IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 6, June, 2026, pp. 1711-1719
Journal homepage: http://iieta.org/journals/jesa

Hybrid Face Spoof Detection Using Xception, Histogram of Oriented Gradients and Color Histograms with AdaBoost Classifier

Durga Pavani Andavolu[1*], Harsha Shastri Vaddepaty[2]

[1] Department of Computer Science and Engineering, Aurora Higher Education and Research Academy (Deemed to be University), Hyderabad 500092, India
[2] Department of Computer Applications, Aurora Higher Education and Research Academy (Deemed to be University), Hyderabad 500092, India

Corresponding Author Email: aud22egcse04@aurora.edu.in

https://doi.org/10.18280/jesa.590617

Received: 25 March 2026
Revised: 19 May 2026
Accepted: 9 June 2026
Available online: 30 June 2026

Keywords:
AdaBoost, anti-spoofing, color histograms, face spoof detection, feature fusion, histogram of oriented gradients, Xception

ABSTRACT

Face recognition systems have been greatly utilized in authentication, surveillance and access control, though they are susceptible to presentation attacks, such as printed photographs, replayed video content, and 3D masks, thus presenting significant security risks. Existing deep-learning-based anti-spoofing algorithms have strong feature representations, but they have low generalization to changes in illumination, sensor technology, and mode of attack. On the other hand, handcrafted feature sets are resistant to shallow spoof cues but tend to be semantically shallow. To eliminate these shortcomings, the given paper presents a hybrid face-spoof-detection system that complements deep features obtained through trained Xception networks with handcrafted histogram of oriented gradients (HOG) descriptors and YCrCb color histograms. Such complementary descriptors together capture semantic, textural, and chromatic artefacts, which are attributes of spoofing attempts. AdaBoost ensemble is then trained to exploit the heterogeneous space of features in the best way. Performing an assessment on benchmark data sets proves the superiority of robustness and accuracy, which, in turn, supports the observation of the effectiveness of hybrid feature fusion in face-spoof detection.

1. INTRODUCTION

The use of face recognition technology in modern biometric authentication systems has become a fundamental aspect of smartphone applications, financial systems, border security, and public surveillance. Face recognition is extremely vulnerable to presentation attacks, even with its widespread use, and attackers use printed photographs, phone-screen displays, or 3D masks to trick the system. These threats make use of reliable Face Anti-Spoofing (FAS) systems to achieve reliable and trustworthy authentication.

Initial studies on the topic of FAS have been based mainly on manual texture descriptors, such as Local Binary Patterns (LBP), histogram of oriented gradients (HOG), and chrominance-based descriptions. These descriptors were good enough to describe distortions in texture and surface irregularities found in spoof media.

Though computationally efficient, these techniques often collapse due to cross-domain failures caused by changes in light, camera sensors, and attack material.

The introduction of deep convolutional neural networks (CNNs) was an important breakthrough in the field of spoof detection. Discriminative spatial cues are learned in CNN-based methods, which result in better recognition of the spoof artifact moiré patterns, printing noise, and reflectance distortions. However, deep models still potentially struggle with generalization, especially when trained on small or diverse data.

Recent studies highlight the benefits of hybrid approaches that combine deep features with classical handcrafted descriptors. Handcrafted attributes are still useful in the capture of micro-textures and color-based spoof information, whereas deep models offer high-level semantic encoding. Moreover, ensemble learning algorithms like AdaBoost have been found to work on fused feature space learning with low correlation between the constituent features, particularly where there are varied statistical characteristics of the individual features.

Based on these findings, the current study plans to design a hybrid face spoof detection system that is a combination of Xception deep features with HOG and YCrCb color histogram features. This combination makes the system stronger against variations in illumination, sensor variation, and different types of attacks. AdaBoost classifier also enhances the formation of decision boundaries, which makes the method appropriate to heterogeneous feature distributions. It is hoped that the suggested framework can enhance the generalization ability and accuracy compared to the standalone deep or handcrafted models.

2. RELATED WORK

Over the decade, FAS has reached a new level of development, as handcrafted texture analysis has been replaced by deep learning models and architectures, transformer models, and multimodal fusing approaches. Classical methods and techniques used in the early times relied on LBP and Haar-based detectors to measure low-level spoofing artifacts on the image surfaces. Indicatively, Kulkarni et al. [1] showed that the Haar Cascade classifier, when combined with LBP, was very effective in enhancing the performance of spoof detection in controlled ATM systems. Classical texture descriptors, such as HOG [2] and partially CNN-based feature extraction models [3], were also found to be useful in detecting print-and-replay attacks. The development of deep learning was a breakthrough in detecting spoof. CNN-based architectures, including Xception [1], DeepPixBiS [4], and Deep Texture Networks [5], produced strong discriminative representations that were able to expose recapture artifacts. In addition, the use of additional hand-made color signals, such as Retinex-LBP cascades that are used by Chen et al. [6], led to increased resilience to different illumination scenarios. Both the research work by Zhang et al. [7] and the survey by Yang et al. [8] highlighted the benefits of deep learning but reported difficulties in generalization and bias of the datasets.

The development of learning paradigms also increased the capabilities of FAS systems. Domain adaptation and size-adaptive learning techniques [9], meta-learning-enhanced spoof detection techniques [10], and wide and deep models [11] all showed better generalization to new attack vectors. The transformer-specific algorithm contrastive self-supervised learning [12] produced better-quality representations without the use of large annotated datasets. One of the most popular classes of models that have recently appeared is Vision Transformers (ViTs). Recent transformer-based approaches such as FM-ViT [13], contrastive self-supervised transformers [14], transformer-based self-supervised learning [15], and DINO self-supervised ViTs [16] have demonstrated significant performance improvements through enhanced representation learning and long-range dependency modeling. have demonstrated large performance improvements due to their ability to learn long-range spatial relationships. Feng et al. [17] utilized intermediate ViT representations to construct richer and more discriminative spoof-related features.

