequently assessed for their class discriminability by dimensionality reduction techniques (PCA, t-SNE and UMAP) in order to demonstrate how well real and fake images are separated in the feature space.

3.2.2 Edge Features (Canny + Sobel)

Edge detection is a vital approach used in image analysis to detect the drastic intensity changes that usually reflect object boundaries or structural transitions. Edge-based features are also useful for fake face detection, as they can capture subtle inconsistencies in facial contours and textural gradients that the generative models tend to introduce. Examples include overly smooth boundaries, blurred transitions, or unnatural sharpness in fake images.

Grayscale Conversion

To simplify edge analysis, the original RGB image $I_{rgb} \in R^{150\times150\times3}$ is first converted to a grayscale image $I_{rgb} \in R^{150\times150\times3}$ using the standard luminance formula:

$$I_{gray}(x,y) = 0.2989 \cdot R(x,y) + 0.5870 \cdot G(x,y) + 0.1140 \cdot B(x,y) \quad \dots (3)$$

Sobel Edge Detection

The Sobel operator is applied to the grayscale image to compute gradient approximations in the horizontal (G_x) and vertical (G_y) directions:

$$G_x = I_{gray} * S_x, \qquad G_y = I_{gray} * S_y \quad \dots (4)$$

where S_x and S_y are the horizontal and vertical Sobel kernels:

$$S_x = [-1\ 0\ +1\ -2\ 0\ +2\ -1\ 0\ +1], \qquad S_y = [+1\ +2\ +1\ 0\ 0\ 0\ -1\ -2\ -1] \quad \dots (5)$$

The gradient magnitude G_{sobel} is computed as:

$$G_{sobel}(x,y) = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \quad \dots (6)$$

To ensure numerical stability and facilitate comparison, the gradient map is normalized to the range $[0,1]$:

$$\hat{G}_{sobel}(x,y) = \frac{G_{sobel}(x,y) - min\ (G_{sobel})}{(G_{sobel})\ - (G_{sobel})\ + \epsilon} \quad \dots (7)$$

where ϵ is a small constant (e.g., $1e-8$) to avoid division by zero.

Canny Edge Detection

The Canny operator identifies edges by detecting local maxima of the intensity gradient. It includes:

- Gaussian smoothing
- Gradient calculation
- Non-maximum suppression
- Hysteresis thresholding

This process yields a binary edge map $G_{canny} \in \{0,1\}^{150\times150}$

Feature Concatenation and Vector Construction

Both Sobel and Canny outputs are flattened and concatenated to form a single feature vector:

$$F_{edge} = \left[vec\left(\hat{G}_{sobel}\right), vec(G_{canny})\right] \in R^{2\cdot 150^2} = R^{45000} \quad \ldots (8)$$

This edge-based feature vector encodes both gradient intensity and binary edge location information.

Relevance to Fake Face Detection

In fake images, edges can be too soft — caused by over-smoothing — as well as too sharp (due to unnatural transitions), especially around the jawline, eyes and hair boundaries. By capturing these edge patterns, the combined Sobel and Canny features can highlight structural irregularities that are typically challenging for GANs to model accurately. The resulting edge feature vectors were used in downstream analysis and visualized through PCA, t-SNE and UMAP to evaluate their ability to separate real from fake faces in feature space.

3.2.3 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) is a widely used feature of face analysis because it can represent localized texture changes in a fine-grained manner. Local Binary Patterns (LBP) encodes the local structure around each pixel by thresholding neighboring pixel values relative to the center pixel and converting the result into a binary number. That is especially useful with the task of picking up the subtle texture irregularities which are so common in many synthetically generated faces.

LBP Operator Definition

Given a grayscale image $I_{gray} \in R^{H\times W}$, the LBP value for a central pixel (x, y) is computed by comparing it to P surrounding pixels located on a circle of radius R. In this study, the commonly used configuration $P = 8, R = 1$ is adopted.

The LBP code at position (x, y) is defined as:

$$LBP_{P.R}(x, y) = \sum_{p=0}^{P-1} \quad s(g_p - g_c) \cdot 2^P \quad \ldots (9)$$

Where:

- $g_c = I_{gray}(x, y)$ is the intensity of the center pixel.
- g_p is the intensity of the p-th neighbor pixel on the circle,
- $s(\cdot)$ is a thresholding function defined as:

$$s(x) = \{1 \quad if\ x \geq 0\ 0 \quad otherwise \quad \ldots (10)$$

This operation transforms the local neighborhood into an 8-bit binary number, which is then converted into a decimal value. The result is a 2D map of LBP codes representing local texture patterns throughout the image.

Histogram Construction and Normalization

To convert the LBP codes into a usable feature vector, a histogram of LBP values is computed over the entire image:

$$H(i) = \#\{(x, y)|LBP_{P,R}(x, y) = i\}, \quad i = 0,1, \ldots, P + 2 \quad \ldots (11)$$

In the "uniform" LBP variant used here, only patterns with at most two transitions (from 0 to 1 or vice versa) in the circular binary sequence are considered "uniform." This reduces the histogram dimension to $P + 2 = 10\ bins\ for\ P = 8$.

The histogram is then normalized to produce the final texture feature vector:

$$\widehat{H}(i) = \frac{H(i)}{\sum_{j=0}^{P+2} H(j)} \quad for\ i = 0,1,\dots,P+2 \qquad \dots(12)$$

This yields a normalized 10-dimensional feature vector $F_{LBP} \in R^{10}$, which compactly represents the texture distribution across the image.

Relevance to Fake Face Detection

LBP is especially useful in identifying discrepancies of skin texture and surface patterns, which can occur either due to improper GAN modeling or post-processing artifacts. Fake images frequently don't have the micro-texture variations common to real facial regions. The LBP descriptor captures these inconsistencies well, hence can be seen as a discriminative feature that differentiates between true and simulated faces. The normalized LBP histogram vectors were subsequently analyzed using PCA, t-SNE and UMAP to evaluate class separability in lower-dimensional spaces.

3.2.4 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a spectral transformation method in the frequency domain in which spatial representation takes the sum of cosine functions moving at different frequencies. DCT is very popular in image compression and analysis since it reduces the bulk of the power of the image to some small low-frequency components. In fake face detection, in light of the image frequency characteristics, a few artifacts and disparities not readily detected in the spatial domain may be found.

