BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **22**, No 3
Sofia • 2022 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2022-0032

A Color-Texture-Based Deep Neural Network Technique to Detect Face Spoofing Attacks

Mayank Kumar Rusia, Dushyant Kumar Singh

Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, Uttar Pradesh, India
E-mails: mayank.qip18@mnnit.ac.in dushyant@mnnit.ac.in

Abstract: *Given the face spoofing attack, adequate protection of human identity through face has become a significant challenge globally. Face spoofing is an act of presenting a recaptured frame before the verification device to gain illegal access on behalf of a legitimate person with or without their concern. Several methods have been proposed to detect face spoofing attacks over the last decade. However, these methods only consider the luminance information, reflecting poor discrimination of spoofed face from the genuine face. This article proposes a practical approach combining Local Binary Patterns* (*LBP*) *and convolutional neural network-based transfer learning models to extract low-level and high-level features. This paper analyzes three color spaces* (*i.e., RGB, HSV, and YCrCb*) *to understand the impact of the color distribution on real and spoofed faces for the NUAA benchmark dataset. In-depth analysis of experimental results and comparison with other existing approaches show the superiority and effectiveness of our proposed models.*

Keywords: *Presentation attack detection, biometrics, computer vision, deep learning, authentication, color-texture analysis.*

1. Introduction

Nowadays, a face recognition system poses a significant challenge to authenticate the face identity of a legitimate user due to a face spoofing attack. Face spoofing attack (a.k.a. face presentation attack) is an illegal attempt performed by an imposter to get the face access of a legitimate user with or without their knowledge. However, face identity theft may also be possible if the legitimate users arrange the spoofed face under an agreeable condition. In both these cases, face spoofing can breach security mechanisms resulting in duplicity of face identity. Among other biometric traits, the human face preserves vital information to identify individuals; thus, a recent surge has been noticed in face biometric-based authentication applications worldwide [1, 2].

On the other hand, spoofing a human face is a comparatively more straightforward task than other biometric traits as it requires only a legitimate user's recaptured frame (i.e., photos or videos). These frames can be represented before the biometric sensors to bypass the face evidence. However, this recapturing (gamut or reproduction) process generates a slightly distorted image compared to the original for various reasons such as different color distribution [3], lighting effects, camera focus, printing material (specular reflection), and display media such as digital photo, print photo, video display, and more. Therefore, the presented face suffers from medium-based color-dependent spoofing.

1.1. Motivation

The motivational factors that inspired us to do this research work are summarized below:

- The chrominance information is more promising than luminance information to discriminate the recaptured image from the original image [4].
- A combined dual-phase (i.e., handcrafted and deep) feature extraction approach at different levels for distinct color distributions has received scant attention in the literature. The absence of such analysis in the literature made the intention of this research work empirically focused, which this article efficiently accomplishes.
- Grayscale images are not appropriate for analyzing the fine details of spoofed and genuine faces, particularly for low-resolution images.
- The printed photo attack is a widely used, most convenient, low-cost face spoofing attempt for imposters.

Conventional machine learning methods relying only on handcrafted features are no more effective in dealing with face spoofing problems because of color disparity found on various presentation attack instruments. To overcome this limitation, we propose a new approach combining the Color-based LBP and Transfer Learning-based methods such as VGG16, InceptionV3, and MobileNetV2 to achieve remarkable performance.

1.2. Contribution

Our significant contributions to this research work can be outlined as follows:

- We explore three distinct color spaces (i.e., RGB, HSV, and YC_rC_b) involving luminance and chrominance information to analyze the intrinsic disparities between different color distributions on the recaptured images.
- We offer a new face image preprocessing consisting of face detection, face alignment, normalization, and image enhancement.
- We apply channel splitting on three different color spaces to effectively analyze the features of each component of the color space.
- We propose a color-based LBP descriptor to extract the color-texture (local level) features channel-wise, as it considers luminance and chrominance information at different locality levels. However, the color-based LBP feature descriptor alone has been insufficient to provide extensive learning to the model as this paper considers various scenarios of face spoofing attacks aligned directly with real-life applications and problems.
- We deploy a modified CNN architecture by utilizing the pretrained weights for the three different transfer learning models (i.e., VGG16, InceptionV3, and

MobileNetV2), with fine-tuning of hyperparameters to obtain more appropriate (deep) features in second-level feature extraction.

- We have conducted extensive experiments on a public benchmark NUAA dataset, exploring the effect of various face spoofing scenarios, such as print attacks (handhold photos), display attacks (digital photos), wrapped photos, and blurry photos under predefined experiment criteria.
- We provide a tabular comparative analysis with other state-of-the-art approaches to reveal that our proposed approach outperforms other countermeasure techniques.

1.3. Organization

This paper is organized into five sections. Section 1 delineates the rationale behind this work with solid motivations to detect face spoofing attacks. A list of abbreviations and their meanings used in this manuscript is shown in Table 1. Section 2 discusses the literature review for the state-of-the-art antispoofing techniques. Section 3 illustrates the proposed methodology given two novel techniques for feature extraction and subsequent classification using deep neural networks. Section 4 reports the experimental results with an appropriate analysis of the results. The complete work is summarized with future scope in Section 5.

Table 1. List of abbreviations with their meaning

Abbreviation	Meaning	Abbreviation	Meaning
PAIs	Presentation Attack Instruments	CLBP	Color-based Local Binary Pattern
LBP	Local Binary Pattern	LGS	Local Graph Structure
CNN	Convolutional Neural Network	NUAA	Nanjing University of Aeronautics and Astronautics
SVM	Support Vector Machines	VGG	Visual Geometry Group
EER	Equal Error Rate	CASIA FASD	Chinese Academy of Science Face Anti-Spoofing Database
EDDTCP	Extended Division Directional Ternary Co-relation Pattern	MSU MFSD	Michigan State University Mobile Face Presentation Attack
LTP	Local Ternary Pattern	HSV	Hue, Saturation, Value
CNNTL	Convolutional Neural Network-based Transfer Learning	YCrCb	Luminance, Chrominance red, Chrominance blue

2. Literature review

Given the fragility of authentication systems, the problem of detecting face spoofing has always attracted the attention of the biometric research community. However, published articles are limited in their scope. This section presents contemporary insight into state-of-the-art methods for distinguishing between genuine and spoofed faces. Here, we present the recently proposed machine learning and deep learning-based approaches by analyzing the features of distinct color space and textures for different public benchmark face spoofing datasets such as NUAA, Replay-Attack, CASIA FASD, and MSU MFSD [2, 5]. The importance of selecting the best feature selection method to determine efficient results is reviewed in [6].

Thomas and Mathew [7] propose a face spoofing detection approach utilizing local binary patterns and support vector machines for feature extraction and classification purposes, respectively. This work analyzes the color texture, image distortion, and image quality for HSV color space. Anand and Vishwakarma [8] proposed a constructive fusion approach to address the face spoofing problem involving LBP and CNN for feature extraction and the SVM method to classify the spoofed and real faces. The LBP-based method obtained an accuracy of 94.31 %, while the CNN-based VGG16 method yielded an accuracy of 98.15 % on the CASIA FASD dataset. Chen et al. [9] proposed a face spoof detection method utilizing a rotation-invariant LBP and ResNet-18 model for color texture feature extraction with SVM classification. The principal component analysis is used to reduce the dimension of the feature space. The NUAA, Replay-Attack, MSU-MFSD, and CASIA-FASD datasets are considered for the experiments. The best EER results for YCrCb and HSV are 0.37%, 3.2%, 5.9%, and 4.1% with NUAA, Replay Attack, CASIA-FASD, and MSU-MFSD datasets, respectively. Boulkenafet, Komulalnen and Hadid [4] introduce the color-texture-based face spoofing detection method, which analyzes the extracted low-level features from different color spaces. Edmunds and Alice [10] propose a model able to retrieve the radiometric distortions from the images. The drawback is that it cannot detect distortion when variable illumination is present for enrolment and authentication.