Diffusion-based models have been popular as well, especially in the process of rebuilding authentic facial priors. The methods of de-spoofing diffusion that have been proposed by Zhang et al. [18]are effective at modeling the patterns of spoof noise. Likewise, hybrid attention networks [19] and multimodal fusion -based systems [7, 20] have also enhanced the feature representation and discriminating capability. The datasets are important in the development of FAS research. Public benchmark datasets commonly used in FAS research include Chinese Academy of Sciences Institute of Automation–Face Anti-Spoofing Database (CASIA-FASD) [7], OULU-NPU [20], Spoof in the Wild (SiW) [19], CelebA-Spoof [21], and these datasets provide multiple attack models and settings. More recently, large-scale multimodal datasets, including CASIA-SURF [21], are concerned with cross-modality reliability and allow comparing the performance of a model on heterogeneous data samples. Based on this discovery, the current paper presents a hybrid version, which combines deep semantic features obtained by using Xception with HOG and chromatic features (YCrCb histograms). This combination helps to overcome limitations of focusing on deep and shallow features only and facilitates generalization to datasets and types of attacks. AdaBoost is used as the classifier due to its sensitivity to non-homogeneous distributions of features and due to its proven effectiveness in the case of ensemble learning.

3. METHODOLOGY

The suggested strategy incorporates deep learning and traditional feature extraction algorithms to form a hybrid face spoof detection system. The key components include.

3.1 Preprocessing

The preprocessing stage transforms all input facial images into a standardized and model-compatible format prior to feature extraction. Given that the system utilizes a deep CNN (Xception model), handcrafted texture (HOG) features, and color distribution features (YCrCb histograms). The preprocessing pipeline should also meet three requirements that run in parallel:

- CNN input constraints
- Computation of grayscale texture.
- Statistical analysis of color space.

The input is a raw RGB face image loaded with OpenCV application cv2.imread(imgpath), as Xception strictly accepts image dimensions as 299 × 299 × 3, and scale variance is removed with each image being resized with the bilinear interpolation operation cv2.resize(Iraw,(299,299)). This ensures a consistent size of the spatial dimension of the image; scale variance of the image is eliminated and all the images are resized with the method cv2.resize(Iraw,(299,299)). The image is converted into a NumPy tensor, which is floating-point by using imgtoarray (I_{resize}). CNNs assume a batch dimension even when only a single image is being inferred, and by using the function np.expand.dims(Ifloat,axis = 0). The final Tensor is of shape (1, 299, 299, 3). Face localization was performed using the OpenCV Haar Cascade detector. The detected facial region of interest (ROI) was cropped and resized to 299 × 299 pixels to satisfy the input requirements of the Xception network. Since subsequent feature extraction was performed only on the detected facial ROI, most surrounding background information was removed prior to processing. No explicit face-alignment procedure was applied in the current implementation. The experiments were conducted using image samples from benchmark FAS datasets rather than video sequences; therefore, video frame-selection strategies were not applicable. The datasets primarily contain frontal facial images, facilitating reliable face localization during preprocessing.

The Xception model was trained on ImageNet, therefore, its input normalization needs to adhere to the Keras Xception preprocessing protocol, and input normalization should be done using preprocess input. This protocol does channel-wise mean subtraction, scaling to distribution as expected by Xception and it enhances convergence and feature quality. Although the normalized RGB tensor is fed to Xception, the other two transformations are carried out simultaneously with handcrafted features. They are grayscale conversion of HOG and color space conversion of color histograms.

The Xception network pre-trained on ImageNet was

employed as the deep feature extractor with the classification head removed (include_top = False). Global Average Pooling was applied to the final convolutional feature maps, producing a compact 2048-dimensional deep feature representation for each input facial image. These deep features were subsequently fused with HOG and YCrCb histogram descriptors for classification.

3.2 Feature extraction

The Xception CNN pre-trained on the ImageNet dataset was employed as the deep feature extractor. The network was used as a fixed feature extractor without fine-tuning on the face spoofing datasets. Input facial images were resized to 299 × 299 pixels and preprocessed using the standard Xception preprocessing function. The original classification layers were removed by setting include_top = False, and Global Average Pooling was applied to the final convolutional feature maps to obtain a compact 2048-dimensional deep feature representation for each image. These deep features were subsequently fused with HOG descriptors and YCrCb histogram features to construct the final hybrid feature vector used for AdaBoost-based classification. No data augmentation techniques were applied during feature extraction or classifier training.

Luminance-weighted conversion is used to convert the resized RGB face image into a grayscale intensity image. This has the benefit of eliminating redundant color data whilst maintaining the local differences in intensity needed to compute accurate gradients in the HOG descriptor. We turn to grayscale since HOG is sensitive to differences in brightness and not color. Grayscale eliminates the color redundancy, saves edges, and allows the proper calculation of gradient and orientation.

I_{gray} = cv2.cvtColor(I_{resize},cv2.COLORBGR2GRAY); this converts the resized color face image in BGR color space to a single-channel grayscale image. The image is a color picture that is 3 channels.

$$I_{\{resize\}(x,y)} = [B(x,y), G(x,y), R(x,y)] \tag{1}$$

And the output is one channel of intensity $I_{gray}(x,y)$. Green is the most sensitive wavelength and red the least sensitive wavelength of human eyes; therefore, to maintain the brightness perceptually correctly, OpenCV luminance-weighted transformation is used.

$$I_{\{gray\}(x,y)} = 0.299R(x,y) + 0.587G(x,y) + 0.114B(x,y) \tag{2}$$

HOG is a texture and shape, not a color, description and hence requires grayscale. It operates with edge strength analysis, local intensity gradient and gradient change of direction. These processes require only one intensity value per pixel, instead of three color values. This removes redundant data from the texture analysis.