Block-Wise DCT Transformation

For implementation of DCT, images are first converted to grayscale in order to accept DCT:

$$I\_gray \in R^{\wedge}(H \times W) \qquad \dots(13)$$

The grayscale image is then partitioned into non-overlapping blocks of size 8×8. For each block $B \in R^{8\times8}$, the 2D DCT is applied:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} B(x,y) \cdot cos\left[\frac{\pi(2x+1)u}{16}\right] \cdot cos\left[\frac{\pi(2y+1)v}{16}\right] \dots(14)$$

where:

- $u, v \in \{0,1,\dots,7\}$
- $\alpha(k) = \begin{cases} \sqrt{\frac{1}{8}} & if k = 0 \\ \sqrt{\frac{1}{8}} & if\ k > 0 \end{cases}$

This produces a DCT coefficient matrix $C \in R^{8\times8}$, where each coefficient represents the contribution of a specific frequency.

Low-Frequency Component Selection

In natural images, the bulk of the visual energy is localized in the low-frequency region (top-left corner of the DCT matrix). Thus, for each block, we retain only the top-left 4×4 low-frequency coefficients and create a compact 16-dimensional feature vector per block:

$$F\_block = vec(C[0:4,0:4]) \qquad \dots(15)$$

All block-level vectors from the entire image are concatenated and optionally averaged to yield the final global DCT feature vector F_DCT.

Why Frequency Domain Matters in Fake Detection

Due to the nature of GAN architectures and training procedures, synthetic images display unnatural frequency distributions. They either suppress high-frequency noise too aggressively or fail to replicate natural frequency transitions. These irregularities are more subtle when compared with pixel-space analyses, but frequency-domain features can identify discrepancies in their structural consistency and texture synthesis. Focusing on low-frequency components allows us to capture the global structures and smoothness patterns of facial images (two crucial regions where fakes tend to diverge from real photographs). Next, the extracted DCT-based feature vectors were visualized and evaluated for class separability using PCA, t-SNE and UMAP.

3.3 CNN Model Design and Training

To complement the handcrafted feature extraction and enable data-driven learning of discriminative features, a custom Convolutional Neural Network (CNN) was designed and trained for the binary classification task of detecting real vs. fake facial images. Figure 2 shows the proposed model architecture.detecting real vs. fake facial images. Figure 2 shows the proposed model architecture.

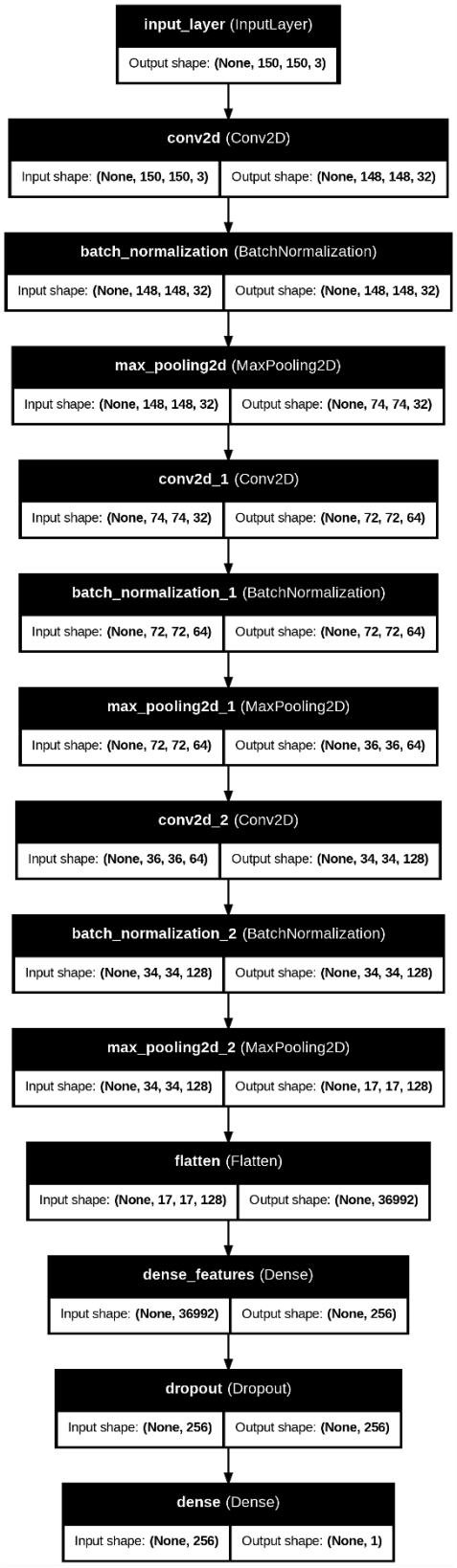

Figure 2 Proposed Model Architecture

3.3.1 Input Configuration

The input images were resized to a fixed spatial resolution of 150 × 150 pixels using three color channels (RGB), yielding an input tensor of shape:

X∈R^(150×150×3). Every pixel was divided by 255 to reach a range [0,1], preserving its numerical stability for training.

3.3.2 Network Architecture

The architecture of the proposed convolutional neural network (CNN) is lightweight, efficient and able to learn deep discriminative features from real and fake images. The network consists of three convolutional blocks with fully connected (dense) layers that are responsible for final classification.

Architectural Overview of Layer-wise Approaches

The network configuration is as follows:

Conv2D (32 filters, kernel of 3×3) → ReLU → Batch Normalization → MaxPooling2D (2×2).

Conv2D (64 filters, kernel of 3×3) → ReLU → Batch Normalization → MaxPooling2D (2×2).

Conv2D (128 filters, kernel of 3×3) → ReLU → Batch Normalization → MaxPooling2D (2×2).

Flatten.

Dense (256 units) → ReLU → Dropout (rate=0.5) → used as feature embedding layer.

Dense (1 unit) → Sigmoid → binary classification output.

This framework allows for progressive feature extraction, dimensionality reduction using convolution and pooling and finally a dense representation appropriate for classification.

Mathematical Representation

Now an input image tensor that is A^((0))=X∈R^(150×150×3) is represented. The l-th convolutional block yields the function:

$$A^{(l)}=\sigma(BN([\![Conv]\!]^{(l)}(A^{(l-1)}))),\ l=1,2,3 \quad \dots (16).$$

Where:

σ(x)=max⁡(0,x) is the ReLU activation function.

BN is short for batch normalization.

⟦Conv⟧^((l)) is the convolutional operation in layer l.

After the convolution, the feature maps are flattened for a vector h∈R^256 which is then passed through a dense layer where dropout is applied to avoid overfitting. The final output ŷ∈(0,1) is computed using the sigmoid activation function to be computed:

$$\hat{y}=1/(1+e^{(-z)}),\ \text{where } z=w^{T}h+b \quad \dots \quad (17)$$

h∈R^256 is the embedding from the penultimate dense layer. w ∈ R^256 and b ∈ R are the weights and bias of the final neuron output. The output ŷ is the odds that the input image is real (i.e., not artificially created). This probability is combined with the binary cross-entropy loss function in training.

