Al-Rafidain Journal of Engineering Sciences

Journal homepage https://rjes.iq/index.php/rjes

ISSN 3005-3153 (Online)

A Comprehensive Review of Real-Time Deepfake Detection Using Light Distribution and Illumination Consistency Analysis

Islam Mu'ayyad Dhiyab, Mohammed Sahib Mahdi Altaei

Al-Nahrain University/College of Science/Department of Computer Science

ARTICLE INFO

Article history:
Received 16 February 2026
Revised 16 February 2026
Accepted 10 March 2026,
Available online 17 March 2026

Keywords:
Deepfake video detections
Machine Learning,
Deep learning

ABSTRACT

The rapid evolution of deep learning and generative models has enabled the creation of highly realistic deepfake videos, raising substantial concerns regarding digital security, privacy, and public trust. Conventional detection methods—often based on static visual artifacts, spatial irregularities, or frequency-domain anomalies—struggle to provide reliable performance in real-time environments. This review presents a comprehensive analysis of emerging real-time deepfake detection techniques that leverage physical light behavior, including illumination distribution, corneal reflection analysis, and vibration-induced blurriness. By exploiting the intrinsic interaction between light and real facial surfaces, these methods introduce active physical probing signals that current generative models cannot accurately replicate. Experimental results reported in the literature consistently demonstrate that illumination-based and physical-response approaches achieve high temporal consistency, robustness, and accuracy, surpassing traditional data-driven algorithms in live video settings. Nonetheless, several challenges remain unresolved, such as variable lighting conditions, device hardware limitations, compression artifacts, and the scarcity of annotated datasets designed specifically for illumination-based analysis. To address these limitations, future research should integrate hybrid frameworks that combine physical-light cues with advanced deep learning models—such as YOLO-based architectures—towards developing reliable, explainable, and real-time deepfake authentication systems.

1. Introduction

In today's society, most people use advanced lenses and cameras. In addition, with the applications developed for the digital environment, it has become very easy for users to share and upload images to the internet. It has become easier to access users'personal information, videos and images in these environments developed for people to share [1]. The growing sophistication of artificial intelligence has led to the emergence of *deepfakes*—AI-generated synthetic media that mimic real human appearance, speech, or behavior with remarkable accuracy. While deepfakes were initially developed for creative and entertainment purposes, they have increasingly become tools for deception and fraud. Deepfakes is a widely used technology to generate fake content from real images and sounds using deep learning techniques [2, 3]. The most frequently used

Corresponding author E-mail address: islammm89@gmail.com
https://doi.org/10.61268/5twsxv77

deepfake production method uses face replacement with deep neural networks and automatic encoders [4]. In this method, the target video and several images of the face desired to be used in this video are generally used to create a deepfake [5]. Deepfakes are fake media content created using artificial intelligence to create fake news agendas, fake political agendas or personal attacks. When used for malicious purposes, deepfakes can harm individuals' reputations by sabotaging personal data security. Since there is no law prohibiting deepfakes today, detecting deepfakes is an important element in separating real images and fake image data and ensuring their security.

In 2019, a United Kingdom–based company became one of the first known victims of AI-synthesized content in a financial scam. The company's CEO was deceived into transferring approximately $243,000 (USD) after receiving a phone call that appeared to come from his superior. The voice, later discovered to be AI-generated, mimicked the CEO's tone and accent with convincing precision. In early 2020, a bank in the United Arab Emirates fell victim to a sophisticated deepfake attack resulting in a financial loss of approximately $35 million USD. The fraudsters used AI-generated voice synthesis to convincingly imitate the voice of a company director known to the bank. The caller instructed the bank manager to transfer funds as part of an alleged corporate acquisition, a story supported by previously received legitimate-looking emails. Believing the voice to be authentic, the manager approved the transfer. Subsequent investigations revealed that the voice had been synthetically generated using deep learning-based speech synthesis, demonstrating the alarming potential of AI in facilitating high-impact fraud and identity impersonation.

This case underscores the growing sophistication of deepfake technologies, which can now manipulate not only images and videos but also audio streams in real time. The incident reveals how trust mechanisms—such as voice familiarity or contextual alignment—can be exploited by deepfake systems. It also motivates the urgent need for robust, real-time detection techniques that can identify manipulation cues, including light distribution inconsistencies in videos, which are difficult for generative models to replicate accurately. Thus, illumination-based detection presents a promising direction for strengthening multimedia authenticity verification systems.

Deepfake is an artificial intelligence technology that produces realistic fake images and videos. It is a form of artificial intelligence that uses deep learning techniques. Methods such as autoencoders and Generative Adversarial Networks (GANs) are typically applied to alter facial expressions, speech, and movement in a way that makes it difficult to distinguish between authentic and synthetic content. The use of Generative Adversarial Networks (GANs) in the production of deepfake images is quite common [6].