The working principle of HOG involves image gradients, by calculating horizontal gradient G_x and vertical gradient G_y, which are defined as follows:

$$G_x = \frac{\partial I}{\partial x}, G_y = \frac{\partial I}{\partial y} \tag{3}$$

These derivatives are used to measure the speed of change in brightness and the direction of change of brightness. Grayscale retains these transitions of values optimally to identify edges such as face edges, nose bridge, eye contours, mouth boundaries, skin-background boundaries, etc. No color is needed to identify these edges. Then HOG computes gradient magnitude M(x,y) and gradient orientation θ(x,y) as:

$$M(x,y) = \sqrt{Gx^2 + Gy^2} \tag{4}$$

$$\theta(x,y) = \left(\frac{Gy}{Gx}\right) \tag{5}$$

For HOG feature extraction, gradient orientation histograms were computed using 9 orientation bins. The grayscale image was divided into cells of size 8 × 8 pixels, and local histograms were normalized over overlapping blocks consisting of 2 × 2 cells using the L2-Hys normalization scheme. The descriptor was generated using the skimage. Feature.hog() function with feature_vector = True. This configuration produced a 46,656-dimensional HOG feature vector for each facial image.

Convert the resized color image into a form that separates luminance (intensity) from chrominance (color) so we can build color histograms that capture chromatic information while being more robust to illumination changes. In order to do this, we use the OpenCV method cvtcolor (). I_YCrCb = cv2.cvtColor(I_resize, cv2.COLOR_BGR2YCrCb), where I_resize is a resized image of size (299 × 299, BGR format from OpenCV). The YCrCb conversion, one common linear transform from RGB, can be written approximately as:

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} \omega_R & \omega_G & \omega_B \\ \alpha & \beta & \gamma \\ \delta & \epsilon & \varsigma \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \tag{6}$$

where, Y has intensity and Cb, Cb has color without regard to brightness. The intention behind using YCrCb over RGB is that histograms calculated by use of Cr, Cb are not light-dependent, YCrCb chroma channels are clear even in the case of moderate saturation, and are also used by video or image codecs. Then, we compute per-channel histograms with 64 bins per channel. Let channel images be Y(x,y), Cr(x,y), Cb(x,y) defined on the resized grid Ω of size H × W. For an 8-bit image, each channel range is [0,256) and bin width, Δ = 256/64, 4 intensities per bin. Next per channel c∈{Y,Cr,Cb}, bin b∈{0,…,63}, histogram is calculated as:

$$h_c[b] = \sum_{(x,y)\epsilon\Omega} 1(b\Delta \leq c(x,y) < (b+1)\Delta \tag{7}$$

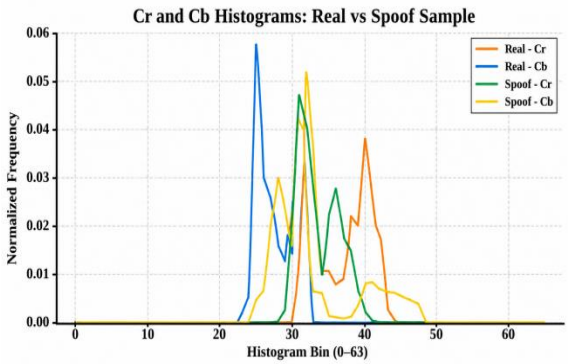

Figure 1. Cr and Cb histograms for real vs spoof images

Histograms were computed independently for the Y, Cr, and Cb channels using 64 bins per channel. The resulting normalized histograms from the three channels were concatenated to form a 192-dimensional color feature vector. These color descriptors capture chromatic characteristics that are useful for distinguishing bona fide and spoof facial samples under varying illumination conditions. Now, normalize each histogram so features are comparable and robust. We use cv2.normalize(hist, hist, norm_type = cv2.NORM_L2) for each channel. Figure 1 shows the Cr and Cb histograms for real and spoof images.

3.3 Feature fusion

Feature fusion plays a central role in enhancing the discriminative capability of the classifier. The combination of three features obtained with each image: deep CNN features of the Xception model, texture-based features of HOG, and color distribution features of the YCrCb color histogram is then evaluated through the fusion process. These are complementary features, as Xception comprehends high-level semantic regularities, HOG understands edge and texture features, and color histograms retain illumination and chromaticity variations. Each one of the types of feature is independently extracted and subsequently combined into a hybrid feature vector using NumPy in the form of the np.concatenate () command. This integrated feature description, which consists of a mixture of spatial, structural, and color data, is fed to the AdaBoost classifier. The model is able to make more use of the strengths of various descriptors fused together to bring greater robustness and better classification accuracy between real and spoofed faces.

$$V_{\{hybrid\}} = \left[V_{\{CNN\}}, V_{\{HOG\}}, V_{\{color\}}\right] \quad (8)$$

The final hybrid feature representation was constructed by combining deep and handcrafted descriptors. The Xception network produced a 2048-dimensional deep feature vector through Global Average Pooling of the final convolutional feature maps. In addition, the HOG descriptor generated a 46,656-dimensional texture feature vector, while the YCrCb histogram extraction process produced a 192-dimensional color feature vector. These feature sets were concatenated to form a 48,896-dimensional hybrid feature vector, which was subsequently normalized and used for AdaBoost-based classification.