3.3.3 Optimization and Loss Function

As the task is binary classification (real vs. fake), the binary cross-entropy loss is imposed:

$$L(y,\hat{y}) = -y\cdot\log(\hat{y}) - (1-y)\cdot\log(1-\hat{y}) \dots (18).$$

Where:

$y \in \{0,1\}$ is the ground truth notation.

$\hat{y} \in [0,1]$ is the predicted probability.

The network was optimised using Adam optimizer that adaptively modifies each parameter's learning rate according to estimates of first and second moments of the gradients.

$$\theta_{(t+1)} = \theta_t - \eta / (\sqrt{\hat{v}_t} + \epsilon) \cdot \hat{m}_t \quad \ldots (19).$$

Where η is the learning rate and $\hat{m}_t$, $\hat{v}_t$ are bias-corrected first and second moment estimates.

3.3.4 Training Configuration

The model was trained with the following parameters:

- Epochs: 150
- Batch Size: 32
- Loss Function: Binary Cross-Entropy
- Optimizer: Adam (default learning rate = 0.001)
- Evaluation Metrics: Accuracy

3.3.5 Validation Strategy

A separate validation set was used to evaluate the model's generalization during training. The model's performance was monitored using:

- Training and validation accuracy
- Training and validation loss

This helped detect overfitting and informed the selection of the best-performing model epoch. A plot of training vs. validation accuracy and loss over epochs was generated to visualize convergence behavior.

3.4 Feature Embedding Extraction from CNN

In addition to its role in classification, the trained Convolutional Neural Network (CNN) serves as a feature extractor, providing deep, learned representations that capture the underlying patterns distinguishing real from fake facial images. These internal activations—known as feature embeddings—are extracted from the network's dense layer prior to the output layer.

Embedding Layer Selection

The feature embedding is obtained from the penultimate dense layer of the CNN, which consists of 256 neurons and uses the ReLU activation function. Formally, the feature embedding is given by:

$$h = \sigma\left(W_1 \cdot Flatten\left(A^{(3)}\right) + b_1\right) \quad \ldots (20)$$

Where:

- $A^{(3)}$ is the output from the third convolutional block,
- $Flatten(\cdot)$ converts the 3D tensor into a 1D vector,
- $W_1 \in R^{256 \times d}$ is the weight matrix of the dense layer,
- $b_1 \in R^{256 \times d}$ is the bias vector,
- $\sigma(x) = max\ (0, x)$ is the ReLU activation function,
- d is the flattened input size to the dense layer.

This 256-dimensional embedding serves as a high-level, abstract representation of the image, incorporating spatial, textural and semantic cues learned during supervised training.

Feature Extraction Procedure

After training the CNN, a new model is constructed to output the activations of the dense embedding layer. For a given input test image , the CNN is used in inference mode and the forward pass yields the feature embedding:

$$F_{CNN} = Model_{embed}(X) \in R^{256} \quad \ldots(21)$$

This process is repeated for each image in the test set, resulting in a feature matrix , where is the number of test samples.

Discriminative Nature of CNN Embeddings

These embeddings represent learned, discriminative features because the network has been trained to minimize classification loss between real and fake classes. As a result, the dense layer captures patterns that are most relevant for distinguishing between the two classes, such as:

- Texture realism,
- Facial symmetry,
- Global structure,
- Anomalies in edge smoothness or contrast.

Unlike handcrafted features, CNN embeddings are automatically optimized during training and adaptively capture both local and global information relevant to the task.

To evaluate their separability, the extracted embeddings were visualized using dimensionality reduction techniques—PCA, t-SNE and UMAP—alongside handcrafted features, providing insight into the structure of the learned feature space

3.5 Dimensionality Reduction and Visualization

To interpret and compare the discriminative capacity of both handcrafted features and CNN-based embeddings, dimensionality reduction techniques were applied. These methods transform high-dimensional feature vectors into two-dimensional (2D) representations, enabling visual analysis of class separability between real and fake face images.

The following techniques were used: Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP). Each was applied independently to all feature types extracted in this study.

3.5.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear technique that projects high-dimensional data onto a lower-dimensional subspace by maximizing variance. For a feature matrix (where is the number of samples and is the feature dimension), PCA transforms the data using:

$$Z = X \cdot W \quad \ldots(22)$$

Where:

- $W \in R^{D\times 2}$ contains the top two eigenvectors of the covariance matrix $\Sigma = \frac{1}{N}X^T X$,
- $Z \in R^{N\times 2}$ is the 2D representation of the data.

PCA was applied to each feature set separately (Color Histogram, Edge, LBP, DCT and CNN embeddings). It provides an initial linear visualization and serves as a baseline for comparison with non-linear methods.

3.5.2. t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a non-linear technique designed to preserve local neighborhood structure in the projected space. It converts high-dimensional distances into conditional probabilities, then minimizes the Kullback–Leibler divergence between the high and low-dimensional distributions.

Let $x_i, x_j \in R^D$ be two feature vectors. The similarity in high-dimensional space is given by:

$$p_{j|i} = \frac{exp\ (-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \quad exp\ (-||x_i - x_j||^2/2\sigma_i^2)} \qquad \ldots (23)$$

These are symmetrized to obtain p_{ij}. In the low-dimensional space, similarities are modeled using a Student t-distribution:

$$q_{ij} = \frac{(1 + ||z_i - z_j||^2)^{-1}}{\sum_{k \neq l} \quad (1 + ||z_k - z_l||^2)^{-1}} \qquad \ldots (24)$$

The cost function to minimize is:

$$C_{t-SNE} = \sum_{i \neq j} \quad p_{ij}\ log\ log\ \frac{p_{ij}}{q_{ij}} \qquad \ldots (25)$$

t-SNE was applied with the following parameters:

- Perplexity: 30
- Learning Rate: 200
- Iterations: 1000

t-SNE is particularly effective in showing how well features group similar instances together in 2D space, making it suitable for evaluating class-wise clustering.

3.5.3 Uniform Manifold Approximation and Projection (UMAP)

UMAP is a manifold-based learning method which preserves the local and global structure in dimension reduction. It makes a weighted k-nearest-neighbor graph and uses fuzzy set theory to optimize the low-dimensional embedding. Based on feature vectors {x_1,…,x_N}, UMAP builds a topological representation by estimating the probability of edge connections using a distance-based kernel. It then finds the low-dimensional embedding {z_1,…,

$$C_{UMAP} = \sum_{(i,j)} \quad w_{ij}\ log\ log\ \frac{w_{ij}}{w'_{ij}} + \left(1 - w_{ij}\right)\ log\ log\ \frac{1 - w_{ij}}{1 - w'_{ij}} \qquad \ldots (26)$$

Where w_{ij} and w'_{ij} represent the fuzzy connectivity in high and low-dimensional spaces, respectively.