The state-of-the-art literature confirms that color texture is the essential factor primarily considered for detecting face spoofing attacks. Furthermore, the NUAA dataset is used mainly in state-of-the-art research due to many frames (i.e., 12600) consisting of various unconstrained scenarios such as the live face, handhold print photo, digital photo, blurred and over-exposed images. On the other hand, deep neural network solutions yield remarkable results for face spoofing detection. Thus, this paper primarily considers these facts to propose a new methodology.

3. Proposed methodology

Considering the findings of the literature section, we propose a new dual-stream feature extraction-based countermeasure technique for face spoofing detection by analyzing three distinct color spaces. The proposed architecture of face spoofing detection is depicted in Fig. 1.

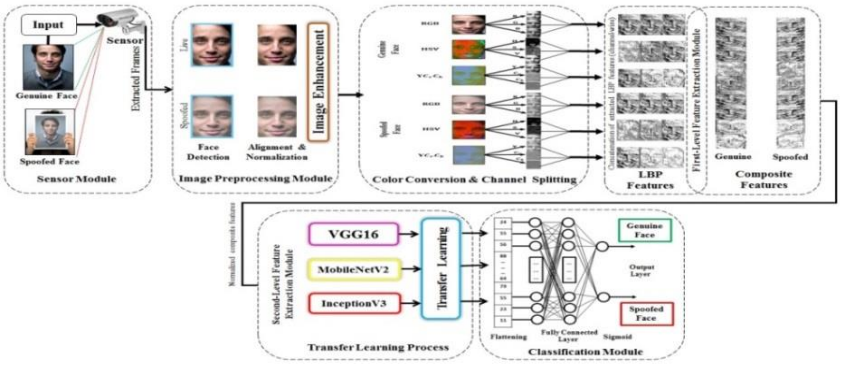


Fig. 1. Architecture of the proposed face spoofing detection approach

Our approach being proposed involves three distinct color spaces: RGB, HSV, and YCrCb. These three color spaces contain more intuitive information that can be retrieved through the channel-splitting process. Thus, each color space is segregated into its respective channels. The color-based LBP feature descriptors efficiently extract the knowledge from these disparate channels and concatenate it to generate a composite feature vector. Extensive learning has been provided to these composite features in the second stage of feature extraction, i.e., the convolutional neural network-based transfer learning. A detailed description of each module is elaborated in the following sub-sections.

3.1. Image preprocessing module

The image or a video frame received from the sensor device contains many impurities, irregularities, and noise due to various reasons such as changes in illumination conditions, camera viewpoints, low resolution, long focal distance, and more. The image preprocessing enhances these images, which includes face detection, segmentation, alignment, normalization, and image enhancement.

3.1.1. Face detection and segmentation

This task is the prerequisite for any face biometric-based application. We have deployed the most popular Voila-Jones method to implement face detection tasks, comprising Haar-like features, integral images, cascade classifier, and Adaboost algorithm. Haar-like features [11] are the grayscale templates that include line, center-surround, and edge features and are more similar to a human face's geometry. Integral images are utilized on facial pixels for faster feature extraction. Cascaded representations of all extracted features are collected from facial and non-facial regions. The Adaboost method uses weak classifiers to attain different features and then increases the strength of each classifier by combining these weak classifiers into a single robust classifier. The features that obtain higher votes are accepted and concatenated, while the rest are rejected, as

$$F(x) = \alpha_1 f(x_1) + \alpha_2 f(x_2) + \alpha_3 f(x_3) + \cdots + \alpha_n f(x_n), \tag{1}$$

$$F_t(x) = \textstyle\sum_{t=1}^{T} f_t(x). \tag{2}$$

3.1.2. Face alignment and normalization processes

Face alignment is an important task to adjust the human's face to the frontal position. The eye landmarks (i.e., twelve points) alone are sufficient to align the face to the frontal position out of sixty-eight facial landmarks. Thus, we first locate the eye landmarks using the Dlib library. To find the left and right eye's center point, we calculate the mean of all points for both the left and right eye concerning the *x*-coordinate and *y*-coordinate using the equations (3) and (4) below. We calculate the coordinate-wise displacement for the left and right eyes to find the left and right eye centroid:

$$\text{L\_E\_center} = \left\{\frac{x\ (\text{L\_E\_LM}s)}{2}, \frac{y\ (\text{L\_E\_LM}s)}{2}\right\}, \tag{3}$$

$$\text{R\_E\_center} = \left\{\frac{x\ (\text{R\_E\_LM}s)}{2}, \frac{y\ (\text{R\_E\_LM}s)}{2}\right\}. \tag{4}$$

Furthermore, we calculate the Euclidean Distance (ED) for these displacements as shown in (5), (6), and (8). The angular displacement (arctangent) formula is shown in (7). Afterward, we define the expected range of visible faces and the eyes to be scaled after alignment, as shown in (9). The eye center is now calculated, considering the left and right eye's position coordinates as shown in (10):

$$dy = \mathrm{R\_E\_Center}_{[1]} - \mathrm{L\_E\_Center}_{[1]}, \tag{5}$$

$$dx = \mathrm{R\_E\_Center}_{[0]} - \mathrm{L\_E\_Center}_{[0]}, \tag{6}$$

$$\mathrm{Angle}\ (\theta) = \tan^{-1}\left\{\frac{dy}{dx}\right\}, \tag{7}$$

$$\mathrm{ED} = \sqrt{(dx^2 + dy^2)}, \tag{8}$$

$$\mathrm{Scale}\ (S) = \frac{\mathrm{Desired\_distance}}{\mathrm{ED}}, \tag{9}$$

$$\mathrm{Eyes\_Center} = \left\{\frac{\mathrm{L\_E\_center[0]+R\_E\_center\ [0]}}{2}, \frac{\mathrm{L\_E\_center[1]+R\_E\_center\ [1]}}{2}\right\}. \tag{10}$$

The parameters detected such as center eye point, scaling factor, and rotation angle are sufficient to form a rotation matrix. This center point creates a new matrix with reduced face width and increased face height. Finally, warp affine transformation is applied with the three essential details: the image, the new matrix (consisting of translation, rotation, and scaling), and the updated shape of the expected face. Face normalization is the process of setting each image to an appropriate range.

3.1.3. Image enhancement

Image enhancement includes various preprocessing operations such as serialization and annotation, resizing, and class-wise labeling of the dataset. Applying these operations reduces the inconsistency and complexity of our use case NUAA dataset [12, 13].

3.2. Color space conversion and channel splitting module

The color space is a mathematical abstraction that allows the reconstruction of the color through various digital and analog representations. The distinct color space distribution may be essential to understanding the suitability of a specific color space to discriminate a fake face from an actual face. In this article, we consider three diverse color spaces such as RGB, HSV, and YCrCb.

3.2.1. RGB color space

RGB color distribution comprises three primary colors: Red, Green, and Blue. This color model is the most reliable and convenient for the human visual system as it can reproduce a wide range of new colors and is more intuitive for visualizing color images.