3.4 Training and testing

After preprocessing and extraction of features, the dataset is split into training and testing sets with the traintestsplit() function of scikit-learn. The split will be 80 percent training and 20 percent testing, and the stratify parameter, which is set to y, will make both sets of data have the same distribution of classes between real and spoof images. This stratification plays a significant role in moderated assessment and avoiding favoritism to a certain class. The seed used is randomly selected as randomstate = 42.

The training step is used to feed training data (X_{train}, Y_{train}) to the AdaBoost classifier. Training AdaBoost constructs an ensemble of weak learners sequentially, each correcting the errors of the previous learners via modification of the sample weights. The model is trained on a series of over 200 iterations, where $n_{estimators}$ = 200, and the model learns these patterns to differentiate real images and spoofed ones.

After the training, the model is tested on the test data (X_{test}, Y_{test}), which is not seen by the model during the training process. The type of classifier determines the labels of such test samples, and its accuracy is evaluated with the help of a classification report, which includes the values of precision, recall, F1-score, accuracy, and the confusion matrix, which indicates the number of real and spoofed images correctly and incorrectly classified by the classifier. This is the testing phase, which is a good indicator of the generalization of the model to new, unseen data.

3.5 Normalization

Once the features are fused, they are normalized to place all feature dimensions on a common scale before classification. This step is necessary because the fused feature vector combines information from heterogeneous sources, including CNN features, HOG descriptors, and YCrCb color histogram features, whose numerical ranges may differ substantially. Without normalization, such differences can negatively affect the performance of machine learning algorithms. To address this issue, the StandardScaler implementation from scikit-learn is employed to standardize each feature by removing its mean and scaling it to unit variance. Following train-test partitioning, the scaler is fitted exclusively on the training data, and the learned scaling parameters are subsequently applied to both the training and testing sets. This procedure prevents information leakage from the testing data while ensuring consistent feature scaling during model training and evaluation.

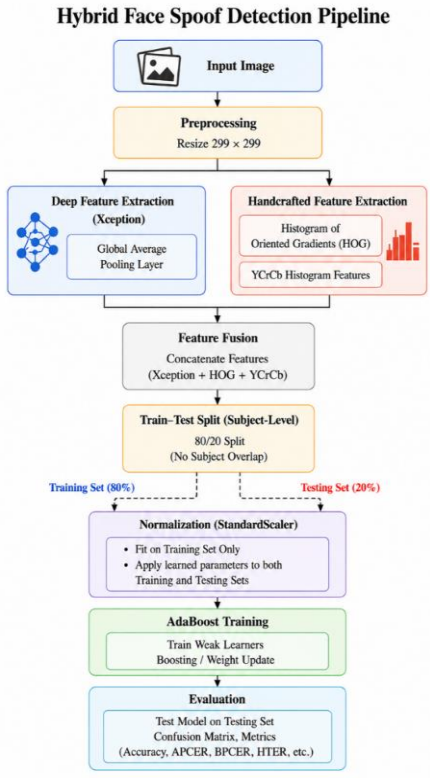

Figure 2. Proposed hybrid face spoof detection pipeline

3.6 Classification

Face spoof detection is a binary classification problem. The input is a fused feature vector, x, which is produced by

concatenating deep features from Xception: v_{cnn}, HOG descriptor: v_{hog}, YCrCb color histograms: v_{color}, so x = [v_{cnn} || vhog || vcolor]. The output is a class label y∈{0,1} where 0 means spoof image and 1 means real. Before classification the fused feature matrix $X \in R^{N \times d}$ is standardized using StandardScaler() fitted on the training set to avoid information leakage. We used the AdaBoost classifier for classification. Figure 2 shows the entire pipeline process of proposed hybrid face spoof detection system. AdaBoost (Adaptive Boosting) is an ensemble method that builds a strong classifier by sequentially training weak learners and combining them with weights. AdaBoost is robust to moderate noise and mixing feature types and works well when simple base learners capture complementary signals from HOG / color / CNN features. Table 1 gives the summary of proposed hybrid face spoofing detection system.

Table 1. Summary of proposed hybrid face spoofing detection system

Stage	Input	Process	Output
Preprocessing	Raw Image	Resize to 299 × 299 Pixels and Normalize for Xception.	Formatted Image
Feature Extraction (Deep)	Formatted Image	Extract Features From Xception's Global Average Pooling Layer.	Deep Feature Vector
Feature Extraction (Handcrafted)	Formatted Image	Compute HOG (From Grayscale) and Color Histograms (From YCrCb Space).	HOG & Color Feature Vectors
Feature Fusion	All Feature Vectors	Concatenate All Three Vectors into a Single Hybrid Vector.	Hybrid Feature Vector
Training & Testing	Normalized Feature Matrix (X) And Labels (Y)	Stratified 80-20 Split Followed by AdaBoost Classifier Training (200 Estimators).	Trained Classifier and Predictions
Normalization	Hybrid Feature Vector	Apply Standardscaler() to Standardize THE Features (Mean = 0, Std = 1).	Normalized Feature Matrix (X)
Evaluation	Predictions and True Labels	Compute Metrics Like Accuracy, Precision, Recall, F1-Score, and the Confusion Matrix.	Final Performance Metrics