UMAP was configured with:

- n_neighbors: 15
- min_dist: 0.1
- metric: Euclidean

Unlike PCA and t-SNE, UMAP can maintain both global data structure and local clusters, making it a valuable complement for evaluating both separability and data topology. All three dimensionality reduction techniques were used on handcrafted features (Color Histogram, Edge, LBP, DCT) and deep features (CNN embeddings). The resulting 2D plots were analyzed to:

• Assess visual class separability (real vs. fake),

• Compare the discriminative power of each feature set,

• Guide the design of future hybrid models.

These visualizations provide intuitive insights into how well different feature types can distinguish between real and synthetic facial images.

4. Results

In this section, visual and analytical results were presented for handcrafted and CNN-based features employing three dimensionality reduction methods: PCA, t-SNE and UMAP. The different methods were applied independently on five feature types, yielding a total of fifteen visualizations. These visualizations facilitate qualitative comparisons of class separability between real and fake face images, revealing dimensions of discrimination for each feature set.

4.1 PCA-Based Visualizations

4.1.1 Color Histogram

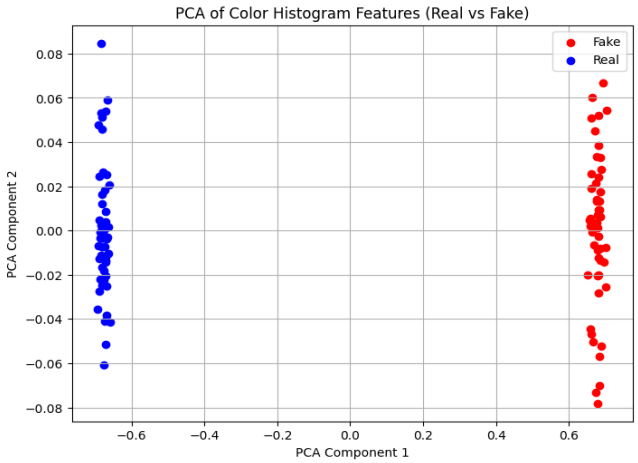

Figure 3 PCA of Color Histogram Features (Real vs Fake)

Figure 3 PCA of Color Histogram Features indicates that color histogram characteristics are discriminative between real and fake faces. The almost ideal linear separation suggests that GAN-generated images have statistically different color distributions (e.g., tone smoothness, saturation, or color balancing) compared to real face images. The robust separation tells us that:

• Fake faces can have smoother transitions of the color or synthetic lighting artifacts.

• Real faces offer more natural ranges of tone and texture owing to authentic lighting and skin characteristics.

4.1.2 Edge Features

This PCA visualization confirms that edge features can help distinguish real vs fake faces. This is likely explained by how generative models like GANs:

• Do not duplicate the fine-grained structural transitions in real images.

• Introduce overly smooth or unnaturally sharp boundaries which can be handled by the Canny and Sobel filters.

These edge inconsistencies—particularly with respect to facial contours, eyes and hairlines—are well-represented in the gradient and edge magnitude maps, therefore high between-class separability.

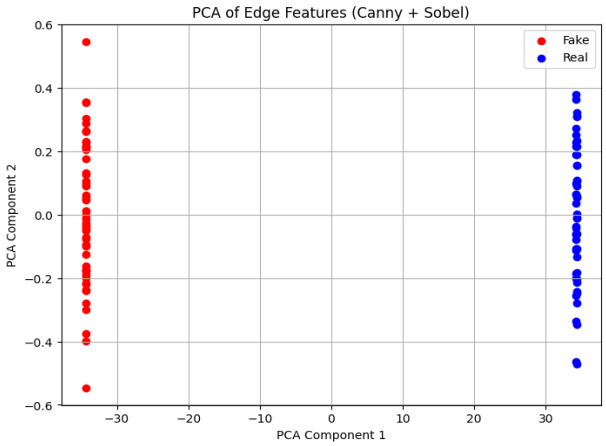

Figure 4 PCA of Edge Features

4.1.3 Local Binary Patterns

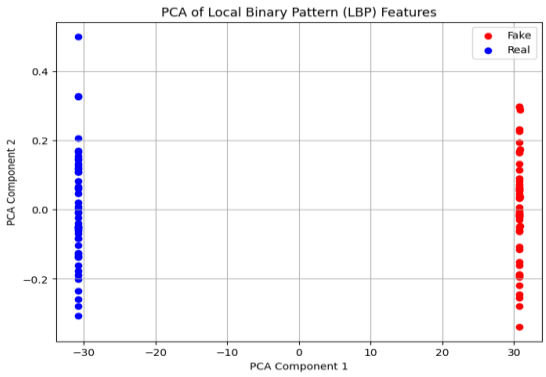

Figure 5 PCA of Local Binary Pattern

LBP features are useful for detecting texture-level irregularities in synthetic images—such as:

• Over-smoothing or unnatural repetition of textures,

• Absence of micro-patterns on skin, hair, or other facial regions.

Because GAN-generated faces often lack natural fine-grained details due to imperfect training or architectural limits, LBP histograms produce distinguishable patterns that are easily captured by PCA. The result is strong linear separability, as illustrated by figure 5.

4.1.4 Discrete Cosine Transform

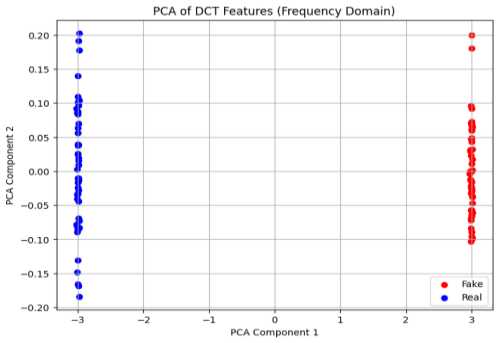

Figure 6 PCA of DCT Features

Figure 6 PCA of DCT features are highly discriminative between real and fake samples in the frequency domain, PCA proved to be good for understanding this. This separation indicates DCT features could be valuable in classification tasks, like deepfake detection or authenticity verification. Plot of real and fake samples shown in figure 6.

