Comprehensive Analysis of Manual and CNN-based Feature Extraction for Deepfake Detection: A Comparative Study on Celeb-DF Dataset

Abstract—**In this paper, we compare manual feature extraction on the dataset of Celeb-DF and Deepfake Detection Techniques with CNN-based methods. We gathered 30 hand-crafted features incorporating geometric, illumination, texture, color, and frequency properties, with qualitative but measurable variations observed among feature classes. Our experiment compares ResNet-50, ResNet-152, ResNet-101, and Vision Transformer (ViT) models, and we compare their feature representation with PCA analysis and t-SNE visualization. ResNet-50 achieved better efficiency with a maximum class separation of 1.53 in PCA analysis and 89.6% variance explanation, and ViT achieved low efficiency with only a 0.12 separation. The extraction pipeline of the facial mask created by the U-Net and the ResNet-50 backbone ran the synthetically created and the original videos of 795 and 158 efficiently. The outcomes of the current study illuminate the comparative effectiveness of learned and hand-crafted features in deepfake doubt and reiterate the need for the development of the evolution of the authentication technologies related to the media.**

Index Terms—**Deepfakes Detection, CNN-feature extraction, Manual feature extraction, PCA-analysis, t-SNE visualization, Celeb-DF dataset**

1. Introduction

Rapid improvement in deepfake techniques created major challenges in trustworthiness across media, as well as online security. Advanced generative adversarial networks' deepfakes and face-swap apps are serious threats to information genuineness, the privacy of citizens, and trust within societies. As such media are becoming increasingly realistic and widespread, it is all the more necessary that effective solutions for their detection are developed.

Current deep forgery detection methods are even categorized into the following two categories: traditional visual detection methods based on the use of hand-crafted features and deep visual detection methods based on the utilization of CNNs. Even though their possible performance was demonstrated by deep learning-based schemes, an understanding of the contribution of the hand-crafted feature remains useful in the design of understandable and reliable deepfake detection systems.

Our work advances of deepfakes by giving an detailed study over the hand-designed choice of feature techniques versus CNN-learned features. Our work gives comparisons among 30 hand-derived features covering geometric, illumination, texture, color, and frequency features versus deep features for the best-performing CNN models for four. Our comparison, through conducting the deep experiments on the data from Celeb-DF, adds an objective feature discriminability and evaluation of model performance.

The new findings from the current research work are as follows: (1) detailed feature study of 30 hand-crafted features with quantitative comparative studies of synthetic and realistic content, (2) Comprehensive comparison of the deep-detect models of four CNN models (ResNet-50, ResNet-152, ResNext-101, and ViT), (3) Comprehensive dimension reduction studies with PCA and t-SNE and measures of separation, and (4) Extracting mask from face pipeline with optimized for the detection of synthetic video.

2. Problem Statement

Rapid advances in the creation also highlighted the need for effective techniques of detection. Active deepfake detection limitations are:

Feature Interpretability: Most deep learning models are black boxes since it is unclear whether visual inputs influence decisions to detect. Human-annotated features provide understandable information regarding what features make up the difference between real-world and forged content. Our entire 30-feature controlled analysis provide precise details regarding what elements of synthetic content are most resistant to the deepfake generation process.

Model Generalization: Some models are able to generate diverse deep features. Some comparisons among architectures are useful so that methods for feature extraction will be better understood.

Dimensionality and Visualization: Effective techniques of dimensionality reduction are also required to see and understand the complex feature structures of CNN. The features of clusters of diverse models in low-dimensional space show their separability and their visualization.

Dataset Bias and Evaluation: Proper assessment implies careful analysis of data distribution, class discrimination, as well as model performance. To solve such issues, the paper presents a whole mode of experimentation of CNN and manual methods in both quantitative experiments as well as visualization results for their performance.

3. Literature Review

Deepfake detection is highly advanced and the treatment is from more traditional computer vision schemes to newer deep learning architectures.

3.1. Manual Feature-based Approaches

Early schemes were of a hand crafted, feature oriented type. In a paper by Li et al. [1] we find the introduction of using physiological signal inconsistencies and out of Matern et al. [2] that examined the use of face landmark analysis and head pose. Also it was introduced that color features and thus the HSV color model do introduce some what of a solution to synthesis artifact detection [3]. Geometric features including the eye aspect ratio and the ratios of the face were examined by Yang et al. [4] and they reported on small problems of Deepfake production.

Also we find the use of texture analysis including Local Binary Patterns (LBP) and Discrete Cosine Transform (DCT) coefficients to be introduced as a solution for detecting compression artifacts and synthesis related problems [5]. However in large scale these solutions fail miserably infront of high end deepfakes and fail in reality to generalize with many methods of generation.

3.2. CNN-based Approaches

The application of deep learning in deepfakes detection has registered impressive breakthroughs in accuracy and robustness. Rossler et al. [6] put forward the database of faceforensics++ and substantiated that ResNet and XceptionNet architecture performs well.

Attention mechanism and transformer-based models gained popularity, and Zhao et al. [7] indicated potential findings with Vision Transformers. Latest works explored ensemble methods with the application of multiple CNN architecture [8], temporal consistency verification [9], and multiple scale feature learning [10].

3.3. Dimensionality Reduction and Visualization

PCA and t-SNE are applied extensively to feature analysis in computer vision applications. Van der Maaten and Hinton [11] proposed t-SNE to visualize high-dimensional data, and PCA continues to be an essential method to interpret feature variance and separability [12]. Their application to features of deepfakes gives insight into model performance and class distributions.

4. Dataset Description

4.1. Celeb-DF Dataset Overview

Celeb-DF is our baseline benchmark data, which is a collection of real and synthetically-generated video clips targeted for use in deepfake detection studies. The data presents a comprehensive testbed for performance evaluation across algorithms in separating real and fabricated facial videos.