3.2.2. HSV color space

HSV [14] represents Hue, Saturation, and Value to represent specific color information. The Hue component is responsible for resembling the actual color. Saturation defines the strength of the whiteness, whereas the value represents the count of intensity (i.e., lightness). The Hue and Saturation components reflect the chrominance information, whereas the value represents the luminance information. The three components, *H*, *S*, and *V*, can be derived from the RGB color model as shown in (11), (12), and (13), respectively:

(11) $$H = \cos^{-1}\left[\frac{0.5[(R-G)+(R-B)]}{\sqrt{[(R-G)^2+(R-B)(G-B)]}}\right],$$

(12) $$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)],$$

(13) $$V = \frac{1}{3}[(R+G+B)].$$

3.2.3. YCrCb color space

YCrCb [4] is one of the prominent color spaces mainly used to represent the colors for digital TV. The scene captured through the real face and the spoofed face has significantly differed in brightness (i.e., Y value or intensity) and chrominance information (i.e., C_r and C_b). This color space provides more effective and fine complementary details of facial color through two-color channel components. The conversion of the RGB color model to YCrCb is easy in mathematical essence, as shown in (14), (15), and (16).

(14) $$Y = R \times 0.301 + G \times 0.586 + B \times 0.113\ ,$$

(15) $$C_\mathrm{b} = R \times (-0.168) + G \times (-0.332) + B \times (0.500) + 128,$$

(16) $$C_\mathrm{r} = R \times (0.500) + G \times (-0.417) + B \times (-0.082) + 128.$$

3.2.4. Channel-splitting process

Channel splitting [5, 14] segregates respective channels from a specific color space distribution to analyze the impact of each color channel in extracting significant features.

3.3. Feature extraction module

This section demonstrates a new efficient dual-phase feature extraction method to obtain more intuitive information for efficient classification. The first method is the Color-based Local Binary Pattern (CLBP), which extracts the channel-wise features from a given color space and passes the composite features to the deep network. The second method is a convolutional neural network-based transfer learning (VGG16, InceptionV3, and MobileNetV2) deployed to provide intensive learning that raises the classification accuracy remarkably.

3.3.1. Algorithm 1

Algorithm 1 illustrates the complete feature extraction phases.

Algorithm 1. Two-Stage Feature Extraction Process

```
Input:      Input image $I_{\mathrm{Face}}$ image size H×W×C.
Output:     Classification of the input image $I_{\mathrm{Face}}$
Initialize Parameters
$X\_\mathrm{train} = [\,],\ Y\_\mathrm{train} = [\,],\ X\_\mathrm{test} = [\,],\ Y\_\mathrm{test} = [\,]$
$X\_\mathrm{train},\ Y\_\mathrm{train} \leftarrow \mathrm{next\ (train\_generator)}$
$X\_\mathrm{test},\ Y\_\mathrm{test} \leftarrow \mathrm{next\ (test\_generator)}$
$X\_\mathrm{train\_LBP} = [\,], Y\_\mathrm{train\_LBP} = [\,], X\_\mathrm{test\_LBP} = [\,], Y\_\mathrm{test\_LBP} = [\,]$
Step 1. Procedure 1: Color − based LBP ($I_{\mathrm{Face}}$)
Step 2. Load input image $I_{\mathrm{Face}}$
Step 3. Convert the color Space of the image $I_{\mathrm{Face}}$
Step 4. Split the color space to extract channel − wise information
Step 5. for images in $X$_train do
Step 6.     for images in $X$_test do
Step 7.         Find the histogram for each distinct channel
Step 8.         $D \leftarrow$ LocalBinaryPattern (neighbourhood, radius, method)
Step 9.         Append the channel − wise histogram with LBP to $X$_test_LBP
Step 10.        Append the channel − wise histogram with LBP to $X$_train_LBP
Step 11.    end for
Step 12.    $X$_train_LBP ← $X$_train_LBP. astype('float32')
Step 13.    $X$_test_LBP ← $X$_test_LBP. astype('float32')
Step 14.    $X$_train_LBP ← $X$_train_LBP $*$ 1/255
Step 15.    $X$_test_LBP ← $X$_test_LBP $*$ 1/255
Step 16.    $X$_train_LBP ← asarray($X$_train_LBP)
Step 17.    $X$_test_LBP ← asarray($X$_test_LBP)
Step 18. $X$_train_LBP ← Reshape($X$_train_LBP )
Step 19. $X$_test_LBP ← Reshape($X$_test_LBP )
Step 20. $Y$_train_LBP ← Assign_label ($Y$_train)
Step 21. $Y$_test_LBP ← Assign_label ($Y$_test)
Step 22. end for
Step 23. end Procedure 1
Step 24. Procedure 2: CNN − based TL models ($I_{\text{Face}}$, $X$_train_LBP, $X$_test_LBP)
Step 25. Load the TL base model excluding its Top layer
Step 26. Shape ($X$_train_LBP [1: ]) ← Shape(input_shape($H, W, C$))
Step 27. $M$ = Flatten (Output(base model))
Step 28. Apply dense network with activation function
Step 29. Apply binary classification with sigmoid function
Step 30. Compile the model using Optimizer, loss, and metrics
Step 31. Fit the model for training and validation data
Step 32. return the face image $I_{\text{Face}}$ with class labels
end Procedure 2
```

The algorithm above highlights the complete feature extraction method, consisting of handcrafted color-texture-based LBP (Stage-1), and CNN-based transfer learning (Stage-2), with the significance of features' fusion.

3.3.2. Color-Based Local Binary Pattern (CLBP)

The local binary pattern is the most widely used grayscale-based feature descriptor. The LBP extracts low-level information such as edge, color, and intensity from each color channel. The histogram of these features represents the frequency occurrence of chrominance (color) and luminance (brightness) information. The example of the LBP operator is shown in Fig. 2.

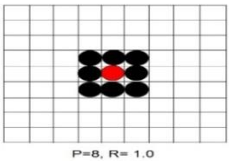


Fig. 2. Example of LBP operator

LBP evaluates a binary code for each image pixel by considering the threshold value in the circularly symmetric neighborhood with the central pixel's value, as shown in Fig. 3. Each value higher than a given threshold value (i.e., a central pixel value) is assigned to zero, whereas the rest values are assigned one.

The LBP operator finalizes the binary and their decimal equivalent by assigning one to each pixel with a higher than a threshold value and zero to the rest of the pixels, starting from the left top corner. Now, the equivalent decimal number is placed in the central position of the matrix. This new central value is the actual pixel value generated through LBP to represent the features better. Fig. 4 depicts the visualization of LBP operation for each color channel image.