4.1.5 Proposed Model

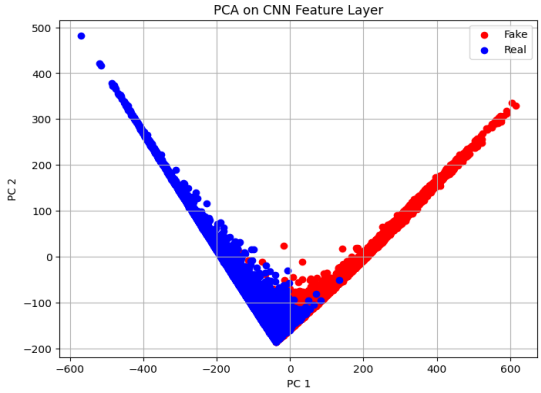

Figure 7 PCA on Proposed Model

Figure 7 indicates the proposed model has learned highly discriminative features to separate real and fake samples. This two-dimensional separation is preserved even in an integrated (PC1 and PC2) setting, suggesting that the proposed model embedding space is an effective and structured separation between classes. Comparing it with our previous techniques by PCA plot:

• The obtained model-based output presents a richer and more complicated representation with superior class separation.

• The shape shows nonlinearity in the data which is well modeled.

4.2 t-SNE-Based Visualizations

4.2.1 Color Histogram

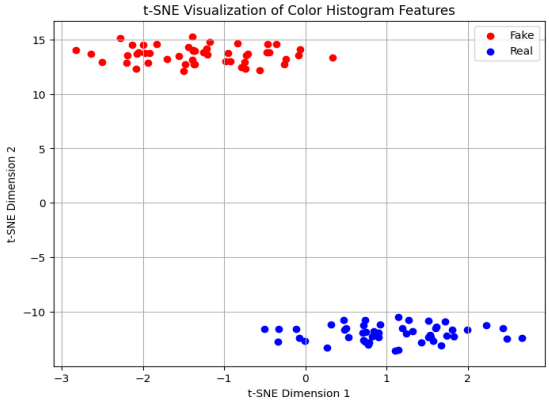

Figure 8 t-SNE Visualization of Color Histogram

Figure 8 Color histogram features are relatively simple but they appear to have fairly strong discriminative power between fake and real images. The t-SNE plot points out that these features, when visualized non-linearly, reveal a highly separable structure – great for making a classification. Compared to PCA, which emphasizes global structure, t-SNE focuses more on local structure and manifold preservation, making it especially good at visualizing this kind of strong class separation.

4.2.2 Edge Features

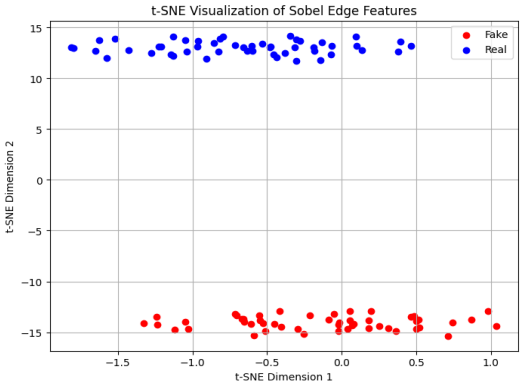

Figure 9 t-SNE Visualization of Edge Features

Figure 9 of edge features, which highlight edges and structural transitions in images, are highly effective in separating real and fake samples in this dataset. The difference could stem from the fact that fake images, such as GAN-generated images, cannot replicate edge continuity or texture fidelity compared to real ones. This suggests that edge-based analysis could play a strong role in detecting tampered or generated content.

4.2.3 Local Binary Pattern

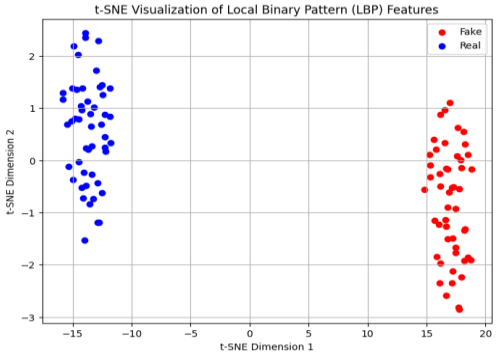

Figure 10 t-SNE Visualization of Local Binary Pattern

The clear separation as seen in Figure 10 shows that LBP features are very discriminative between real and fake samples of the dataset. LBP records fine-grained texture patterns and since fake images are not always able to create authentic textures in a truly realistic manner, LBP is very popular for detecting deepfake or synthetic media.

4.2.4 DCT Features

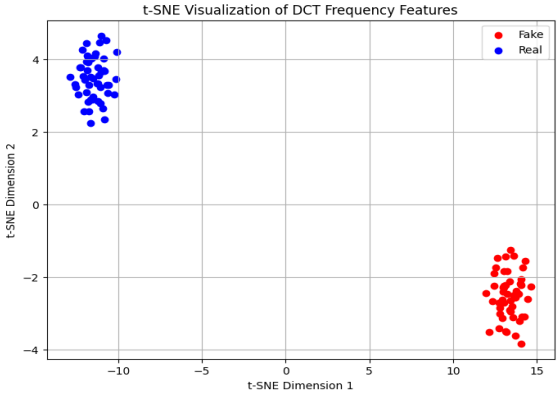

Figure 11 t-SNE Visualization of DCT Features

Frequency-domain properties are recorded in Figure 11 of DCT and can be used to detect artifacts such as loss due to image compression or manipulation. DCT becomes a prominent feature for fake detection since synthetic (fake) images often have different frequency distributions than natural (real) ones. It is established that: • There is a wide, clear separation between the fake and real clusters on the t-SNE dimensions, both of which are significant. • Both of these two clusters have very little internal variation which indicates a high intra-class consistency. • Given the spatial separation, it can be deduced that DCT features are very good at separating real vs. fake images.

4.2.5 Proposed Model

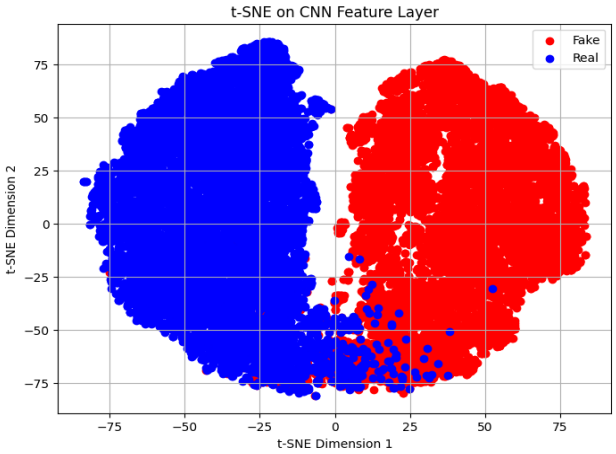

Figure 12 t-SNE on Proposed model

The proposed model is able to learn discriminative features which distinguish between fake and real images well, which is shown in Figure 12. The CNN-based feature representation provides a much better separation in the t-SNE projection as compared to standard features (e.g., color histograms, Sobel edges, or LBP). This figure 12 shows that:

• Distinct Clusters: There is a very clear and clean separation between fake (red) and real (blue) samples.