4.2. Dataset Structure and Composition

The dataset is separated into separate directories:

- **Celeb-real**: 158 real celebrity videos
- **Celeb-synthesis**: 795 synthesized deepfake videos

4.3. Identity Distribution and Synthesis Matrix

The data contains 15 unique identities in real-world videos (id0 to id14) and 16 identities in synthetic-world videos (id0 to id17, with only id15 being observed as targets). It is a deepfake pattern of creation that is measured through a 16×16 synthesis fusion matrix, with indicated sparse pairing relationships that involve few identity combinations.

The data contains 953 videos with 158 real-world and 795 synthetic samples, spread through more than 15 real-world and 16 forgery-world identities. It is synthetic content creation that is not uniformly distributed in source-target pairs, as is typical with manifold manipulation strategy variations. The results are typical of the manifold as well as with data skewness.

4.4. Class Distribution

The data has a very balanced ratio of real to fake videos. There are a total of 52.6% (501) real and 47.4% (452) fake. The training set also has the same ratio with 52.0% (4,973) real and 48.0% (4,588) fake examples. Validation contains 50.4% (670) real and 49.6% (660) fake videos for unbiased evaluation.

The test set comprises a higher number of actual videos at 55.9% (1,561) than 44.1% (1,231) of forged ones. Frame-level extraction from multiple frames per video is sufficient for training and evaluation. .

5. Methodology

5.1. Manual Feature Extraction

We identified a total of 30 full manual features within geometric, illumination, texture, color, and frequency features. Each feature was created to address special qualities of facial genuineness and synthesis artifacts that are typically compromised during deepfake production.

1) Geometric Features (Facial Landmark Based)

Geometric features from landmarked faces provide basic information regarding the proportions and configurations of faces, usually distorted in manipulated video footage. Eight descriptors derive such differences, for example, the Eye Ratio for abnormal eye openness, the Eye-to-Width Ratio for orientation, and the Height-to-Width and Eye-Mouth Ratios for irregular structures.

Some other descriptors, Eye Distance, Face Width and Height, and the Eye-Mouth Distance, provide scale-invariant dimensions and locations. Together, the metrics reveal subtle irregularities, amenable to proper identification of fabricated or manipulated facial material. Algorithm 1 shows the process of extracting the facial landmark features.

Algorithm 1 Facial Landmark Feature Extraction

Input: I: Input facial image of dimensions $H \times W$, M: Binary facial mask

Output: $\mathbf{f}_L$: Landmark feature vector of dimension 7×1

```
*Initialisation:*
Apply facial mask: $I_{masked} \leftarrow I \odot M$
Convert to grayscale:
$I_{gray}[x, y] = \frac{0.299 \cdot R[x,y] + 0.587 \cdot G[x,y] + 0.114 \cdot B[x,y]}{255}$, $\forall x, y \in [1, H] \times [1, W]$
Initialize Haar cascade classifier for face detection.
*Face Detection Process:*
Detect faces using cascade classifier:
$\mathcal{F} \leftarrow \text{detectMultiScale}(I_{gray}, \text{scaleFactor} = 1.1, \text{minNeighbors} = 4)$
if $|\mathcal{F}| = 0$ then
    Find contours: $\mathcal{C} \leftarrow \text{findContours}(M, \text{RETR\_EXTERNAL})$
    Extract bounding box: $(x, y, w, h) \leftarrow \text{boundingRect}(\arg\max_{c \in \mathcal{C}} \text{Area}(c))$
else
    $(x, y, w, h) \leftarrow \mathcal{F}[0]$
end if
*Landmark Estimation:*
$x_{\text{left\_eye}} \leftarrow x + 0.2w$, $x_{\text{right\_eye}} \leftarrow x + 0.5w$
$y_{\text{eye}} \leftarrow y + 0.25h$, $y_{\text{mouth}} \leftarrow y + 0.65h$
*Distance Calculations:*
$d_{\text{eye}} \leftarrow x_{\text{right\_eye}} - x_{\text{left\_eye}}$
$d_{\text{eye\_mouth}} \leftarrow y_{\text{mouth}} - y_{\text{eye}}$
*Feature Extraction:*
$f_1 \leftarrow \frac{d_{\text{eye}}}{w}$, $f_2 \leftarrow \frac{h}{w}$, $f_3 \leftarrow \frac{d_{\text{eye\_mouth}}}{h}$
$f_4 \leftarrow d_{\text{eye}}$, $f_5 \leftarrow w$, $f_6 \leftarrow h$, $f_7 \leftarrow d_{\text{eye\_mouth}}$
$\mathbf{f}_L \leftarrow [f_1, f_2, f_3, f_4, f_5, f_6, f_7]^T$
*Final Output:*
return $\mathbf{f}_L$
```

2) Illumination Features

Four illumination features capture light consistency and direction:

Light Direction: Overall lighting direction calculated through gradient analysis, since synthetic faces can have inconsistent lighting patterns.

Direction Consistency: Measures the consistency of light direction across facial regions.

Highlight-Shadow Ratio: Bright to dark region ratio, finding unnatural lighting patterns that are typical in deepfakes.

Illumination Uniformity: In the facial area, the standard deviation of the lighting range measures the consistency of illumination.

Algorithm 2 shows the process of extracting the illumination features:

Algorithm 2 Illumination Feature Extraction

Input: I: Input facial image of dimensions $H \times W$, M: Binary facial mask

Output: $\mathbf{f}_I$: Illumination feature vector of dimension 4×1

```
Apply facial mask: $I_{masked} \leftarrow I \odot M$
Convert to LAB color space: $I_{LAB} \leftarrow \text{BGR2LAB}(I_{masked})$
Extract luminance channel: $L \leftarrow I_{LAB}[:, :, 0]$
*Gradient Computation:*
$G_x \leftarrow \text{Sobel}(L, 1, 0, \text{ksize} = 5)$
$G_y \leftarrow \text{Sobel}(L, 0, 1, \text{ksize} = 5)$
$\theta \leftarrow \text{arctan2}(G_y, G_x) \times \frac{180}{\pi}$                    (gradient angle)
*Highlight and Shadow Detection:*
$T_{high} \leftarrow \text{percentile}(L[M > 0], 95)$
$H_{mask} \leftarrow \{L > T_{high}\} \cap M$
$T_{low} \leftarrow \text{percentile}(L[M > 0], 20)$
$S_{mask} \leftarrow \{L < T_{low}\} \cap M$
*Feature Calculations:*
$V \leftarrow \{(x, y) : M[x, y] > 0\}$
if $|V| > 0$ then
    $f_1 \leftarrow \text{std}(\theta[V])$                    (direction consistency)
    $f_2 \leftarrow \frac{\sum H_{mask}}{\max(\sum S_{mask}, 1)}$                    (highlight-shadow ratio)
    $f_3 \leftarrow \max(L[V]) - \min(L[V])$                    (illumination uniformity)
    $S_{sin} \leftarrow \sum_{(x,y) \in V} \sin(\theta[x, y] \frac{\pi}{180})$
    $S_{cos} \leftarrow \sum_{(x,y) \in V} \cos(\theta[x, y] \frac{\pi}{180})$
    $f_4 \leftarrow \text{arctan2}(S_{sin}, S_{cos}) \times \frac{180}{\pi} \bmod 360$                    (light direction)
else
    $f_1, f_2, f_3, f_4 \leftarrow 0$
end if
$\mathbf{f}_I \leftarrow [f_4, f_1, f_2, f_3]^T$
return $\mathbf{f}_I$
```

3) Texture and Edge Features

Two textural features examine structural consistency:

Gradient Magnitude: This feature is calculated with Sobel filters for determining the strength of an image edge. It is distinguishing variability in sharp focus and structural shape between real and composite content. It achieves a good measure of the sharpness of the local texture and edge retention by capturing the amount of the change in intensity throughout pixels.

Local Contrast: This characteristic is computed through an estimation of the standard deviation of the brightness within small local neighborhoods. It is an effective way for characterising small scale variability and detailed textures. Greater local contrast is related to steeper and more compact structure, and reduced contrast is related to smoother and possibly forged regions. This facilitates the differentiation of real and forged textures.

Algorithm 3 shows the steps for extracting the texture and edge features:

Algorithm 3: Texture and Edge Feature Extraction

```
Input: $I$: Facial image of dimensions $H \times W$, $M$: Binary facial mask
Output: $\mathbf{f}_T$: Texture and edge feature vector
1: Apply facial mask: $I_{masked} \leftarrow I \odot M$
2: Convert to grayscale if needed: $I_{gray} \leftarrow \text{Grayscale}(I_{masked})$
3: Compute the gradients using Sobel filters:
$G_x \leftarrow \text{Sobel}(I_{gray}, 1, 0)$
$G_y \leftarrow \text{Sobel}(I_{gray}, 0, 1)$
Gradient magnitude: $M_g \leftarrow \sqrt{G_x^2 + G_y^2}$
4: Local contrast computation:
Divide the image into small neighborhoods
For every neighborhood, $f_{lc} \leftarrow \text{std}(L_{neighborhood})$
5: Aggregate features:
$f_1 \leftarrow \text{mean}(M_g[M > 0])$ (Gradient magnitude)
$f_2 \leftarrow$ mean of local contrast values
6: Assemble feature vector: $\mathbf{f}_T \leftarrow [f_1, f_2]^T$
7: return $\mathbf{f}_T$
```

4) Color Space Features (HSV)

Seven features of HSV color space are analyzed for the color tone and variation across the face images and form discriminatory information for the separation of real and counterfeit faces:

Hue Mean and Standard Deviation: These features calculate the mean hue and its variability across face regions. Synthetic faces typically display small but noticeable movements of color distributions. By investigating the mean and variation of hue, we are able to identify probable synthetic distributions of color associated with manipulated content.

Saturation Mean and Standard Deviation: Saturation is employed for measuring the brightness or intensity of hues. Calculation of the mean and standard deviation of the saturation across face regions can detect unusually uniform or synthetic hue intensity, normally occurring in deepfakes, for the purpose of distinguishing between created and natural textures.

Value Mean and Standard Deviation: Value corresponds to the amount of brightness of an image. Estimation of the mean and the variance of brightness within face areas takes into account the variation of lighting and shadowing. Such minor variations of lighting of synthetic images can not be simulated, and such statistics form a good indicator of such a malfunction.

Color Contrast: Colour contrast examines the disparities between hue and saturation readings across the face areas. Higher contrasts indicate normal variability of colours, and reduced or stable contrasts can reveal manipulated or fabricated photographs.

Algorithm 4 shows the HSV Color Features Extraction process :

Algorithm 4: HSV Color Feature Extraction

```
Input: $I$: Input facial image, $M$: Binary facial mask
Output: $\mathbf{f}_{HSV}$: HSV color feature vector
1: Apply facial mask: $I_{masked} \leftarrow I \odot M$
2: Convert to HSV: $I_{HSV} \leftarrow \text{HSV}(I_{masked})$
3: Split channels: $H_c, S_c, V_c \leftarrow \text{channels}(I_{HSV})$
4: Identify the valid pixels: $P_{valid} \leftarrow \{(x, y) : M[x, y] = 1\}$
5: Compute HSV statistics (if valid pixels exist):
$f_1 \leftarrow \mu(H_c[P_{valid}])$ (Hue mean)
$f_2 \leftarrow \sigma(H_c[P_{valid}])$ (Hue std)
$f_3 \leftarrow \mu(S_c[P_{valid}])$ (Saturation mean)
$f_4 \leftarrow \sigma(S_c[P_{valid}])$ (Saturation std)
$f_5 \leftarrow \mu(V_c[P_{valid}])$ (Value mean)
$f_6 \leftarrow \sigma(V_c[P_{valid}])$ (Value std)
$f_7 \leftarrow f_2 \cdot f_4$ (Color contrast: Hue std × Saturation std)
6: Assemble vector: $\mathbf{f}_{HSV} \leftarrow [f_1, f_2, f_3, f_4, f_5, f_6, f_7]^T$
7: return $\mathbf{f}_{HSV}$
```

5) Color Channel Correlations

Three correlation characteristics analyze HSV channels' relationships to identify discrepancies in color interactions:

HS Correlation: This function computes the correlation between color hue and saturation channels throughout the face. Abnormalities in the normal color tone and intensity relationship tend to occur in fake faces, and therefore this correlation is an effective measure of manipulated or created imagery.