The notable advances in artificial neural network (ANN)–based technologies have played a crucial role in enabling the manipulation and generation of realistic multimedia content. AI-enabled software tools such as *FaceApp* [7] and *FakeApp* [8]

have made it possible to perform highly convincing face swapping in both images and videos. This face-swapping mechanism allows anyone to alter physical features such as facial expression, hairstyle, gender, or age, producing synthetic yet realistic human appearances. The widespread accessibility of these tools has contributed to the emergence of what is now widely known as Deepfake technology—synthetic media generated through deep learning algorithms that mimic real individuals with high visual fidelity.

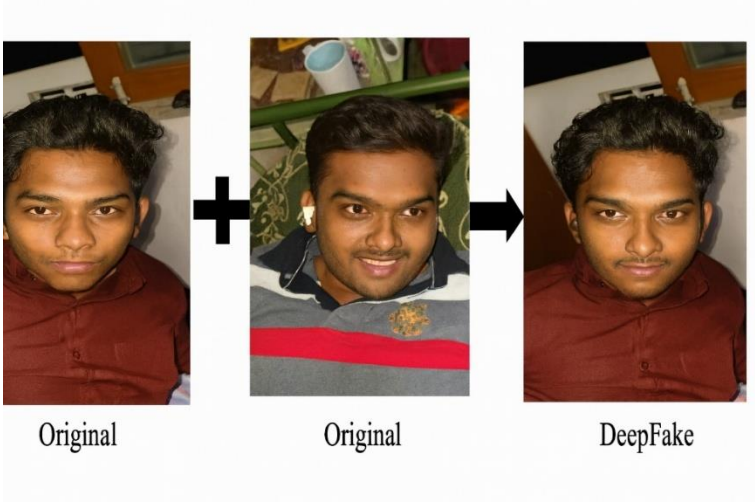

Figure 1. Deepfake Generation Process

The primary objective of this review is to provide a comprehensive analysis of real-time deepfake detection techniques based on light distribution and illumination consistency. It aims to identify existing methodologies, evaluate their performance, and highlight their strengths and limitations. Furthermore, this review seeks to establish how light-based analysis can enhance the accuracy, efficiency, and reliability of real-time deepfake detection systems, thereby contributing to the development of more robust multimedia authentication frameworks.

2. Deepfake Generation Overview

Over the years, a vast amount of digital data has been generated and made publicly available across various internet platforms and social media networks. With the rapid advancement of machine learning (ML) and deep learning (DL) techniques—particularly Generative Adversarial Networks (GANs)—the production of realistic fake media, commonly known as *deepfakes*, has become increasingly easy [9].

Several deepfake generation techniques have been developed to manipulate human facial features in digital media. The most common manipulation types include identity swap, expression swap, and complete face synthesis. These techniques rely heavily on large, publicly available datasets that provide diverse visual information about human faces under different conditions. Machine learning plays

a central role in the creation of deepfakes. The process typically involves training deep neural networks, a subfield of machine learning, to learn the target person's facial characteristics under varying conditions such as lighting, facial angles, and expressions. During this process, models are divided into training and testing phases. In the training phase, the model learns to generate synthetic or altered videos, while in the testing phase, the model output is evaluated or used for deepfake detection [10], [11].

The technique of deepfake generation plays a significant role in advancing traditional forgery creation methods by reducing artifacts and eliminating manipulation traces that detection systems have typically relied on [12], [13]. Based on deep learning (DL) principles, deepfake generation has revolutionized the process of extracting input attributes and reconstructing them to produce highly realistic manipulated images and videos. Several DL-based approaches are widely used in deepfake generation systems, including autoencoders, autoregressive models [14], and Generative Adversarial Networks (GANs) [15]. These artificial intelligence algorithms are primarily designed for unsupervised data representation learning, where the goal is to learn efficient latent representations of input data without explicit supervision.

From a technical perspective, these models operate by encoding input data into a hidden latent space representation and subsequently reconstructing the output through a decoding process. Among these, the autoencoder plays a crucial role in many deepfake generation tools. The autoencoder network learns to extract meaningful features from input images while minimizing irrelevant noise. By sampling from a Gaussian distribution to generate latent representations and feeding them into the encoder, the model can synthesize new, manipulated images. The encoder compares pixel-level details between input and latent data, while the decoder reconstructs the final output image from this compressed representation [16].