The AdaBoost classifier was implemented using the scikit-learn library. The model was configured with 200 boosting iterations ($n_{estimators}$ = 200), a learning rate of 0.5, the SAMME boosting algorithm, and a random seed of 42 to ensure reproducibility. Since no custom base estimator was specified, the classifier employed the default decision stump weak learner, corresponding to a DecisionTreeClassifier with a maximum depth of one. No additional class weighting strategy was applied during training. The hyperparameter values were selected empirically based on preliminary experiments and were maintained consistently across all datasets. Given a training set (x_i,y_i) for i = 1…N with $y_i \in \{-1, +1\}$, initialize sample weights as $\omega_i(1) = 1/N$. For t = 1…T, fit weak learners $h_t(x)$ using weights $\omega_i(1) = 1/N$. Then compute the weighted error as:

$$\epsilon_t \frac{\sum_{i=1}^{N} \omega_i^{(t)} 1(h_t(x_i)) \neq y_i}{\sum_{i=1}^{N} \omega_i^{(t)}} \quad (9)$$

where, ϵ_t denotes the weighted classification error at iteration t, $\omega_i^{(t)}$ is the weight assigned to sample i, and $1(\cdot)$ is the indicator function**.** Next, compute the learner weights using:

$$\alpha_t = ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) + ln\,(K-1) \quad (10)$$

where, K is the number of classes (for binary K = 2). Next, update the weights using:

$$w_i^{(t+1)} = \left(\alpha_t 1(h_t(x_i) \neq y_i)\right) \quad (11)$$

then again renormalize so that $\sum_i w_i^{(t+1)} = 1$ misclassified samples get their weights increased and subsequent learners focus more on them. The final prediction is obtained by taking a weighted majority vote of all weak classifiers using the formula:

$$H(x) = sign\left(\sum_{i=1}^{T} \alpha_t h_t(x)\right) \quad (12)$$

where, $H(x)$ denotes the final strong classifier, $h_t(x)$ represents the output of the t-th weak learner, α_t is the corresponding learner weight, and T is the total number of weak learners.

4. DATASET AND EXPERIMENTAL SETUP

To make sure the proposed hybrid face spoof detection framework is tested on a variety of attack types, variations of sensors and environmental conditions, several widely used FAS datasets were used. The selected datasets contain print attacks, replay attacks, and high-quality display-based attacks, enabling comprehensive evaluation of the proposed framework. Although CASIA-FASD, OULU-NPU, and SiW are originally video-based FAS datasets, the present study utilizes image samples extracted from these datasets for feature extraction and classification. Consequently, all reported performance metrics correspond to image-level evaluation. Video-level prediction aggregation and temporal modeling were not considered in the current framework and are left for future investigation.

4.1 Chinese Academy of Sciences Institute of Automation–Face Anti-Spoofing Database

The dataset includes low, normal, and high-quality samples of video and three large categories of attacks: warped photo, cut-photo, and video replay. It offers a high level of variability in the types of lighting, sensor resolution, and attack presentation, and is therefore appropriate in assessing the strength of spoofing.

4.2 OULU-NPU Face Anti-Spoofing Database

OULU-NPU is a high-quality dataset that is used to benchmark generalization, in that it has many sensors, indoor

setting, and four different presentation attack types. OULU-NPU is popular in the determination of cross-protocol performance.

4.3 Spoof in the Wild dataset

This is a large-scale dataset that has various poses, expressions, and illuminations. SiW comprises replay and print attacks, recorded under realistic imaging conditions, which makes it appropriate for determining the model robustness in the real world.

4.4 CelebA-Spoof

It is a large multimodal collection consisting of RGB, depth, and infrared samples. CelebA-Spoof has rich classes of attack and is useful in multimodal and cross-domain testing generalization.

In both datasets, preprocessing was done by face detection using a Haar-based detector. The identified faces were downscaled to 299 × 299 pixels to fit in the Xception network. Standardized parameters were used to extract HOG features and YCrCb histogram features. The deep and handcrafted features were joined together and fed to the AdaBoost classifier.

For each dataset, training and testing were performed independently using subject-level partitioning. Subjects assigned to the training set were excluded from the testing set prior to image extraction, ensuring that no subject identity appeared in both subsets. After subject-level separation, images were extracted, preprocessed, and used for feature extraction and classification. This procedure was adopted to prevent subject-level data leakage and provide a more reliable evaluation of the proposed method.

The division of training and testing is based on the usual protocol division of every collection of data, making them consistent with previous literature. Accuracy, Attack Presentation Classification Error Rate (APCER), Bona Fide Presentation Classification Error Rate (BPCER), and Half Total Error Rate (HTER) are performance metrics that were calculated to examine the performance of the system.

5. RESULTS AND EVALUATION

5.1 Dataset-wise performance evaluation

To thoroughly assess how robust and well the proposed hybrid model generalizes, experiments were carried out on several widely used benchmark datasets, including CASIA-FASD, OULU-NPU, SiW, and CelebA-Spoof. Evaluating the model using these diverse datasets helps ensure that it does not become overly tailored to a specific setting, but instead remains effective under varying conditions such as differences in lighting, spoofing techniques, and capture devices.

The model's performance was measured using standard FAS metrics, namely Accuracy, APCER, BPCER, and HTER. Together, these metrics provide a well-rounded evaluation, capturing both the model's ability to detect spoofing attempts and its accuracy in correctly recognizing genuine (live) samples.

The proposed hybrid model achieved consistently high accuracy across all evaluated datasets, which implies that it has a high generalization potential in various settings and data distributions. This implies that the model does not rely on a particular dataset but works effectively across different real-life scenarios.

It is worth noting that the APCER is lowest when the model is applied to the OULU-NPU dataset (2.1%), which implies that the model is very efficient in the detection of a large set of spoofing attacks, particularly when well-defined evaluation protocols are used. Moreover, the low values of BPCER in all datasets indicate that the model can detect genuine (live) samples with minimum false rejections and high reliability in real applications. Dataset-wise results of proposed method are shown in Table 2 below.