HV Correlation: Hue-value relationship characterizes the correlation of the light and tonal character of the colour. Violations of anticipated correlations in natural faces can signal synthetic fabrication, since the models of deepfakes can lack the usual dependencies of lightness and hue.

SV Correlation: Saturation-value correlation examines the relationship between the brightness and intensity of a color. Simulacra photographs are liable to disrupt these natural interactions, and analysis of the correlation identifies the areas where the color movement diverges from the actual textures of the face..

Algorithm 5 shows the Color Channel Correlation Features Extraction process:

Algorithm 5: Color Channel Correlation Feature Extraction

```
Input: $H_c, S_c, V_c$: HSV channels, $M$: Binary facial mask
Output: $\mathbf{f}_{corr}$: Color channel correlation feature vector
1: Identify valid pixels: $P_{valid} \leftarrow \{(x, y) : M[x, y] = 1\}$
2: Compute correlations (if valid pixels exist):
$f_1 \leftarrow \rho(H_c[P_{valid}], S_c[P_{valid}])$ (HS correlation)
$f_2 \leftarrow \rho(H_c[P_{valid}], V_c[P_{valid}])$ (HV correlation)
$f_3 \leftarrow \rho(S_c[P_{valid}], V_c[P_{valid}])$ (SV correlation)
3: Assemble vector: $\mathbf{f}_{corr} \leftarrow [f_1, f_2, f_3]^T$
4: return $\mathbf{f}_{corr}$
```

6) Skin Consistency Feature

This measure calculates the deviation of the texture and the color only among skin-only pixels, and therefore maintains the analysis isolated and independent of non-skin sectors. In the vast majority of deepfakes, the author inadvertently introduces slight noise, artifacts, or small irregularities within the skin portions of the face, difficult to detect through the naked eye.

Focusing on the differences within skin texture and complexion, the measure effectively retrieves inconsistencies that are the sign of manipulated content. Differences reported by such an approach become good manipulation indicators and therefore become a good authentication and verification measure. In summary, skin-particular texture deviation analysis provides an effective tool for synthetic or manipulated faces identification.

Algorithm 6 shows the process of Skin Consistency Feature Extraction:

```
**Algorithm 6: Skin Consistency Feature Extraction**
Input: $I_{HSV}$: HSV image, $M$: Binary facial mask
Output: $f_1$: Skin consistency feature
Define HSV skin ranges: $R_{skin} = \{[0, 20], [170, 180]\}$ for Hue with suitable $S, V$ bounds
Generate skin mask: $M_{skin} \leftarrow (I_{HSV} \in R_{skin}) \odot M$
Compute skin pixel ratio: $f_1 \leftarrow \frac{|M_{skin}|}{|P_{valid}|}$ (Skin consistency feature, set 0 if denominator $= 0$)
return $f_1$
```

7) Discrete Cosine Transform (DCT) Features

The Discrete Cosine Transform (DCT) provides frequency domain characteristics for differentiation of real and manipulated material. The DCT Mean calculates the overall value of DCT coefficients, showing global frequency characteristics. The DCT Standard Deviation calculates variability within the coefficients, signifying frequency-domain noise introduced by compression or synthesis.

The DCT Range, calculated as the difference of the maximum and the minimum DCT quantities, estimates the degree of frequency elements spread. Finally, the Energy Ratio, as the ratio of synthetic and original frequency energy, capitalizes on the different impact of compression and synthetic generation on frequency subbands and enables the identification of manipulated material. Algorithm 7 shows the DCT Feature Extraction process:

```
**Algorithm 7: DCT Feature Extraction**
Input: $I$: Input facial image, $M$: Binary facial mask
Output: $\mathbf{f}_{DCT}$: DCT feature vector
Apply facial mask: $I_{masked} \leftarrow I \odot M$
Convert to grayscale: $I_{gray}[x, y] = 0.299R[x, y] + 0.587G[x, y] + 0.114B[x, y]$
Divide image into $8 \times 8$ blocks: $N_h \leftarrow \lfloor H/8 \rfloor$, $N_w \leftarrow \lfloor W/8 \rfloor$
Initialize array: $E_{DCT} \leftarrow []$
For each block $(i, j)$:
    Extract block: $B \leftarrow I_{gray}[8i : 8(i + 1), 8j : 8(j + 1)]$
    Extract mask block: $B_M \leftarrow M[8i : 8(i + 1), 8j : 8(j + 1)]$
    If $\sum B_M \geq 32$:
        $B_{DCT} \leftarrow \text{DCT}(B)$
        $e \leftarrow \sum |B_{DCT}|$ (Block energy)
        Append $E_{DCT} \leftarrow E_{DCT} \cup \{e\}$
Compute DCT features (if valid blocks exist):
    $f_1 \leftarrow \text{mean}(E_{DCT})$ (DCT mean)
    $f_2 \leftarrow \text{std}(E_{DCT})$ (DCT std)
    $f_3 \leftarrow \max(E_{DCT}) - \min(E_{DCT})$ (DCT range)
    Sort energies: $E_{sorted} \leftarrow \text{sort}(E_{DCT})$
    Split point: $k \leftarrow \lfloor |E_{sorted}|/2 \rfloor$
    $E_{low} \leftarrow \sum_{i=0}^{k-1} E_{sorted}[i]$
    $E_{high} \leftarrow \sum_{i=k}^{|E_{sorted}|-1} E_{sorted}[i]$
    $f_4 \leftarrow \frac{E_{high}}{E_{low} + 10^{-10}}$ (Energy ratio high/low)
else
    $f_1, f_2, f_3, f_4 \leftarrow 0$
Assemble vector: $\mathbf{f}_{DCT} \leftarrow [f_1, f_2, f_3, f_4]^T$
return $\mathbf{f}_{DCT}$
```

8) Pixel Variance Features

Two variance features inspect pixel intensity patterns:

Variance Mean: Mean variation in pixel intensity across face areas.