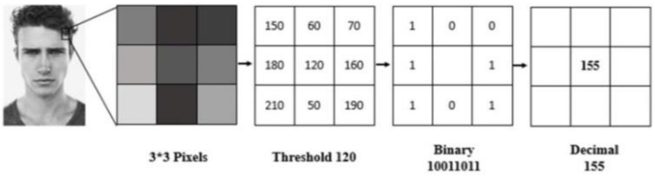
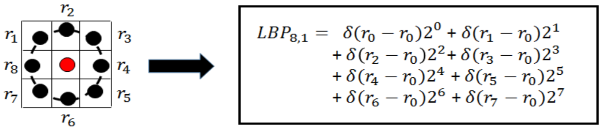


Fig. 3. Illustration of general LBP operations



$$\begin{aligned} LBP_{8,1} = {} & \delta(r_0 - r_0)2^0 + \delta(r_1 - r_0)2^1 \\ & + \delta(r_2 - r_0)2^2 + \delta(r_3 - r_0)2^3 \\ & + \delta(r_4 - r_0)2^4 + \delta(r_5 - r_0)2^5 \\ & + \delta(r_6 - r_0)2^6 + \delta(r_7 - r_0)2^7 \end{aligned}$$

Fig. 4. Visualization of LBP operation for each color channel image

LBP is intended to deal with grayscale images only. However, LBP considers the luminance and chrominance information at different locality levels. Therefore, we have modified a normal LBP to a color-based LBP consisting of eight neighborhood pixels with a radius value. The working of the LBP descriptor for the specified number of neighbors and radius of the center is represented in the equations

$$(17)\qquad \mathrm{LBP}^{i}_{P,R}(A,B) = \begin{cases} \sum_{n=0}^{P-1} \delta\left(r_n^{(i)} - r_{\mathrm{c}}^{(i)}\right) \times 2^n & \text{if } U^{(i)} \le 2, \\ P(P-1) + 2 & \text{otherwise,} \end{cases}$$

$$(18)\qquad U^{(i)} = \left|\delta\left(r_{P-1}^{(i)} - r_{\mathrm{c}}^{(i)}\right) - \delta\left(r_0^{(i)} - r_{\mathrm{c}}^{(i)}\right)\right| + \sum_{n=1}^{P}\left|\delta\left(r_n^{(i)} - r_{\mathrm{c}}^{(i)}\right) - \delta\left(r_{n-1}^{(i)} - r_{\mathrm{c}}^{(i)}\right)\right|,$$

where $\delta(A) = 1$ if $A \ge 0$, otherwise $0. r_{\mathrm{c}}$ and r_n $(n = 0, 1, 2, \dots, P-1)$ are the intensity values of the central pixel (A, B) and P is its neighborhood pixels located at the circle of radius $R(R > 0)$. Let I be a face image for color space S, where, $S \in \{\mathrm{RGB}, \mathrm{HSV}, \mathrm{YCrCb}\}$ and, let $H_S^{(i)}$, $\{i = 1, \dots, M\}$ be its uniform LBP histogram extracted from the M channel of the space S. The color LBP feature of the image I represented in the space S can be defined as

$$(19)\qquad H_S = \left[H_S^{(1)} \dots \dots H_S^{(M)}\right].$$

The conceptual diagram of the proposed color-based LBP feature extraction process is shown in Fig. 5.

Fig. 5 clarifies the steps involved in image preprocessing, color space conversion, channel splitting, and the first-stage feature extraction process. Here, channel-wise extracted features for each color space are combined to obtain a composite feature vector. However, the color-based LBP feature descriptor was insufficient to provide extensive learning to the model. It is seen that channel-wise LBP features have been extracted, and a normalized composite feature vector is passed to the deep network for a different level of feature extraction. Thus, we need a deep neural network model to deal with the image data effectively.

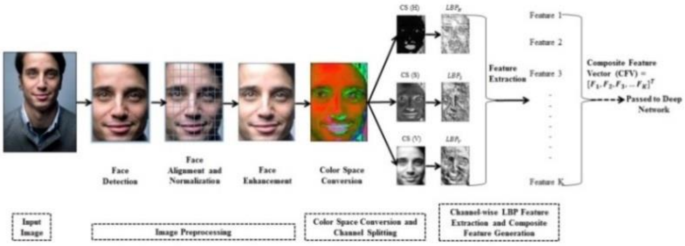

CS (H)
LBP_H
CS (S)
LBP_S
CS (V)
LBP_V
Feature 1
Feature 2
Feature 3
Feature K
Feature Extraction
Composite Feature Vector (CFV) = $[F_1, F_2, F_3, \dots F_K]^T$
Passed to Deep Network
Face Detection
Face Alignment and Normalization
Face Enhancement
Color Space Conversion
Input Image
Image Preprocessing
Color Space Conversion and Channel Splitting
Channel-wise LBP Feature Extraction and Composite Feature Generation

Fig. 5. Conceptual diagram of the proposed color-based LBP feature extraction process

3.3.3. Convolutional Neural Network-based Transfer Learning method (CNNTL)

The convolutional neural network is a widely used deep network, particularly for image classification problems, as it efficiently extracts spatial features from input streams [15]. Transfer learning is one of the most popular methods inspired by the convolutional neural network and is used in many applications, including face recognition. As the name suggests, transfer learning utilizes the learning experience to reduce the efforts to train massive networks and the overhead of computing the weights for a whole network from scratch [16]. Intelligence (i.e., pretrained weights) and hyperparameters fine-tuning are effectively utilized to extract the deep features that can better represent real-world scenarios. The transfer learning uses the ImageNet pretrained weights. ImageNet includes 1000 distinct classes of image data. A conceptual diagram of the proposed transfer learning-based feature extraction process is shown in Fig. 6.

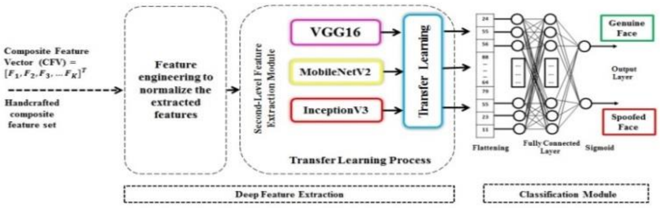

Composite Feature Vector (CFV) = $[F_1, F_2, F_3, \dots F_K]^T$
Handcrafted composite feature set
Feature engineering to normalize the extracted features
Second-Level Feature Extraction Module
VGG16
MobileNetV2
InceptionV3
Transfer Learning
Transfer Learning Process
Flattening
Fully Connected Layer
Sigmoid
Output Layer
Genuine Face
Spoofed Face
Deep Feature Extraction
Classification Module

Fig. 6. Conceptual diagram of the proposed transfer learning-based feature extraction process

The three prominent transfer learning models, VGG16, InceptionV3, and MobileNetV2 are deployed on the normalized LBP composite features, leaving the top classification layer frozen. Thus, the top layer restricts the model from presenting the output. The output of the base model is considered with the new customized dense networks and the classification layer. We also have added three dense customized networks to the VGG16 model (i.e., fully connected, 512, 256) feature maps. In the InceptionV3 model, we consider four customized dense layers (i.e., FC, 1024, 512, 256) feature maps. In contrast, the MobileNetV2 contains three denser layers (i.e., FC, 512,128). A new classification layer is added to each model at the end of the network structure. The specification of the transfer learning models (VGG16, InceptionNetV3, and MobileNetV2) is represented in Table 2.

Table 2. Detailing of VGG16, MobileNetV2, InceptionV3

Model	Developed by	Input	Specification	No of parameters	Accuracy/ error	Refe-rences
VGG16 (2014)	Simonyan and Zisserman	224 ×224	Layers-16, CL-13, FCL-03	138 M	7.3 % Top-5, Error	[18]
MobileNetV2 (2018)	Daniel Falbel, JJ Allaire, François	224 ×224	Residual bottleneck layers-19	2.11 M	72 %, Top-1 Accuracy	[17]
InceptionV3 (2014)	Szegedy	299 ×299	Layers-48	7.0 M	6.67 % Top-5, Error	[19]

3.3.4. Visual Geometry Group (VGG16) network architecture

VGG16 is a CNN architecture-based model mostly preferred for computer vision problems, especially image-based face classification. However, the architecture of VGG16 is slightly complex, containing thirteen convolutional layers and three fully-connected layers with many parameters (138 million).