• High Density: Both classes are densely packed with minimal overlap, indicating strong intra-class compactness and inter-class separability.

• Large Scale Representation: The spread of data across a wide range in both dimensions suggests that the CNN captures complex and rich features.

4.3 Umap-Based Visualizations

4.3.1 Color Histogram

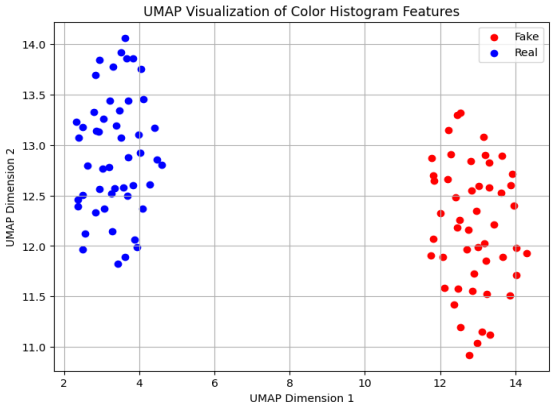

Figure 13 UMAP Visualization of Color Histogram Features

The Figure 13 visualization indicates that color histograms are effective at separating the fake images from the real ones in this dataset. UMAP captures the underlying manifold structure well and successfully visualizes the data in a 2D space, where classes are easily separated. This is promising, particularly because color histograms are straightforward, low-complexity features. Figure 13 shows that:

• Well Structured Bimodal Distribution: The red (fake) and blue (real) points form two well-separated vertical clusters along the UMAP Dimension 1 axis.

• Minimal Overlap: There is little to no cross-over between the two classes indicating clear class separation based only on color distribution.

• Tight Clusters: Each class is grouped tightly, showing consistent color patterns within each class.

4.3.2 Edge Features

Figure 14 visualization indicates Sobel edge features are very effective in distinguishing fake from real images in the dataset. The UMAP projection indicates that edge details — e.g., gradients and contours — significantly differ between the categories. Figure 14 shows that:

• Strong Cluster Separation:

- fake and real samples are clearly separated, both horizontally and vertically.
- No overlap exists between the two groups, indicating highly discriminative edge patterns between the two classes.

• Compactness: Both classes form tight clusters, which suggests consistency in Sobel edge features within each class.

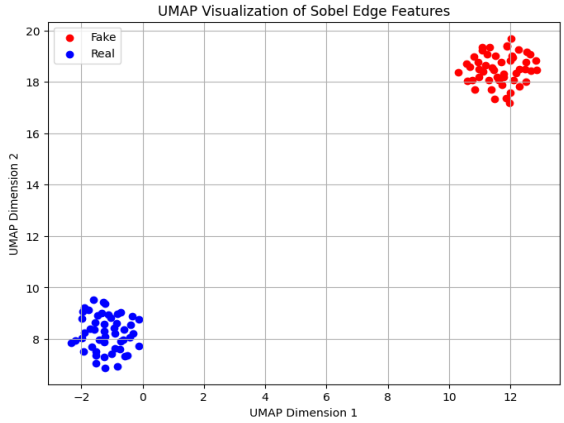

Figure 14 UMAP Visualization of Edge Features

4.3.3 LBP Texture

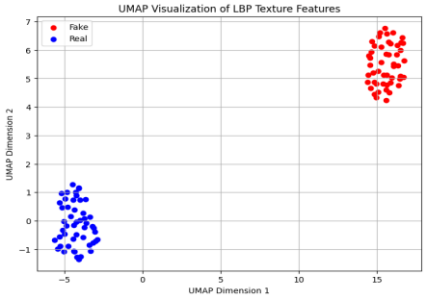

Figure 15 UMAP Visualization of LBP Texture Features

UMAP plot (Figure 15) shows that LBP texture features are capable of successfully separating fake from real instances, which would make it appropriate for fake image detection and authenticity verification. The observations as shown in Figure 15 are as follows:

• The fake and real dots form two well-separated clusters.

• This strong separation shows that the LBP features are highly discriminative, meaning they distinguish between real and fake samples.

• There is no significant overlap between the two classes, which is a positive indicator with regards to feature quality and potential classification accuracy.

4.3.4 DCT Features

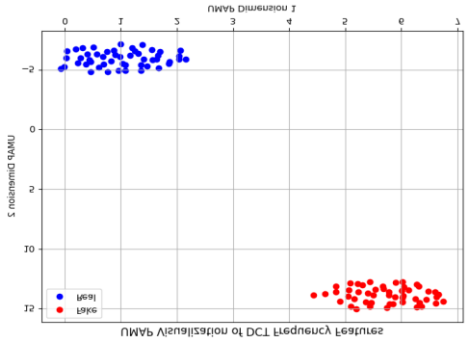

Figure 16 UMAP Visualization of DCT Features

Figure 16 of UMAP visualization shows that there is a very good class separation between fake and real samples with the DCT frequency features. This makes DCT useful in applications such as image forensics, deepfake detection and media authenticity verification. Interpretation of Figure 16:

• The red and blue clusters are well distinguished with:

o Top-right quadrant: Fake samples (red) clustering.

o Bottom-left quadrant: Real samples (blue) clustering.

• This large split indicates that the DCT frequency features are effective for separating real vs. fake data very well.

• There is virtually no overlap between the samples of either group, so:

o High discriminative power.

o Possibility of good classification performance with these features.

4.3.5 Proposed Model

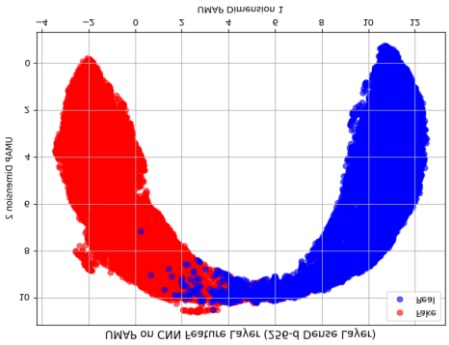

Figure 17 UMAP on Proposed Model

Figure 17 UMAP visualization shows that the proposed model features can reasonably distinguish between fake and real samples.

• Both classes show good separation overall, especially at the ends of the arc-shaped manifold.

• This indicates that the CNN feature layer correctly captures separability.