The autoregressive model represents another important class of DL-based generation system. It is a statistical model that focuses on modeling the natural image distribution, where the conditional probability of each pixel depends on the previously generated pixels [17]. However, the evaluation and prediction processes in autoregressive models are often computationally intensive, as each pixel must be generated and processed sequentially [18], [19]. These models analyze pixel correlations to differentiate between manipulated and authentic content, where a lower inter-pixel correlation may indicate possible image tampering.

Another major approach is the Generative Adversarial Network (GAN), which remains the backbone of most modern deepfake generation systems. A GAN consists of two neural networks—the generator and the discriminator—that compete in a minimax optimization process [20]. The generator aims to produce realistic synthetic outputs that can fool the discriminator, while the discriminator learns to distinguish between genuine and fake data. Through iterative

backpropagation and optimization, both networks improve until reaching an equilibrium between real and generated outputs. Numerous deepfake generation applications have been developed using GAN architectures, including FaceApp [21], Faceswap [22], ZAO [23], RCNN-based Super-Resolution systems (e.g., *Dong et al., Image Super-Resolution Using Deep Convolutional Networks*), and StackedGAN frameworks for enhancing low-quality videos [24].These tools demonstrate the capability of deep learning to produce increasingly realistic and convincing synthetic media, posing new challenges for detection systems and digital forensics.

There are four primary categories of deepfake generation, particularly focused on facial manipulation. *Figure 2* illustrates the various types of deepfake generation techniques applied to human faces:

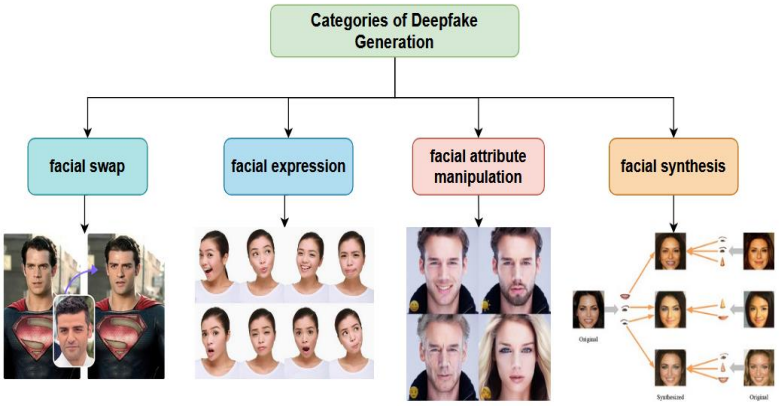

Figure 2. Types of Deepfake Generation Methods

- Face Swap

The face swap technique aims to replace an original face with a target face while preserving the original facial expressions [20]. In 2018, researchers proposed a latent space–based face-swapping approach for both face and hair regions using a Generative Adversarial Network (GAN) framework [25]. This approach employed two Variational Autoencoders (VAEs) to encode the face and hair regions into latent representations, followed by a GAN-based synthesis module for generating the swapped output. The main limitation of this method was its restriction to low-resolution images (128×128). To improve quality, the authors later integrated a Deep Neural Network (DNN) and an additional VAE to enhance latent feature extraction and reconstruction [26]. In 2019, a Recurrent Neural Network (RNN)–based method was proposed to further improve the realism of face swapping [27]. This method included

three main components: A U-Net–based Recurrent Reenactment Generator (GR), a Pix2PixHD-based Segmentation Generator (GS), and a Pix2PixHD-based Inpainting Generator (GC). The GR extracts the target's pose and expression to create a reenacted face, GS generates segmentation masks for the face and hair, and GC reconstructs missing or occluded areas. To maintain temporal coherence during face interpolation, Delaunay triangulation and barycentric coordinates were applied. However, resolution inconsistencies across different viewing angles remained a challenge. In 2020, researchers from Peking University and Microsoft introduced FaceShifter, a method designed to handle occlusion challenges in face swapping [28]. This model integrates two components: The Attribute-Embedded Integration Network (AEINet), which denormalizes local feature integration across multiple levels, and the Heuristic Error Acknowledging Network (HEARnet), which detects occlusion regions by leveraging heuristic error between input and manipulated images. FaceShifter demonstrated state-of-the-art performance in realistic facial swapping under complex conditions.