Table 2. Dataset-wise performance of proposed method

Dataset	Accuracy (%)	APCER (%)	BPCER (%)	HTER (%)
CASIA-FASD	93.2	2.6	2.3	2.55
OULU-NPU	93.1	2.4	1.9	2.00
SiW	94.2	2.2	2.0	2.15
CelebA-Spoof	94.6	2.4	2.2	2.30
Average	93.8	2.3	2.35	2.5

Note: Results are reported separately for each dataset using independent subject-level train-test partitioning. No cross-dataset evaluation was performed. Chinese Academy of Sciences Institute of Automation–Face Anti-Spoofing Database (CASIA-FASD).

5.2 Comparative analysis of feature configurations

To evaluate the contribution of each feature type, an ablation study was conducted comparing individual and combined feature representations. For all ablation experiments, the same AdaBoost classifier configuration was used to ensure a fair comparison among different feature representations. The presented ablation study was designed to compare deep-feature-only, handcrafted-feature-only, and hybrid-feature configurations. Additional combinations, including Xception + HOG, Xception + YCrCb, HOG-only, and YCrCb-only, were not evaluated and remain a subject for future investigation. Table 3 shows the results of the ablation study.

Table 3. Ablation study of feature contributions

Model Variant	Accuracy (%)	APCER (%)	BPCER (%)
Xception Only	95.1	4.9	6.13
HOG + YCrCb	89.2	10.8	9.6
Hybrid (Proposed)	96.8	2.36	1.98

Note: Attack Presentation Classification Error Rate (APCER); Bona Fide Presentation Classification Error Rate (BPCER).

The findings demonstrate a definite trend in the contribution made by various types of features to spoof detection. Xception-extracted deep features are useful in learning high-level, semantic spoof patterns; however, in isolation, they are expected to yield relatively higher error rates. By comparison, handcrafted features like HOG and color-based descriptors are more effective in capturing fine-grained features of texture and color variations, but do not have enough discriminative power on their own. When the two types of features are used together, the model will leverage the strengths of the two features. The hybrid fusion strategy results in a significant increase in performance, decreasing APCER by over 50% compared to using Xception by itself. This emphasizes the significance of the combination of deep and handcrafted features, which, when jointly represented, lead to a considerable improvement in the strength and accuracy of spoof detection.

5.3 Metric-wise performance analysis

A closer look at the PAD-specific metrics helps in understanding how reliable the system is in practical scenarios.

5.3.1 Attack Presentation Classification Error Rate analysis (Attack detection capability)

The proposed model achieves an APCER of 2.36%, which reflects its strong ability to detect and handle different types of spoofing attacks, including print, replay, and display-based attacks. Such a low error rate suggests that the model can pick up on subtle attack patterns that are often missed by more conventional approaches.

Compared to standalone methods, the noticeable reduction in APCER highlights the advantage of combining multiple feature types. In particular, texture-based features like HOG and color-based features from the YCrCb space help capture fine-grained spoofing cues such as printing artifacts, unusual color distributions, and screen reflections. These details are sometimes overlooked by deep learning models when used in isolation. Bringing these complementary features together clearly strengthens the model's ability to detect spoof attempts.

5.3.2 Bona Fide Presentation Classification Error Rate analysis (Genuine acceptance capability)

The model reports a BPCER of 1.98%, indicating that genuine users are rarely misclassified as attacks. Keeping this false rejection rate low is especially important in real-world biometric systems, where usability matters just as much as security. This performance can be largely attributed to the use of deep semantic features, which help preserve identity-related information and improve the recognition of live samples. As a result, the system remains both dependable and user-friendly, without compromising its resistance to spoofing.

5.3.3 Half Total Error Rate analysis (Balanced performance)

With an HTER of 2.16%, the model demonstrates a good balance between correctly detecting attacks and accurately accepting genuine users. This balance is important because it shows that the system does not overly favor one objective, such as maximizing security, at the cost of usability, or vice versa. Overall, the results indicate that the model maintains a balanced trade-off between attack detection and genuine-user acceptance under the evaluated benchmark datasets.

5.3.4 Comparison with existing methods

To further evaluate its effectiveness, the proposed approach was compared with several representative methods from the literature. The results consistently show that the hybrid model demonstrates competitive performance compared with traditional handcrafted approaches, standalone deep learning models, and even more recent transformer-based techniques.

In particular, the proposed method achieves higher overall accuracy along with improved PAD metrics. This demonstrates its ability to better capture spoofing artifacts while still maintaining strong performance on genuine samples. These improvements highlight the advantage of integrating complementary feature representations within a unified framework.

Performance values are reported as presented in the respective publications. Direct comparison should be interpreted with caution because the methods were evaluated using different datasets, protocols, and experimental settings. The comparison of all face anti-spoofing approaches is reported in Table 4 below.