• The arc-like shape suggests that UMAP preserved a curved structure in the latent space and the CNN is extracting non-linear patterns from the data.

4.4 Analysis of Proposed Model Performance

The visualization results in Sections 4.1-4.3 can clearly demonstrate that the proposed Convolutional Neural Network (CNN) technique is superior to traditional handcrafted features in regards to discriminative performance. Throughout the three dimensionality reduction methods PCA, t-SNE and UMAP, features extracted by the CNN form clear, discrete clusters that discriminate a real face from a fake face, indicating high degree of intra-class cohesion and inter-class separability.

4.4.1 Feature Learning Capability

To that end, the proposed CNN was trained end-to-end on labeled real and fake images, allowing it to learn hierarchical representations corresponding to classification. In contrast to handcrafted methods that treat low-level descriptors (e.g., edge gradients, color histograms, texture codes) exclusively, the CNN captures a multitude of spatial, textural and semantic cues, including:

• Global structural integrity (e.g., facial symmetry and geometry),

• Specific texture patterns (e.g., skin irregularities and microtextures) as detail representations,

• Contextual relationships (e.g., the spatial arrangement of facial features).

This is represented through its dense layer embeddings, which, compared visually, show minimal intra-class variance and strong inter-class margins.

4.4.2 Comparison with Handcrafted Features

Handcrafted features (LBP, Edge, DCT, color histograms, etc.) were able to achieve decent separation of classes in visual space, but they were bounded by different domain assumptions and lack the ability to adjust to new settings. Overall, the CNN embeddings perform better than handcrafted descriptors on an overall basis when compared to:

• Dimensionality-reduced space separation, where closed clusters and minimal overlap with CNN embeddings continued to co-occur;

• Complex feature representation, providing the model with the ability to pull apart nonlinear, task-specific features that handcrafted models could not generalize to;

• Robustness to slight variations (lighting inconsistencies, facial deformations, color shifts), typical of synthetic images. As a result, we know that handcrafted features are better in interpretability and computational cost, but lack the precision of intricate visual cues necessary for high-accuracy deepfake detection.

4.4.3 Effective Class Discrimination

The embedding layer of CNN contains a latent space where the samples of real and fake faces are linearly and non-linearly separable, preserving them through PCA, t-SNE and UMAP projections. It is interesting to observe that the t-SNE and UMAP plots show little class overlap as well as high intra-class compactness, which are strong signals for strong feature discrimination. Such findings confirm that the model not only learned visual patterns by memory, but generalized the underlying differences in distribution between real and GAN-generated images.

5. Conclusion And Future Scope

5.1 Conclusion

Here we provide comprehensive features, both handcrafted and deep learning-based, to differentiate the two classes of facial images. We focus attention on the development of interpretable and robust deepfake detection

algorithms for distinguishing real and fake facial images. On the basis of the publicly available "140k Real and Fake Faces" dataset, we extracted and evaluated many handcrafted factors: color histograms, Sobel and Canny edge features, LBP, Discrete Cosine Transform (DCT). Then, we compared these to embeddings obtained from a custom-built Convolutional Neural Network (CNN). We visualized how the separability pattern is for individual real and fake samples in feature space using dimensionality reduction techniques. The findings unequivocally prove that handcrafted features can provide valuable class separability especially in the texture and frequency domains, although our proposed CNN performs better on feature richness, class separation and robustness. The CNN learned complex nonlinear patterns from an image corpus, which classical techniques are not able to find, and we could then predict authentic facial images more strongly than synthetic facial images. This paper has mentioned that it is significant to come up with explainable and reliable deepfake detection systems to tackle the challenges facing society like misinformation control, protection of digital identity, and maintenance of trust in visual communication systems. These findings may be used as powerful, data-driven guidelines to the construction of the feature-conscious hybrid CNN architecture, in which the most discriminative handcrafted features (color, texture, edge, and frequency) may be introduced in a systematic way and combined with automated CNN embeddings to enhance detection accuracy, interpretability, and generalization to a wide range of deepfake samples. These results are consistent with the feasibility of hybrid models combining handcrafted and learned features, making an effort to combine interpretability of the handcrafted aspects of the model with the capability and flexibility of the learned features.

5.2 Future Scope

Based on the knowledge and effectiveness provided in this work, several additional directions for future studies can be established:

Hybrid feature fusion architectures:

Future work should take a hybrid CNN architecture approach combining handcrafted features (e.g., DCT, LBP) fused at various stages in the network with CNN embeddings. These designs can be used to enhance performance and interpretability in low-resource or real-time environments.

Generalization across Datasets:

To test the possibility of practically deploying upcoming models, it is desirable to test their generalization performance across different datasets (e.g., FaceForensics++, Celeb-DF) to test how well they generalize across novel GAN architectures, as well as post-processing artifacts.

Video-Based Detection:

Building on the static image analysis in this work to the temporal models which applies to the deepfake detection task involving video-based deepfake detection (e.g., by employing CNN-LSTM, or 3D CNNs) could make possible some temporal consistency checks to mitigate the forgery transitions at the frame level.

Lightweight and Edge-Compatible Models:

Mobile or edge device compatibility is an important focus for optimization. If we approach the investigation of model pruning, quantization and the application of lightweight descriptors, then deepfake detection may be feasible in real-time, resource-constrained operations.

Explainable AI (XAI) Integration:

It might also provide a more detailed understanding of CNN decision-making processes that could lead to enhancing trust in automated fake image detection systems, in particular in high-stakes sectors such as forensics and biometric authentication, by using explainability frameworks such as Grad-CAM, SHAP, or saliency maps.

Robustness to Adversarial Attacks and Post-Processing:

Addiction to adversarial perturbations and post-processing (compression, filtering, etc.) which is the mainstay to avoid detection approaches should not be exploited by any future models.

Finally, this work provides a comprehensive basis for creating interpretable, efficient, highly discriminative fake image detection systems and indicates a clear path for future developments in the field of visual media authentication.