3.3.5. MobileNetV2 network architecture

MobileNetV2 is the second extended version of MobileNet after MobileNetV1. The MobileNetV2 is a CNN-based model that provides convenience to deal with low-power computational devices, such as mobile and Raspberry Pi, for real-time application. Therefore, MobileNetV2 has a small three-layer architecture consisting of 2.11 million parameters.

3.3.6. InceptionV3 network architecture

The InceptionV3 is an improved version of InceptionV1. This model has forty-eight layers of deep networks; however, it breaks large convolution into a smaller grid in conjunction with multiple-size filters. This feature makes the model more efficient and intuitive than other contemporaries, especially in image analysis.

The model’s description, including the features maps and other hyperparameters used in this work, is presented in Table 3.

Table 3. Specification of the tuned parameters for proposed models

Fine Tuned Parameters		VGG16	InceptionV3	MobileNetV2
Input image		224 × 224	299 × 299	224 ×224
CL		13 blocks (total 16)	45 blocks (total 48)	Bottleneck layers-19
FC, Dense, and Classification		04 (F, 512, 256, 1)	05 (F, 1024, 512, 256, 1)	04 (F, 512, 128, 1)
Learning Rate		0.0001	0.0001	0.0001
Kernel Size		3 × 3	1 × 1, 3 × 3, 5 × 5, 7 × 7	1 × 1, 3 × 3, 7 × 7
Batch Size		32	32	32
Pooling		Max	Average	Global average
Optimization		Adam	Adam	Adam
Dropout		0.25 (dense)	0.5 (dense)	0.25 (dense)
Number of epochs		30	30	30
Activation	Conv	ReLu	ReLu	ReLu
	FC	ReLu	ReLu	ReLu
	Class	Sigmoid	Sigmoid	Sigmoid
Callbacks		ReducedLR, EarlyStopping	ReducedLR, EarlyStopping	ReducedLR, EarlyStopping

We have performed fine-tuning of hyperparameters after unfreezing the top layers of the base model to train the whole network for our data stream. In order to obtain the best classification accuracy, various hyper-parameters, such as learning rate, activation function, the total count of epochs, batch size, pooling, kernel size, optimizer, early stopping, and dropout were precisely fine-tuned after several experiments and have been finalized accordingly. The final layer of the deep network is the classification layer, where the sigmoid activation function predicts the class of the given image.

4. Experiments and result analysis

This section describes the requirement for the experimental setup. The experiments have been performed on an interactive python notebook (i.e., Google Colaboratory). Google Colab is an open-source cloud-based platform with free GPU and TPU support irrespective of the system's configurations. The "Tesla K80" GPU device have been accessed through the CUDA version 11.2 for fast preprocessing on image matrices during training. Python 3.7.10 version with Tensorflow 2.4.1 and Keras 2.4.3 have been considered to perform all the experiments in terms of programming. Subsequently, the outcomes of these experiments have been evaluated and analyzed comparatively for all of our proposed models.

4.1. Dataset collection and possible scenarios

The performance of deep learning models depends on the machine's data (i.e., dataset) and level of data understanding (i.e., learning). Thus, we have considered a benchmark dataset, i.e., NUAA, for our experiments. This dataset contains twenty-five videos captured via a webcam with an image size of 640 by 480 for real faces and thirty-three videos with the exact resolution for the spoofed face. These captured video streams consist of fifteen subjects, and each subject is of the Asian race. Eighty percent of the subjects are men, and the remaining twenty percent are women, with both subjects being 20 to 30 years of age. Images/frames can be extracted from the video files. The training and test split of these two class samples are shown in Table 4.

Table 4. Specification of the NUAA dataset

NUAA Dataset			
Class	Distribution	Number of samples	Total
Genuine face	Train	4080	5100
	Test	1020	
Spoofed face	Train	6000	7500
	Test	1500	
Total number of samples in the NUAA dataset			**12,600**

Some sample images from the NUAA dataset consisting of live face, handhold print, digital photo, and blurred and over-exposed images are depicted in Fig. 7.

The dataset consists of multiple images of genuine and fake faces taken at different environmental conditions with different time instances for acquisition purposes, reflecting more generalized real-world scenarios. Next section illustrates the performance measures evaluated in our proposed approach.

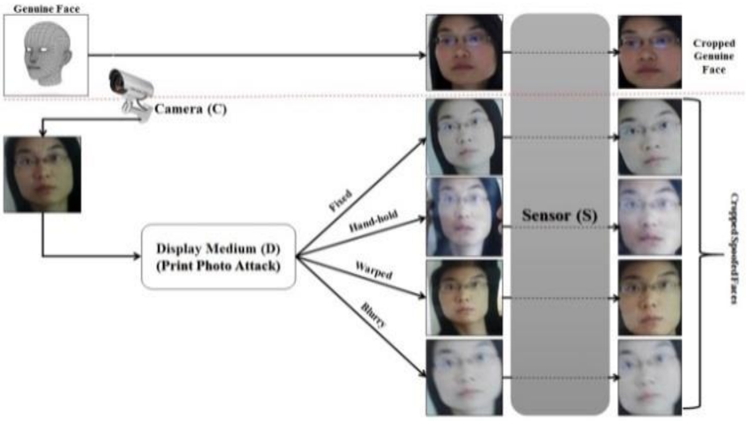


Fig. 7. The possible scenarios covered in the proposed methodology

4.2. Performance measures

To measure the effectiveness of the models being proposed, the test samples should predict the correct class for the given input image or video streams. The correct predicted labels out of total test samples represent the accuracy of the model. In comparison, the absolute difference between the predicted class labels and the expected class labels shows the loss of the model. The vital parameters used to evaluate performance matrices can be classified into the following categories:

1. True positive. The number of test samples classified as True and predicted as True.

2. False positive. The number of test samples classified as False and predicted as True.

3. True negative. The number of test samples classified as False and predicted as False.

4. False negative. The number of test samples classified as True and predicted as False.

The classification measures, such as accuracy, precision, recall, F1-Score, and negative predicted value, can be calculated as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{(\text{TP+TN+FP+FN})}, \tag{20}$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})}, \tag{21}$$

$$\text{Recall (Sensitivity)} = \frac{\text{TP}}{(\text{TP+FN})}, \tag{22}$$

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN+FP})}, \tag{23}$$

$$\text{Negative Predicted Value} = \frac{\text{TN}}{(\text{TN+FN})}, \tag{24}$$

$$\text{F1-Score} = \frac{2\times\text{Precision}\times\text{Recall}}{(\text{Precision+Recall})}. \tag{25}$$

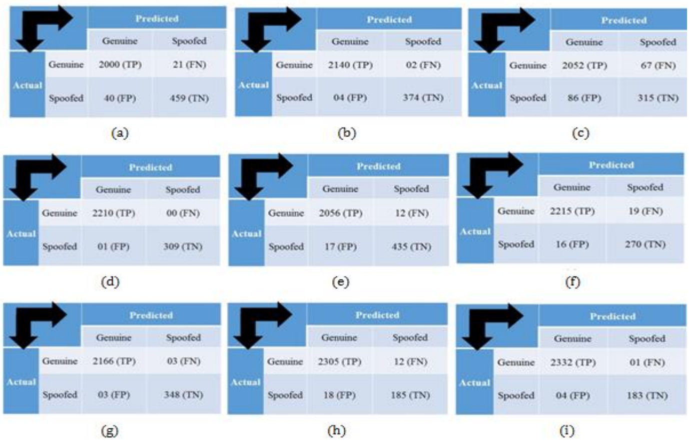
4.3. Experimental outcome

We have performed the feature extraction and classification experimentations for the three diverse color spaces (i.e., RGB, HSV, and YCrCb) for three different models (i.e., VGG16, InceptionV3, and MobileNetV2) after first level feature extraction through color-based LBP. Here, we calculate classification measures to validate the performance of the proposed model. Fig. 8 (a)-(i) represents the confusion matrix for all three transfer learning models concerning the three color spaces.