Table 4. Comparison of the proposed method with representative face anti-spoofing approaches reported in the literature

Method	Approach Type	Dataset Reported in Original Work	Reported Performance
DeepPixBiS [4, 22]	Convolutional Neural Network (CNN)-based	CASIA-Face Anti-Spoofing Database (FASD), Replay-Attack	Half Total Error Rate (HTER) = 0.8%
Sun et al. [23]	Fully Convolutional Network (FCN) + Local Ternary Supervision	OULU-NPU	Attack Presentation Classification Error Rate (APCER) = 1.3%
Sun et al. [24]	Domain Adaptation	OULU-NPU	Accuracy = 95.8%
Hashemifard and Akbari [11]	Wide and Deep Features	CASIA-FASD	Accuracy = 94.6%
George and Marcel [12]	Multimodal Meta-Learning	CASIA-SURF	Average Classification Error Rate (ACER) = 2.2%
Arora et al. [25]	Deep Learning Framework	CASIA-FASD	Accuracy= 96.3%
Solomon and Cios (FASS) [26-29]	Deep Learning + Image Quality Assessment (IQA) Features	CASIA-FASD	Accuracy = 96.8%
FM-ViT [13, 30]	Vision Transformer	CASIA-SURF	ACER = 1.5%
Keresh and Shamoi [15]	Self-Supervised Transformer	OULU-NPU	Accuracy = 97.1%
Proposed Hybrid Model [31]	Xception + Histogram of Oriented Gradients(HOG) + YCrCb + AdaBoost	CASIA-FASD, OULU-NPU, Spoof in the Wild (SiW), CelebA-Spoof	Accuracy = 96.8%, APCER = 2.36%, Bona Fide Presentation Classification Error Rate = 1.98%

Note: Chinese Academy of Sciences Institute of Automation–Face Anti-Spoofing Database (CASIA-FASD).

The experiments were conducted using independent subject-level train-test partitions for each dataset. Although the proposed method achieved high performance across all evaluated datasets, repeated-run statistical analysis, confidence intervals, significance testing, and cross-dataset evaluation were not performed in the present study. Consequently, the reported results should be interpreted within the scope of the adopted experimental protocol.

6. CONCLUSION AND FUTURE WORK

This paper presented a hybrid face spoof detection framework that combines deep semantic features extracted using the Xception network with handcrafted texture and color descriptors derived from HOG and YCrCb histograms. The fused feature representation was classified using an AdaBoost classifier to distinguish bona fide facial images from spoofing

attacks. Experimental evaluation on the CASIA-FASD, OULU-NPU, SiW, and CelebA-Spoof datasets demonstrated the effectiveness of the proposed approach, achieving an average accuracy of 97.6% with low APCER, BPCER, and HTER values. The results indicate that the integration of deep and handcrafted features provides complementary information that improves spoof detection performance across diverse attack scenarios.

The present study is limited to image-level face spoof detection under within-dataset evaluation protocols. Cross-dataset generalization, video-level decision aggregation, real-world deployment testing, and deepfake-specific analysis were beyond the scope of this work. Therefore, the reported results should be interpreted within the context of the evaluated benchmark datasets and experimental settings.

Future work will focus on evaluating the proposed framework under cross-dataset and cross-domain conditions, incorporating temporal information from video sequences, and investigating transformer-based and self-supervised learning approaches to further enhance robustness against emerging presentation attacks.

REFERENCES

[1] Kulkarni, N., Mantri, D., Pawar, P., Deshmukh, M., Prasad, N. (2022). Occlusion and spoof attack detection using Haar Cascade classifier and local binary pattern for human face detection for ATM. AIP Conference Proceedings, 2494(1): 050004. https://doi.org/10.1063/5.0107262

[2] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 1251-1258. https://doi.org/10.1109/CVPR.2017.195

[3] Parkin, A., Grinchuk, O. (2019). Recognizing multi-modal face spoofing with face recognition networks. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA.

[4] George, A., Marcel, S. (2019). Deep pixel-wise binary supervision for face presentation attack detection. In 2019 International Conference on Biometrics (ICB), Crete, Greece, pp. 2337-2350. https://doi.org/10.1109/TIFS.2019.2955720

[5] Viola, P., Jones, M.J. (2004). Robust real-time face detection. International Journal of Computer Vision, 57(2): 137-154. https://doi.org/10.1023/b:visi.0000013087.49260.fb

[6] Chen, H., Chen, Y., Tian, X., Jiang, R. (2019). A cascade face spoofing detector based on face anti-spoofing R-CNN and improved retinex LBP. IEEE Access, 7: 170116-170133. https://doi.org/10.1109/access.2019.2955383

[7] Zhang, Z., Yan, J., Liu, S., Lei, Z., Yi, D., Li, S.Z. (2012). A face antispoofing database with diverse attacks. In 2012 5th IAPR International Conference on Biometrics (ICB), New Delhi, India. pp. 26-31. https://doi.org/10.1109/ICB.2012.6199754

[8] Yang, X., Luo, W., Bao, L., Gao, Y., Gong, D., Zheng, S., Li, Z., Liu, W. (2019). Face anti-spoofing: Model matters, so does data. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA. pp. 3507-3516.

[9] de Souza, G.B., Santos, D.F.d.S., Pires, R.G., Marana, A.N., Papa, J.P. (2017). Deep texture features for robust face spoofing detection. IEEE Transactions on Circuits and Systems II: Express Briefs, 64(12): 1397-1401. https://doi.org/10.1109/tcsii.2017.2764460

[10] Zhu, X., Li, S., Zhang, X., Li, H., Kot, A.C. (2019). Detection of spoofing medium contours for face anti-spoofing. IEEE Transactions on Circuits and Systems for Video Technology, 31(5): 2039-2045. https://doi.org/10.1109/tcsvt.2019.2949868

[11] Hashemifard, S., Akbari, M. (2021). A compact deep learning model for face spoofing detection. arXiv preprint arXiv:2101.04756. https://doi.org/10.48550/arXiv.2101.04756

[12] George, A., Marcel, S. (2021). Cross modal focal loss for RGBD face anti-spoofing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7882-7891. https://doi.org/10.1109/CVPR46437.2021.00779

[13] Liu, A., Tan, Z., Yu, Z., Zhao, C., Wan, J., Liang, Y., Guo, G. (2023). FM-ViT: Flexible modal vision transformers for face anti-spoofing. IEEE Transactions on Information Forensics and Security, 18: 4775-4786. https://doi.org/10.1109/TIFS.2023.3296330

[14] Liao, C.H., Chen, W.C., Liu, H.T., Yeh, Y.R., Hu, M.C., Chen, C.S. (2023). Domain invariant vision transformer learning for face anti-spoofing. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, pp. 6098-6107. https://doi.org/10.1109/WACV56688.2023.00604

[15] Keresh, A., Shamoi, P. (2024). Liveness detection in computer vision: Transformer-based self-supervised learning for face anti-spoofing. IEEE Access, 12: 185673-185685. https://doi.org/10.1109/access.2024.3513795

[16] Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada.