Variance Standard Deviation: Standard deviation of the variance, quantifying how patterns of pixels vary across the face. Algorithm 8 illustrates the procedure for extracting Pixel Variance Features:

```
**Algorithm 8: Pixel Variance Feature Extraction**
Input: $I_{gray}$: Grayscale facial image, $M$: Binary facial mask
Output: $\mathbf{f}_{Var}$: Pixel variance feature vector
Identify valid pixels: $P_{valid} \leftarrow \{(x, y) : M[x, y] = 1\}$
Compute the pixel intensity variance per block (or overall region):
    $V_{block} \leftarrow \text{Var}(I_{gray}[P_{valid}])$
Compute variance features:
    $f_1 \leftarrow \text{mean}(V_{block})$ (Variance mean)
    $f_2 \leftarrow \text{std}(V_{block})$ (Variance std)
Assemble vector: $\mathbf{f}_{Var} \leftarrow [f_1, f_2]^T$
return $\mathbf{f}_{Var}$
```

5.2. Facial Mask Extraction Pipeline

To make good quality face masks, we have a landmark-based pipeline, which includes an alternative with the forehead and a smooth combination of alpha.

1) Face Detection and Landmark Localization

The Dlib frontal face detector and 68-point landmark prophet are used to accurately detect the exact detection of important facial structures such as eyes (landmark 36–47), nose (27-35), mouth (48–67), and eyebrows (17–26). These sites provide the ability to detect an accurate position on the face so that important areas of the face can be accurately identified. By introducing such areas, the system area is capable of strengthening the geometric shape of the face for the addition of the region, expression analysis, or manipulation recognition, such as manipulation. This process guarantees high accuracy in identifying vital parts of the face.

2) Forehead Estimation (Optional)

When forehead inclusion is enabled, left and right eyebrow points are utilized to estimate the forehead width and center. A 7-point curved arc over eyebrows is created with:

$$y = y_t - h \cdot [0.8 \cdot 4t(1-t)]$$

where $t \in [0,1]$, y_t is the eyebrow top, and h is a fraction of the face height.

3) Mask Construction and Refinement

All chosen landmarks, including optional forehead points, are accumulated and a convex hull is calculated. Padding is added by growing each point out from the centroid. A binary mask is created by filling the padded hull polygon.

4) Smoothing and Alpha Blending

The binary mask is smoothed with a Gaussian blur and converted to grayscale. The mask is normalized to create an alpha channel. The resulting masked image is achieved through:

$$\text{result} = \alpha \cdot \text{image}$$

This produces a soft-edged, region-specific mask over the face with optional forehead coverage.

5) Key Parameters

PADDING determines the region expansion, and BLUR_KERNEL_SIZE and BLUR_SIGMA determine the smoothing range. FOREHEAD_HEIGHT and FOREHEAD_WIDTH_RATIO determine forehead arc geometry.

Algorithm 9 shows the process of extracting the facial masks:

Algorithm 9 Soft Facial Mask Generation with Forehead

```
Input: $I$: Input facial image of size $H \times W$
$P$: Padding size for mask expansion
$K, \sigma$: Gaussian blur kernel size and sigma
$F_{forehead}$: Forehead inclusion flag (True/False)
$H_{forehead}, W_{forehead}$: Forehead height fraction and width ratio
Output: $I_{masked}$: Masked facial image with soft edges
Initialisation:
Convert image to grayscale: $I_{gray} \leftarrow \text{Grayscale}(I)$
Detect faces using dlib detector: $faces \leftarrow \text{DetectFaces}(I_{gray})$
Create a black mask: $M \leftarrow \text{zeros}(H, W)$
for each $face$ in $faces$ do
    a) Extract facial landmarks: $L \leftarrow \text{PredictLandmarks}(face)$
    b) Select key points: eyes, nose, mouth (landmarks 36–67)
    c) if $F_{forehead} = \text{True}$ then
        i) Include eyebrow points (17–26)
        ii) Compute forehead height: $h_f \leftarrow H_{forehead} \cdot face.height$
        iii) Compute forehead width: $w_f \leftarrow W_{forehead} \cdot (\text{eyebrow right} - \text{eyebrow left})$
        iv) Generate curved forehead points above eyebrows and add to key points
    d) Compute convex hull of all points: $H \leftarrow \text{ConvexHull}(points)$
    e) Expand hull with padding $P$ to get $H_{padded}$
    f) Fill polygon $H_{padded}$ on mask $M$
end for
Apply Gaussian blur to mask: $M_{smooth} \leftarrow \text{GaussianBlur}(M, K, \sigma)$
Convert mask to alpha channel: $\alpha \leftarrow M_{smooth}/255$
Apply soft mask to image: $I_{masked} \leftarrow I \cdot \alpha$
Return $I_{masked}$
```

5.3. CNN Feature Extraction

We consider four CNN models for deep feature extraction:

1) ResNet Architectures

ResNet-50: 50-layer residual network that extracts 2048-dimensional features from the global average pooling layer. Trained on ImageNet with standard normalization.

ResNet-152: Deeper 152-layer version that offers improved representational capacity with 2048-dimensional output.

2) ResNext-101

101-layer network using cardinality-based architecture (32 groups) with enhanced feature diversity via parallel pathway aggregation.

3) Vision Transformer (ViT)

Architecture based on transformers (ViT-B/16) that has been pre-trained on ImageNet-21k and employs attention mechanisms for global context modeling. Features extracted from the [CLS] token representation.