Based on the parameters used in the confusion matrix, we calculate the other performance metrics as per the formula from (20) to (25). Table 5 depicts the performance metrics evaluated for the NUAA dataset on proposed models.

Table 6 demonstrates the final experimental results for all the proposed models considering three color spaces (i.e., RGB, HSV, and YcrCb).

The experiment results reveal that the RGB color space performs extremely best with an accuracy of 99.76% for the color-based LBP and InceptionV3 model, while the HSV color space provides excellent results with an accuracy of 99.96% for the color-based LBP and VGG16 model. The YCrCb space provides an accurate result of 99.80% accuracy for the color-based LBP and MobileNetV2 methods. Thus, The HSV color space represents higher accuracy than the other two proposed models.



		Predicted	
		Genuine	Spoofed
Actual	Genuine	2000 (TP)	21 (FN)
	Spoofed	40 (FP)	459 (TN)

(a)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2140 (TP)	02 (FN)
	Spoofed	04 (FP)	374 (TN)

(b)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2052 (TP)	67 (FN)
	Spoofed	86 (FP)	315 (TN)

(c)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2210 (TP)	00 (FN)
	Spoofed	01 (FP)	309 (TN)

(d)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2056 (TP)	12 (FN)
	Spoofed	17 (FP)	435 (TN)

(e)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2215 (TP)	19 (FN)
	Spoofed	16 (FP)	270 (TN)

(f)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2166 (TP)	03 (FN)
	Spoofed	03 (FP)	348 (TN)

(g)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2305 (TP)	12 (FN)
	Spoofed	18 (FP)	185 (TN)

(h)

		Predicted	
		Genuine	Spoofed
Actual	Genuine	2332 (TP)	01 (FN)
	Spoofed	04 (FP)	183 (TN)

(i)

Fig. 8. The confusion matrix for each color model (a)-(i)

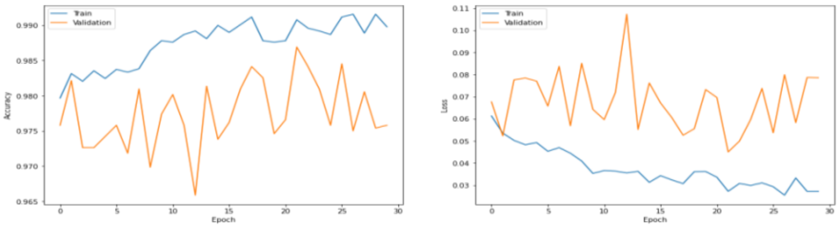
Table 5. Performance measures evaluated for the proposed model

Proposed Model	Feature extraction methods		Performance measures					
			Accuracy	Precision	Recall	F1-Score	Specificity	Negative predicted value
Proposed Model-I	LBP$_{(RGB)}$	VGG16	97.58	98.03	98.96	98.50	91.98	95.63
	LBP$_{(RGB)}$	InceptionV3	**99.76**	99.81	99.90	99.86	98.94	99.47
	LBP$_{(RGB)}$	MobileNetV2	93.93	95.98	96.84	96.40	78.55	82.46
Proposed Model-II	LBP$_{(HSV)}$	VGG16	**99.96**	99.95	100	99.98	99.68	100
	LBP$_{(HSV)}$	InceptionV3	98.85	99.18	99.42	99.30	96.23	97.32
	LBP$_{(HSV)}$	MobileNetV2	98.61	99.15	99.28	99.22	94.41	93.42
Proposed Model-III	LBP$_{(YCrCb)}$	VGG16	99.76	99.86	99.86	99.86	99.15	99.15
	LBP$_{(YCrCb)}$	InceptionV3	98.81	99.23	99.49	99.35	91.13	93.90
	LBP$_{(YCrCb)}$	MobileNetV2	**99.80**	99.83	99.96	99.89	97.86	99.46

Table 6. Experimental outcome for all proposed models

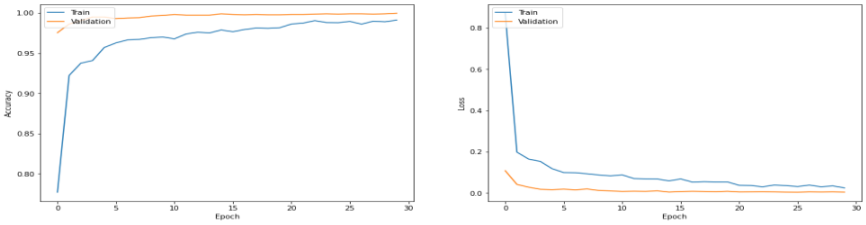
Proposed Model	Methods		Train		Validation	
			Accuracy	Loss	Accuracy	Loss
Proposed Model-I	LBP$_{(RGB)}$	VGG16	99.98 %	0.0272	97.58 %	0.0785
	LBP$_{(RGB)}$	InceptionV3	99.03 %	0.0272	**99.76 %**	0.0044
	LBP$_{(RGB)}$	MobileNetV2	99.72 %	0.0090	93.93 %	0.1844
Proposed Model-II	LBP$_{(HSV)}$	VGG16	99.67 %	0.0120	**99.96 %**	0.0015
	LBP$_{(HSV)}$	InceptionV3	98.39 %	0.0473	98.85 %	0.0342
	LBP$_{(HSV)}$	MobileNetV2	99.34 %	0.0188	98.61 %	0.0446
Proposed Model-III	LBP$_{(YCrCb)}$	VGG16	98.45 %	0.0444	99.76 %	0.0059
	LBP$_{(YCrCb)}$	InceptionV3	98.98 %	0.0319	98.81 %	0.0284
	LBP$_{(YCrCb)}$	MobileNetV2	99.61 %	0.0138	**99.80 %**	0.0064

In contrast, the overall least accuracy is found in RGB color space, with an accuracy of 93.93% for color-based LBP and MobileNetV2 methods. The trade-off between training and validation accuracy and respective loss for training and validation are represented in Figs 9-11 (Proposed Model-I for RGB color space), Figs 12-14 (Proposed Model-II for HSV color space), and Figs 15-17 (Proposed Model-III for YCrCb color space), respectively.