Reference

1. Passos, L. A., Jodas, D., Costa, K. A., Souza Júnior, L. A., Rodrigues, D., Del Ser, J., ... & Papa, J. P. (2024). A review of deep learning-based approaches for deepfake content detection. Expert Systems, 41(8), e13570.
2. Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., & Yu, N. (2021). Multi-attentional deepfake detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 2185-2194).
3. Xu, Y., Raja, K., Verdoliva, L., & Pedersen, M. (2023). Learning pairwise interaction for generalizable deepfake detection. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 672-682).
4. Lin, Y., Song, W., Li, B., Li, Y., Ni, J., Chen, H., & Li, Q. (2024, September). Fake it till you make it: Curricular dynamic forgery augmentations towards general deepfake detection. In European Conference on Computer Vision (pp. 104-122). Cham: Springer Nature Switzerland.
5. Chen, L., Zhang, Y., Song, Y., Liu, L., & Wang, J. (2022). Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 18710-18719).
6. Lai, Y., Yu, Z., Yang, J., Li, B., Kang, X., & Shen, L. (2024). Gm-df: Generalized multi-scenario deepfake detection. arXiv preprint arXiv:2406.20078.
7. Saikia, P., Dholaria, D., Yadav, P., Patel, V., & Roy, M. (2022, July). A hybrid CNN-LSTM model for video deepfake detection by leveraging optical flow features. In 2022 international joint conference on neural networks (IJCNN) (pp. 1-7). IEEE.
8. Zhang, D., Li, C., Lin, F., Zeng, D., & Ge, S. (2021, August). Detecting Deepfake Videos with Temporal Dropout 3DCNN. In IJCAI (pp. 1288-1294).
9. Al-Dhabi, Y., & Zhang, S. (2021, August). Deepfake video detection by combining convolutional neural network (cnn) and recurrent neural network (rnn). In 2021 IEEE international conference on computer science, artificial intelligence and electronic engineering (CSAIEE) (pp. 236-241). IEEE.
10. Elhassan, A., Al-Fawa'reh, M., Jafar, M. T., Ababneh, M., & Jafar, S. T. (2022). DFT-MF: Enhanced deepfake detection using mouth movement and transfer learning. SoftwareX, 19, 101115.
11. Khalil, S. S., Youssef, S. M., & Saleh, S. N. (2021). iCaps-Dfake: An integrated capsule-based model for deepfake image and video detection. Future Internet, 13(4), 93.
12. Ismail, A., Elpeltagy, M., S. Zaki, M., & Eldahshan, K. (2021). A new deep learning-based methodology for video deepfake detection using XGBoost. Sensors, 21(16), 5413.
13. Yasir, S. M., & Kim, H. (2025). Lightweight Deepfake Detection Based on Multi-Feature Fusion. Applied Sciences, 15(4), 1954.
14. Du, M., Pentyala, S., Li, Y., & Hu, X. (2020, October). Towards generalizable deepfake detection with locality-aware autoencoder. In Proceedings of the 29th ACM international conference on information & knowledge management (pp. 325-334).
15. Shao, R., Wu, T., Nie, L., & Liu, Z. (2025). Deepfake-adapter: Dual-level adapter for deepfake detection. International Journal of Computer Vision, 1-16.
16. Yan, Z., Zhang, Y., Fan, Y., & Wu, B. (2023). Ucf: Uncovering common features for generalizable deepfake detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 22412-22423).
17. Guan, J., Zhou, H., Hong, Z., Ding, E., Wang, J., Quan, C., & Zhao, Y. (2022). Delving into sequential patches for deepfake detection. Advances in Neural Information Processing Systems, 35, 4517-4530.
18. Shuai, C., Zhong, J., Wu, S., Lin, F., Wang, Z., Ba, Z., ... & Ren, K. (2023, October). Locate and verify: A two-stream network for improved deepfake detection. In Proceedings of the 31st ACM International Conference on Multimedia (pp. 7131-7142).
19. Wang, J., Wu, Z., Ouyang, W., Han, X., Chen, J., Jiang, Y. G., & Li, S. N. (2022, June). M2tr: Multi-modal multi-scale transformers for deepfake detection. In Proceedings of the 2022 international conference on multimedia retrieval (pp. 615-623).
20. Waseem, S., Abu-Bakar, S. A. R. S., Omar, Z., Ahmed, B. A., Baloch, S., & Hafeezallah, A. (2023). Multi-attention-based approach for deepfake face and expression swap detection and localization. EURASIP Journal on Image and Video Processing, 2023(1), 14.

21. Guan, J., Zhou, H., Hong, Z., Ding, E., Wang, J., Quan, C., & Zhao, Y. (2022). Delving into sequential patches for deepfake detection. Advances in Neural Information Processing Systems, 35, 4517-4530.

22. Wang, J., Sun, Y., & Tang, J. (2022). LiSiam: Localization invariance Siamese network for deepfake detection. IEEE Transactions on Information Forensics and Security, 17, 2425-2436.

23. Pu, W., Hu, J., Wang, X., Li, Y., Hu, S., Zhu, B., ... & Lyu, S. (2022). Learning a deep dual-level network for robust DeepFake detection. Pattern Recognition, 130, 108832.

24. Ganguly, S., Ganguly, A., Mohiuddin, S., Malakar, S., & Sarkar, R. (2022). ViXNet: Vision Transformer with Xception Network for deepfakes based video and image forgery detection. Expert Systems with Applications, 210, 118423.

25. Chakraborty, R., & Naskar, R. (2024). Role of human physiology and facial biomechanics towards building robust deepfake detectors: A comprehensive survey and analysis. Computer Science Review, 54, 100677.

26. Akhtar, Z., Pendyala, T. L., & Athmakuri, V. S. (2024). Video and audio deepfake datasets and open issues in deepfake technology: being ahead of the curve. Forensic Sciences, 4(3), 289-377.

27. Le, B., Tariq, S., Abuadbba, A., Moore, K., & Woo, S. (2023). Why do deepfake detectors fail. arXiv preprint arXiv:2302.13156.

28. Shree, M. S., Arya, R., & Roy, S. K. (2024). Investigating the Evolving Landscape of Deepfake Technology: Generative AI's Role in it's Generation and Detection. Int. Res. J. Adv. Eng. Hub (IRJAEH), 2, 1489-1511.

29. Bendiab, G., Haiouni, H., Moulas, I., & Shiaeles, S. (2025). Deepfakes in digital media forensics: Generation, AI-based detection and challenges. Journal of Information Security and Applications, 88, 103935.

30. Ren, S., Xu, H., Ng, T., Zewde, K., Jiang, S., Desai, R., ... & Muthukrishnan, R. (2025). Do Deepfake Detectors Work in Reality?. arXiv preprint arXiv:2502.10920.

31. Tekam, E. R., Nasare, R., Divtelwar, G., Upadhye, P. V., Wadyalkar, M. R., & Khobragade, P. (2025). Detection of deepfake images using hybrid convolutional neural networks and transformer models. In Proceedings of the 2025 IEEE 1st International Conference on Smart Innovations in Systems, Infrastructure, Mechanical, Power, AI and Computing Technologies (SISIMPACT) (pp. 876–881). IEEE. https://doi.org/10.1109/SISIMPACT67725.2025.11439691