5.4. Dimensionality Reduction and Analysis

1) Principal Component Analysis (PCA)

We use PCA to obtain the first three principal components of each CNN model, calculating variance explained ratios and measures of class separation. Separation is calculated in terms of Euclidean distance between class centroids in reduced space.

2) t-SNE Visualization

t-SNE visualization uses parameters: n_components=3, perplexity=30, n_iter=1000, with PCA initialization for convergence stability. This gives non-linear dimensionality reduction for intricate feature relationships.

6. Experimental Results

6.1. Manual Feature Analysis

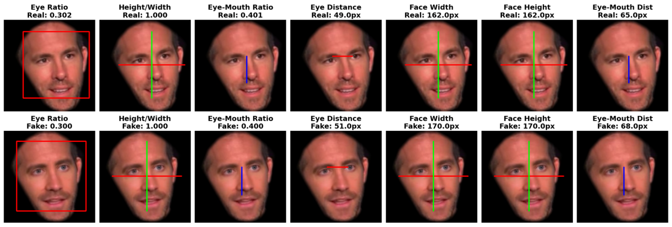

Fig. 1: Manual Feature Importance Ranking

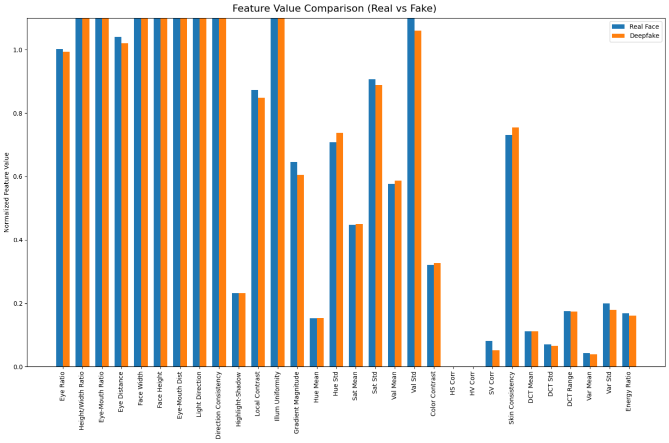

Fig. 2: Manual Feature Value Comparison: Real vs. Synthetic

1) Manual Feature Analysis

The 30 hand-crafted features have differing degrees of discriminative ability between genuine and synthetic faces. Inspection identifies several interesting trends, underlining features to persistently distinguish between genuine and spoofed facial images, with others demonstrating minimal effectiveness, implying a fusion of features is necessary to provide secure detection of synthetic content. Table I shows the Manual Features Analysis (Definitions, Categories, and Comparative Analysis). Figure 3 shows the landmark features extraction and masking mesh of a sample image face.

Geometric Feature Stability: All geometric characteristics exhibit little difference between authentic and artificial content, with values close to normalized 1.00. Eye-Mouth Ratio demonstrates poor discrimination (Real: 1.00, Fake: 0.98), whereas Height-to-Width Ratio detects quantitative variations (Real: 1.05, Fake: 1.03), indicating that contemporary deepfakes preserve face proportion well.

Illumination Pattern Differences: Highlight-Shadow Ratio is a proper discriminator (Real: 0.25, Fake: 0.23), and it indicates that fake faces might not be able to mimic natural light behaviors. The rest of the light features are stable, i.e., there is smart light preservation in the latest deepfake methods.

Texture and Edge Characteristics: Both the Gradient Magnitude (Real: 0.63, Fake: 0.60) and Local Contrast (Real: 0.86, Fake: 0.85) have minor but consistent variations, and therefore texture duplication is an issue with deepfake synthesis.

Color Space Variations: HSV color space attributes are examined in this subsection. Hue Standard Deviation indicates discrimination (Real: 0.70, Fake: 0.74), and Value Standard Deviation indicates large differences (Real: 1.10, Fake: 1.06), thereby suggesting that the color variability pattern is varied for real and fake media.

Correlation Features: Zero in all three color channel correlations (HS, HV, SV) for both the classes, and these can be interpreted as a deliberately planned normalization effect or processing pipeline information in the dataset.

Frequency Domain Insights: It is low for DCT-based features, and for DCT mean value (true: 0.74, forged: 0.76), it suggests equal frequency behavior between true and forged materials, with high-level deepfake generation ability.

Variance Pattern Consistency: Comparable characteristics of pixel variance are the same or nearly so in class-rooms and variance and variance standard deviations, an indication that the pixel is the preservation of the pattern of pattern of intensity that carries into synthetic generation.

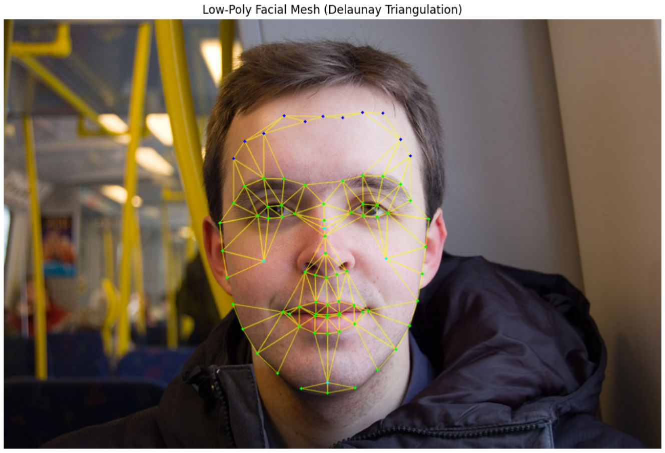

Fig. 3: Landmark Features and Masking (mesh Representation)