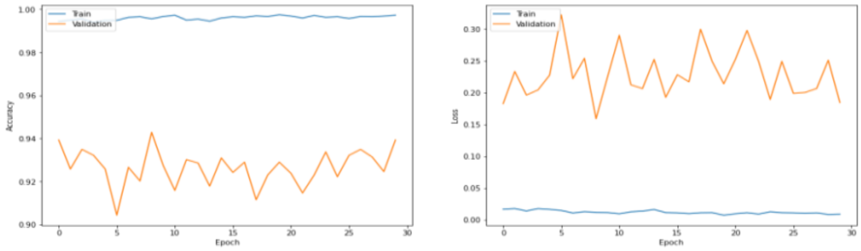
(a) (b)

Fig. 9. Trade-off between training vs. validation for VGG16 (RGB): accuracy (a); loss (b)



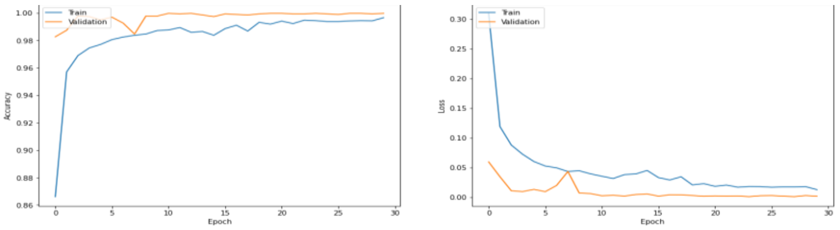
(a) (b)

Fig. 10. Trade-off between training vs. validation for InceptionV3 (RGB): accuracy (a); loss (b)



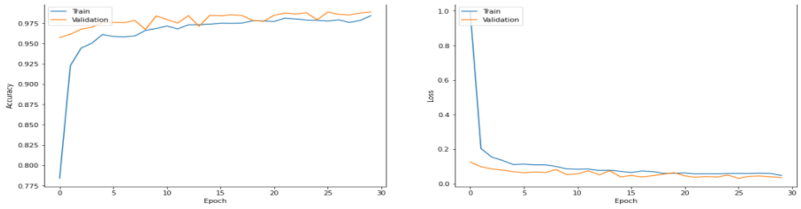
(a) (b)

Fig. 11. Trade-off between training vs. validation for MobileNetV2 (RGB): accuracy (a); loss (b)



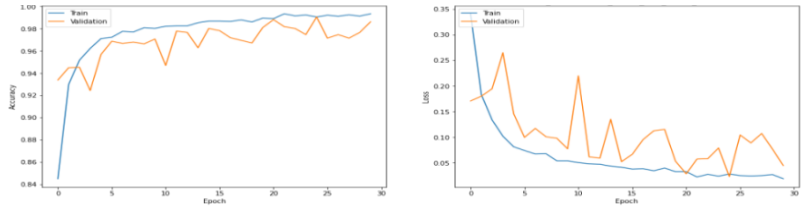
(a) (b)

Fig. 12. Trade-off between training vs. validation for VGG16 (HSV): accuracy (a); loss (b)



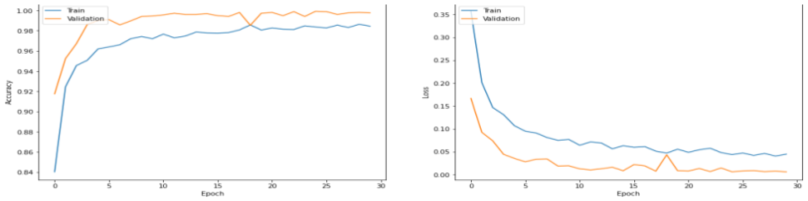
(a) (b)

Fig. 13. Trade-off between training vs. validation for InceptionV3 (HSV): accuracy (a); loss (b)



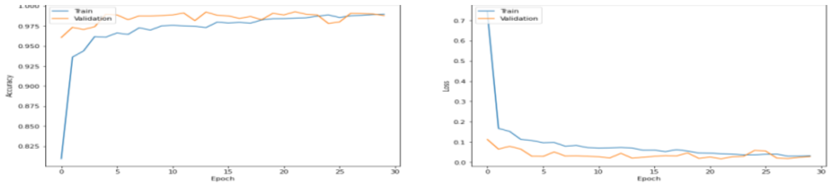
(a) (b)

Fig. 14. Trade-off between training vs. validation for MobileNetV2 (HSV): accuracy (a); loss (b)



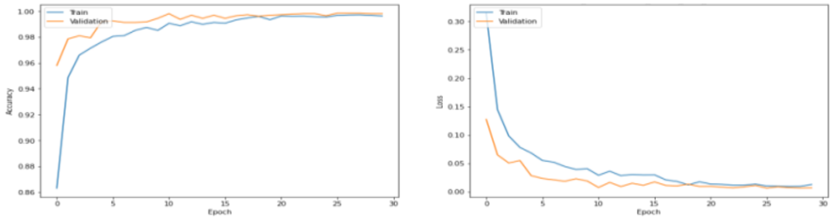
(a) (b)

Fig. 15. Trade-off between training vs. validation for VGG16 (YCrCb): accuracy (a); loss (b)



(a) (b)

Fig. 16. Trade-off between training vs. validation for InceptionV3 (YCrCb): accuracy (a); loss (b)



(a) (b)

Fig. 17. Trade-off between training vs. validation for MobileNetV2 (YCrCb): accuracy (a); loss (b)

These results clarify that HSV color space provides more promising results to classify the spoofed face and the real face, considering the chrominance information (i.e., color). In comparison, the performance of the YCrCb color space is in the second top position to classify the real face and the spoofed face. The RGB color space represents comparatively least score despite the InceptionV3 model.

4.3. Comparison with other State-of-the-Art methods

The results have been compared with other state-of-the-art methods to check the effectiveness of our outcomes from the proposed models. Table 7 represents the year-wise performance comparison between techniques based on accuracy, baseline architecture, and face spoofing attack scenarios. All comparisons have been analyzed individually, and we have only considered the results of print photo attacks for the NUAA dataset. The comparative analysis shows that all our proposed models outperform other existing approaches.

Table 7. Comparison of the proposed models with state-of-the-art methods

Reference	Baseline architecture/	Accuracy	Year
Tan, Li, Liu et al. [12]	DoG, Logistic regression	88.15%	2010
Yang, Lei, Liao et al. [20]	Component-dependent face coding, Fisher criterion	97.78%	2013
De Souza, da Silva Santos, Pires et al. [21]	LBPNet (LBP + CNN)	97.60%	2017
Anand, and Vishwakarma [8]	LBP (Mono) CNN (Mono) Min Fusion Max Fusion	94.31% 98.15% 52.9 % 52.5 %	2020
Raghavendra and Kunte [22]	EDDTCP with SVM EDDTCP with k-NN EDDTCP with LDA	93.04%, 89.83%, 92.22%	2020
S. Kumar, S. Singh and J. Kumar [23]	SegNet (CNN-based)	97%	2021
Proposed Model-I	LBP$_{(RGB)}$ + VGG16	97.58 %	2021
	LBP$_{(RGB)}$ + InceptionV3	**99.96 %**	2021
	LBP$_{(RGB)}$ + MobileNetV2	93.93 %	2021
Proposed Model-II	LBP$_{(HSV)}$ + VGG16	**99.96 %**	2021
	LBP$_{(HSV)}$ + InceptionV3	98.85 %	2021
	LBP$_{(HSV)}$ + MobileNetV2	98.61 %	2021
Proposed Model-III	LBP$_{(YCrCb)}$ + VGG16	99.76 %	2021
	LBP$_{(YCrCb)}$ + InceptionV3	98.81 %	2021
	LBP$_{(YCrCb)}$ + MobileNetV2	**99.80 %**	2021

Interestingly, a significant performance difference is found among the handcrafted feature-based methods, deep neural network methods, and our integrated methods, which involve local (handcrafted) and global (deep) features for the face spoofing task. Table 7 clearly shows that the performance of our InceptionV3 model with RGB color feature-based LBP is superior to other techniques. The VGG16 method with LBP and HSV color space outperforms other methods by a margin. In the same way, the accuracy of the mobileNetV2 with the YCrCb method leads by a significant margin to other methods. Thus, it is clear from the above results that the combination of transfer learning methods with the color-based LBP can significantly improve the accuracy of the face spoofing detection task.