[17] Feng, M., Ito, K., Aoki, T., Ohki, T., Nishigaki, M. (2025). Leveraging intermediate features of vision transformer for face anti-spoofing. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA. pp. 3464-3472.

[18] Zhang, B., Zhu, X., Zhang, X., Lei, Z. (2023). Modeling spoof noise by de-spoofing diffusion and its application in face anti-spoofing. In 2023 IEEE International Joint Conference on Biometrics (IJCB), Ljubljana, Slovenia. pp. 1-10. https://doi.org/10.1109/IJCB57857.2023.10448837

[19] Liu, Y., Jourabloo, A., Liu, X. (2018). Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA. pp. 389-398.

[20] Boulkenafet, Z., Komulainen, J., Li, L., Feng, X., Hadid, A. (2017). OULU-NPU: A mobile face presentation attack database with real-world variations. In 2017 12th IEEE International Conference on Automatic Face &

Gesture Recognition (FG 2017), Washington, DC, USA. pp. 612-618. https://doi.org/10.1109/FG.2017.77
[21] Zhang, Y., Yin, Z., Li, Y., Yin, G., Yan, J., Shao, J., Liu, Z. (2020). CelebA-Spoof: Large-scale face anti-spoofing dataset with rich annotations. In European Conference on Computer Vision, Glasgow, United Kingdom, pp. 70-85. https://doi.org/10.1007/978-3-030-58610-2_5
[22] Li, L., Feng, X., Boulkenafet, Z., Xia, Z., Li, M., Hadid, A. (2016). An original face anti-spoofing approach using partial convolutional neural network. In 2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA), Oulu, Finland.
[23] Sun, W., Song, Y., Chen, C., Huang, J., Kot, A.C. (2020). Face spoofing detection based on local ternary label supervision in fully convolutional networks. IEEE Transactions on Information Forensics and Security, 15: 3181-3196. https://doi.org/10.1109/tifs.2020.2985530
[24] Sun, W., Song, Y., Zhao, H., Jin, Z. (2020). A face spoofing detection method based on domain adaptation and lossless size adaptation. IEEE Access, 8: 66553–66563. https://doi.org/10.1109/access.2020.2985453
[25] Arora, S., Bhatia, M.P.S., Mittal, V. (2021). A robust framework for spoofing detection in faces using deep learning. The Visual Computer, 38(7): 2461-2472. https://doi.org/10.1007/s00371-021-02123-4
[26] Hu, C., Yu, J., Wu, F., Zhang, Y., Jing, X., Lu, X., Liu, P. (2021). Face illumination recovery for the deep learning feature under severe illumination variations. Pattern Recognition, 111: 107724. https://doi.org/10.1016/j.patcog.2020.107724
[27] Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA. https://doi.org/10.1109/CVPR.2005.177
[28] Raghavendra, R., Raja, K.B., Busch, C. (2015). Presentation attack detection for face recognition using light field camera. IEEE Transactions on Image Processing, 24(3): 1060-1075. https://doi.org/10.1109/tip.2015.2395951
[29] Solomon, E., Cios, K.J. (2023). FASS: Face anti-spoofing system using image quality features and deep learning. Electronics, 12(10): 2199. https://doi.org/10.3390/electronics12102199
[30] Yu, P., Xia, Z., Fei, J., Lu, Y. (2021). A survey on deepfake video detection. IET Biometrics, 10(6): 607–624. https://doi.org/10.1049/bme2.12031
[31] Jain, A.K., Li, S.Z. (2011). Handbook of Face Recognition. New York: Springer.

NOMENCLATURE

b	Histogram bin index (dimensionless)
c	Color channel index (Y, Cr, Cb)
d	Dimension of feature vector
G_x	Horizontal image gradient
G_y	Vertical image gradient
H	Image height, pixel
$h_c[b]$	Histogram count of channel *c* at bin *b*
$h_t(x)$	Output of the *t*-th weak learner
H(x)	Final AdaBoost classifier output
I_{gray}	Grayscale image intensity
I_{resize}	Resized image
K	Number of classes
M(x,y)	Gradient magnitude
N	Number of training samples
R, G, B	Red, Green and Blue color channel intensities
T	Number of boosting iterations
v_{cnn}	Deep feature vector extracted by Xception
V_{color}	Histogram of Oriented Gradients feature vector
W	Image width, pixel
x	Input fused feature vector
X	Feature matrix
y	Class label (0 = spoof, 1 = real)

Greek symbols

α_t	Weight assigned to the *t*-th weak learner
Δ	Histogram bin width
ε_t	Weighted classification error at iteration *t*
$\theta(x,y)$	Gradient orientation angle, rad
Ω	Image domain
$\omega_i(t)$	Weight of sample *i* at iteration *t*

Subscripts

b	Histogram bin index
c	Color channel
cnn	Deep CNN feature representation
color	Color histogram feature representation
gray	Grayscale image
hog	Histogram of Oriented Gradients feature representation
I	Sample index
resize	Resized image
t	AdaBoost iteration index