TABLE I: Manual Features with Definitions, Categories, and Comparative Analysis

Feature Name	**Category**	**Definition**	**Real Value**	**Fake Value**
Geometric Features (Facial Landmark Based)				
Eye Ratio	Geometric	Ratio of eye width to eye height	1.00	0.99
Eye/Width Ratio	Geometric	Eye width normalized by face width	1.09	1.09
Height/Width Ratio	Geometric	Face height to width ratio	1.05	1.03
Eye-Mouth Ratio	Geometric	Eyes-mouth distance / face height	1.00	0.98
Eye Distance	Geometric	Euclidean distance between eye centroids	1.00	1.00
Face Width	Geometric	Landmark-based face width	1.00	1.00
Face Height	Geometric	Landmark-based face height	1.00	1.00
Eye-Mouth Distance	Geometric	Pixel distance from eyes to mouth center	1.00	1.00
Illumination Features				
Light Direction	Illumination	Dominant lighting direction via gradient	1.00	1.00
Direction Consistency	Illumination	Light direction consistency across face	1.00	1.00
Highlight/Shadow Ratio	Illumination	Ratio of bright to dark regions	0.25	0.23
Illumination Uniformity	Illumination	Standard deviation of lighting across face	1.00	1.00
Texture and Edge Features				
Gradient Magnitude	Texture	Edge strength via Sobel filters	0.63	0.60
Local Contrast	Texture	Standard deviation in local neighborhoods	0.86	0.85
Color Space Features (HSV)				
Hue Mean	Color	Average hue across facial regions	0.15	0.16
Hue Standard Deviation	Color	Hue variability across face	0.70	0.74
Saturation Mean	Color	Average color saturation	0.45	0.45
Saturation Std	Color	Saturation variation	0.89	0.89
Value Mean	Color	Average brightness	0.58	0.59
Value Standard Deviation	Color	Brightness variation	1.10	1.06
Color Channel Correlations				
HS Correlation	Correlation	Hue-saturation correlation	0.00	0.00
HV Correlation	Correlation	Hue-value correlation	0.00	0.00
SV Correlation	Correlation	Saturation-value correlation	0.00	0.00
Skin and Texture Consistency				
Skin Consistency	Texture	Std dev in skin-only regions	0.09	0.08
Color Contrast	Color	Overall color contrast measure	0.33	0.33
Frequency Domain Features (DCT)				
DCT Mean	Frequency	Average DCT coefficient magnitude	0.74	0.76
DCT Standard Deviation	Frequency	Variation in DCT coefficients	0.12	0.12
DCT Range	Frequency	Max - min DCT values	0.07	0.07
Energy Ratio	Frequency	High vs low frequency energy ratio	0.00	0.00
Pixel Variance Features				
Variance Mean	Variance	Average variance in pixel intensity	0.17	0.17
Variance Std	Variance	Standard deviation of variance	0.19	0.19

6.2. PCA Analysis Results

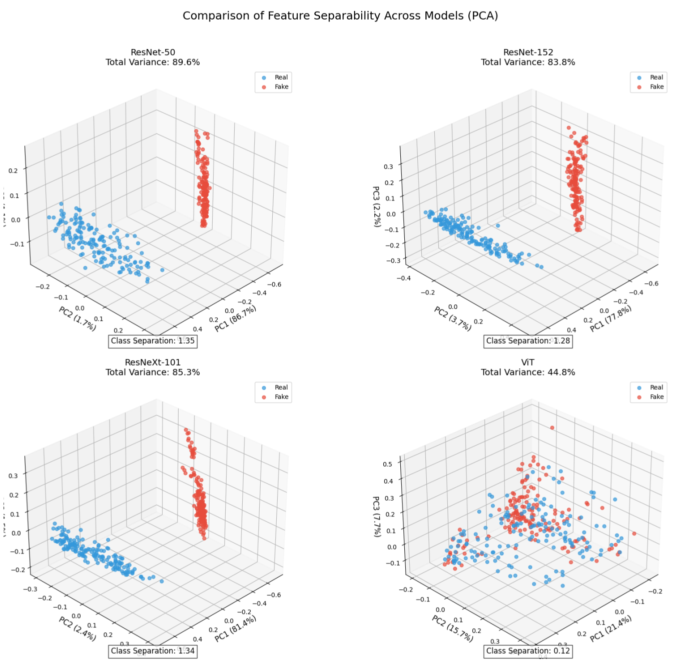

Fig. 4: PCA Visualization Results for All CNN Architectures

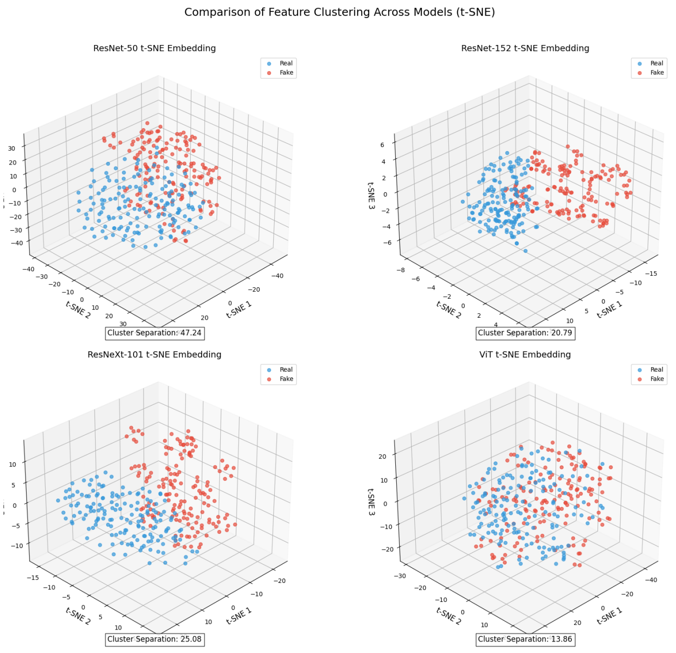

Fig. 5: t-SNE Visualization Results for CNN Features

1) Variance Explanation Analysis

ResNet-50 reveals the largest cumulative variance (89.6% of which PCA reports 86.7% of that variance, indicating that the features reside highly on the first component.ResNet-152 is equivalent (83.8% cumulative but somewhat weaker PCA performance. ResNet-101 accounts for more of the variance that is present (85.3% with a 61.4% of the PCA, and we see a more refined feature relationship. Vit reports the smallest variance (44.8% of which is a note that facilities are balanced between the components. Figure 4 is PCA visualization results for all of the CNN architectures.