5. Conclusion

This article investigates the effectiveness of different color spaces (RGB, HSV, and YCrCb) to discriminate the spoofed face from the genuine face by implementing image preprocessing and the dual-phase feature extraction (i.e., handcrafted and deep) methods. The color-based LBP (first-stage) provides more intuitive results for analyzing the intrinsic disparities of color space on different locality levels. The spoofed face is the recaptured image of the original image; thus, it involves color distortion, which the LBP has promisingly has detected. This paper applies the channel-splitting process to segregate distinct color channels on which LBP is deployed to get adequate patterns. All these extracted features have been concatenated to create a new composite feature vector. However, LBP alone is insufficient to provide deep learning to our model. Thus, we have passed these composite feature vectors to the next level (i.e., transfer learning module) after normalizing the feature set. Intelligence (i.e., pretrained weights) and hyperparameters fine-tuning are effectively utilized to extract the deep features that can better represent real-world scenarios. We have implemented our customized dense network and new classification layer for all three models (i.e., VGG16, InceptionV3, and MobileNetV2). We have achieved the best accuracy of 99.96% for HSV color-based LBP with VGG16, while 99.80% for YCrCb color-based LBP with MobileNetV2, and the accuracy of 99.76% for RGB with the InceptionV3 method. In the future, we will attempt to produce a new real-time model capable of recognizing faces for the 3D mask in conjunction with other presentation artifacts interfaces such as replay video and testing on cross-datasets.

References

1. Moon, Y., I. Ryoo., S. Kim. Face Antispoofing Method Using Color Texture Segmentation on FPGA. – Security and Communication Networks, Vol. **2021**, 2021, sp. 9939232.
2. Zhang, L. B., F. Peng, L. Qin, M. Long. Face Spoofing Detection Based on Color Texture Markov Feature and Support Vector Machine Recursive Feature Elimination. – Journal of Visual Communication and Image Representation, Vol. **51**, 2018, pp. 56-69.
3. Junqin, H., J. Luo. Face Spoofing Detection Based on Combining Different Color Space Models. – In: Proc. of IEEE 4th International Conference on Image, Vision and Computing (ICIVC'19), IEEE, 2019, pp. 523-528.
4. Boulkenafet, Z., J. Komulalnen, A. Hadid. Face Spoofing Detection Using Colour Texture Analysis. – IEEE Transactions on Information Forensics and Security, Vol. **11**, 2016, No 8, pp. 1818-1830.

5. Boulkenafet, Z., J. Komulalnen, A. Hadid. Face Anti-Spoofing Based on Color Texture Analysis. – In: Proc. of IEEE International Conference on Image Processing (ICIP'15), IEEE, 2015, pp. 2636-2640.
6. Venkatesh, B., J. Anuradha. A Review of Feature Selection and Its Methods. – Cybernetics and Information Technologies, Vol. **19**, 2019, No 1, pp. 3-26.
7. Thomas, S. K., A. Mathew. A Noval Approach for Face Spoof Detection Using Color-Texture, Distortion and Quality Parameters. – International Journal on Recent and Innovation Trends in Computing and Communication, Vol. **5**, 2017, No 2, pp. 218-220.
8. Anand, A., D. K. Vishwakarma. Face Anti-Spoofing by Spatial Fusion of Colour Texture Features and Deep Features. – In: Proc. of 3rd International Conference on Intelligent Sustainable Systems (ICISS'20), IEEE, 2020, pp. 1012-1017.
9. Chen, F. M., C. Wen, K. Xie, F. Q. Wen, G. Q. Sheng, X. G. Tang. Face Liveness Detection: Fusing Colour Texture Feature and Deep Feature. – IET Biometrics, Vol. **8**, 2019, No 6, pp. 369-377.
10. Edmunds, T., C. Alice. Face Spoofing Detection Based on Colour Distortions. – IET Biometrics, Vol. **7**, 2018, No 1, pp. 27-38.
11. Rusia, M. K., D. K. Singh, M. A. Ansari. Human Face Identification Using LBP and Haar-Like Features for Real Time Attendance Monitoring. – In: Proc. of 5th International Conference on Image Information Processing (ICIIP'19), IEEE, 2019, pp. 612-616.
12. Tan, X., Y. Li, J. Liu, L. Jiang. Face Liveness Detection from a Single Image with Sparse Low Rank Bilinear Discriminative Model. – In: Proc. of European Conference on Computer Vision, Berlin, Heideberg, Springer, 2010, pp. 504-517.
13. Rusia, M. K., D. K. Singh. A Comprehensive Survey on Techniques to Handle Face Identity Threats: Challenges and Opportunities. – Multimed. Tools Appl., 2022, pp. 1-80.
14. Abdullakutty, F., P. Johnston, E. Elyan. Fusion Methods for Face Presentation Attack Detection. – Sensors, Vol. **22**, 2022, No 14, p. 5196.
15. Ansari, M. A., D. K. Singh. ESAR, An Expert Shoplifting Activity Recognition System. – Cybernetics and Information Technologies, Vol. **22**, 2022, No 1, pp. 190-200.
16. Sulaiman, V., N. Ravikumar, A. Davari, S. Ellmann, A. Maier. Classification of Breast Cancer Histology Images Using Transfer Learning. – In: Proc. of International Conference Image Analysis and Recognition, Springer Cham, 2018, pp. 812-819.
17. Xiang, Q., X. Wang, R. Li, G. Zhang, J. Lai, Q. Hu. Fruit Image Classification Based on Mobilenetv2 with Transfer Learning Technique. – In: Proc. of 3rd International Conference on Computer Science and Application Engineering, 2019, pp. 1-7.
18. Han, D., Q. Liu, W. Fan. A New Image Classification Method Using CNN Transfer Learning and Web Data Augmentation. – Expert Systems with Applications, Vol. **95**, 2018, pp. 43-56.
19. Xia, X., C. Xu, B. Nan. Inception-v3 for Flower Classification. – In: Proc. of 2nd International Conference on Image, Vision and Computing (ICIVC'17), IEEE, 2017, pp. 783-787.
20. Yang, J., Z. Lei, S. Liao, S. Z. Li. Face Liveness Detection with Component Dependent Descriptor. – In: Proc. of Int. Conference on Biometrics (ICB'13), 2013, pp. 1-6.
21. De Souza, G. B., D. F. da Silva Santos, R. G. Pires, A. N. Marana, J. P. Papa. Deep Texture Features for Robust Face Spoofing Detection. – IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. **64**, 2017, No 12, pp. 1397-1401.
22. Raghavendra, R., R. S. Kunte. A Novel Feature Descriptor for Face Anti-Spoofing Using Texture Based Method. – Cybernetics and Information Technologies, Vol. **20**, 2020, No 3, pp. 159-176.
23. Kumar, S., S. Singh, J. Kumar. Face Spoofing Detection Using Improved SegNet Architecture with a Blur Estimation Technique. – International Journal of Biometrics, Vol. **13**, 2021, No 2-3, pp. 131-149.

Received: 08.02.2022; Second Version: 12.07.2022; Accepted: 22.07.2022