2) Class Separation Analysis

Class distinguishing measures signify radical performance differences:

- **ResNet-50:** Shows the best discrimination (1.53) in PCA space.
- **ResNet-152:** Displays consistent splitting (1.28) across component pairings.
- **ResNext-101:** Resists dissociating in dissociation (1.34) with cardinal advantages.
- **ViT:** Exhibits weak separation (0.12), indicating low discriminative capability

Figure 5 shows t-SNE Visualize Result for CNN Features while Table-II shows the PCA Analysis (Variance Distinctness).

TABLE II: PCA Analysis (Variance & Separation)

Model	View	PC1	PC2/3	Sep.
ResNet-50	PC1-2	86.7	8.9	1.35
	PC1-3	86.7	1.7	1.53
	3D	86.7	8.9/1.7	1.35
ResNet-152	PC1-2	86.6	8.7	1.28
	PC1-3	86.6	1.7	1.28
	3D	86.6	8.7/1.7	1.28
ResNext-101	PC1-2	61.4	8.4	1.34
	PC1-3	61.4	2.1	1.34
	3D	61.4	8.4/2.1	1.34
ViT	PC1-2	21.4	15.7	0.12
	PC1-3	21.4	3.5	0.12
	3D	21.4	15.7/3.5	0.12

7. Discussion

7.1. Manual vs. CNN Features

Comparative study determines the complementing features of manually extracted methods and CNN. Manually extracted features provide interpretable results, in their case, Light Direction and color-space features provide discriminative power. Complimentary features, however, of CNN include increased separability in multi-dimensional feature spaces.

Low geometric importance of features implies that the recent deepfakes were better at preserving face proportions and thus more advanced techniques for picking up subtle artifacts of the color and texture spaces are required.

7.2. CNN Architecture Comparison

ResNet-50 is found as the best architecture for the task through a tradeoff of model complexity and quality of features. That it is better than ResNet-152 means more depth leads to overfitting or feature redundancy for the deepfakes detection purpose.

Competitive results of ResNet-152 demonstrate the advantage of cardinality for learning diverse and rich representations of features. Under-performing ViT showcases that the attention mechanisms are not inherently better for the task of detection, possibly the consequence of the character of the database or structural limitations.

7.3. Dimensionality Reduction Insights

PCA analysis indicates that the form of good deepfake-detector features is generally dense on principal components and PCA dominance is observable for ResNet models. This implies that linear transformations can efficiently recover discriminative information of key importance.

The excited variance patterns imply that the low-parameter models (ResNet-50) could allow more feature representation condensation, while the higher complexity models (ViT) diffuse the information through many dimensions and thus reduce their discriminability.

8. Conclusion and Future Work

It is a deep analysis that provides insightful observations of the comparative efficiency of CNN-inherent and manually generated features for deepfakes detection. Our findings indicate that the best performance of ResNet-50 is with improved class discrimination (1.53) and explanation of variance (89.6%), and the manually generated features contain interpretable information in which Light Direction is the most discriminatory feature.

Systematic comparison of the different architectures of CNNs is such that complexity of the model is not necessarily a guarantee of performance, for instance, in ResNet-50 model as compared to deeper models. Poorer-than-suboptimal performance of the ViT testifies to the importance of the choice of architecture for specific tasks.

It is reserved for destiny works to study collection methods that combine CNN and hand-crafted functions, sum up video frames for temporal modeling, and test comprehensive performance on a series of datasets in order to generalize well. Deepfake-oriented interest mechanism study and adversarial robustness research on one-of-a-kind types of heterogeneous abilities are also mandates for advanced detection performance.

The comparative survey thus as outlined and quantitative measures offer the promise to develop deepfake detection as a domain and provide methodological contributions and applied values for future research.

Acknowledgment

The authors express their gratitude to the Celeb-DF dataset owners for releasing such a valuable dataset for research on deepfakes detection.

Declarations

Conflict of interest: The authors confirm that they have no conflict of interest regarding the publication of this manuscript.

Authors' contributions: All authors have made equal contributions to the development of this work and have reviewed and approved the final version of the manuscript.

Informed consent: Informed consent was obtained from all participants included in the study.

Funding: This research was not supported by external or internal funding sources.

Data Availability: The authors declare that no associated data is available for this research article.

References

[1] Y. Li, M. C. Yang, and S. J. Zhang, ”In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking,” in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1-7.

[2] F. Matern, C. Riess, and M. Stamminger, ”Exploiting visual artifacts to expose deepfakes and face swaps,” in *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019, pp. 83-92.

[3] H. H. Nguyen, J. Yamagishi, and I. Echizen, ”Capsule-forensics: Using capsule networks to detect forged images and videos,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2307-2311.

[4] X. Yang, Y. Li, and S. Lyu, ”Exposing deep fakes using inconsistent head poses,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8261-8265.

[5] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, ”Leveraging frequency analysis for deep fake image recognition,” in *International Conference on Machine Learning*, 2020, pp. 3247-3258.

[6] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, ”Faceforensics++: Learning to detect manipulated facial images,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1-11.

[7] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, ”Multi-attentional deepfake detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2185-2194.

[8] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. C. Ferrer, ”The deepfake detection challenge (dfdc) dataset,” *arXiv preprint arXiv:2006.07397*, 2020.

[9] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, ”Deepfake video detection through optical flow based cnn,” in *IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2019, pp. 1205-1207.

[10] S. Tariq, S. Lee, H. Kim, Y. Shin, and S. S. Woo, ”Detecting both machine and human created fakes,” *arXiv preprint arXiv:1910.01717*, 2019.

[11] L. Van der Maaten and G. Hinton, ”Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579-2605, 2008.

[12] I. T. Jolliffe and J. Cadima, ”Principal component analysis: a review and recent developments,” *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, p. 20150202, 2016.