- Facial Expression

The facial expression deepfake technique, also known as Face-to-Face reenactment, transfers facial expressions from a source person to a target person, allowing the target to appear as if expressing emotions or speech never performed in reality. In 2018, Choi et al. proposed StarGAN, a multi-domain translation network based on CycleGAN, which uses a single model to handle multiple facial expression mappings using mask vectors [29]. In the same year, Wu et al. enhanced boundary mapping using CycleGAN and an encoder–decoder (Pix2Pix) network for reconstructing synthetic expressions [30]. Bansal et al. introduced RecycleGAN, which integrated cycle consistency loss, adversarial loss, and a recycle formulation to capture spatiotemporal coherence in generated videos [31]. Song et al. applied Mel-Frequency Cepstral Coefficients (MFCC) to extract audio features and trained a conditional recurrent network to synchronize lip movement with speech in an adversarial framework [32]. In 2020, Soumya et al. proposed the Interpretable and Controllable Face Reenactment Network (ICface). It works in two stages: first, extracting facial attributes such as action unit (AU) values and head pose angles; second, integrating them with the input image through a GAN to control pose and expression transformations. ICface demonstrated strong performance with minimal distortion but required improvement in handling visual artifacts [33]. Yunlian et al. later proposed Ordinal Ranking Adversarial Networks (ORAN), combining StarGAN and CycleGAN to rank facial expressions and age intensity using a multiscale discriminator and one-hot labels [34]. In 2020, another method using 3D convolutional filters and a spatiotemporal scheme was introduced to generate high-quality deepfake videos from static images [35]. However, it struggled with high-resolution texture consistency. In 2021, Chaoyou et al. presented a semi-supervised encoder–decoder approach utilizing

LightCNN to map and reconstruct facial boundaries for expression and pose synthesis. They also developed the MVF-HQ dataset to support high-resolution expression reenactment research [36]. Prominent tools for expression transfer include Jiggy [37] and Impersonator++ [38].

- Facial Attribute Manipulation

Facial attribute manipulation modifies personal features such as hairstyle, eye color, skin tone, gender, age, or wrinkles, allowing significant changes in a person's appearance [20]. The StarGAN framework supports multi-domain attribute translation using mask vectors [29]. Xiao et al. proposed ELEGANT, a CycleGAN-based translation network, which achieved realistic attribute changes but suffered from minor artifacts [39]. To transfer makeup styles, Li et al. developed BeautyGAN, which introduced pixel-level makeup loss and perceptual loss functions to preserve identity and minimize artifacts [40]. In 2019, AttGAN was proposed to enhance image realism and reduce artifacts by ensuring semantic attribute consistency, though it struggled with large-scale attribute transformations [41]. Lue et al. later presented STGAN, which introduced Selective Transfer Units (STUs) within the encoder–decoder framework, achieving better synthesis quality [42]. Jo et al. proposed SC-FEGAN, a GAN-based freeform editing system that allows users to modify facial features using sketches, colors, or masks. It employed holistically nested edge detection and histogram equalization to improve edge definition and realism [43]. In 2021, the URCA-GAN architecture was introduced to modify specific attributes differently between input and target images using URCAM and StarGAN modules [44]. Later studies [45], [46] further refined loss functions to preserve identity, expression, and age consistency. Affifi et al. introduced HistoGAN, a StyleGAN-based model that performs natural skin tone modification using a color histogram–based generative approach [47]. The most widely used application for facial attribute manipulation is FaceApp [19].

- Facial Synthesis

Facial synthesis generates entirely artificial yet realistic faces by learning latent representations from large-scale datasets. It is commonly used in gaming and 3D modeling, but poses serious ethical concerns when applied to fabricate real individuals for malicious purposes. In 2019, Karras et al. introduced ProGAN, a progressive GAN architecture that employed Adaptive Instance Normalization (AdaIN) to achieve high-quality image synthesis [48]. However, its output occasionally contained noticeable artifacts. In 2020, they enhanced this approach with StyleGAN, which achieved superior image quality, texture consistency, and fewer artifacts [49]. Both ProGAN and StyleGAN architectures are now widely used to generate facial synthesis datasets and represent the foundation of modern deepfake creation.

3. Lighting Inconsistencies in Deepfakes

The interaction between light and facial surfaces plays a fundamental role in revealing the authenticity of visual media. Genuine videos exhibit consistent illumination behavior, following physical laws of light propagation, shadow formation, and reflection. In contrast, deepfake videos often demonstrate lighting inconsistencies because the generative models (such as GANs or autoencoders) are optimized for visual realism rather than photometric accuracy. These inconsistencies can appear as misaligned shadows, unnatural color tones, or temporal mismatches in lighting changes, especially in dynamic scenes such as live video conferencing.

Gerstner et al. (CVPR 2022) [50] introduced a real-time deepfake detection method based on active illumination, leveraging the principle that a real human face naturally responds to controlled lighting changes, while a synthetic face cannot accurately reproduce these variations in real time. The method incorporates an active illumination source into the video-conferencing interface by displaying a fixed-size, uniform-color image whose hue H(t) is modulated sinusoidally over time. As shown in the paper, the hue changes according to:

$$H(t) = 0.1307 \times \cos\left(\frac{t}{8}\right), t \in [0,16]$$

Producing a smooth oscillation between yellowish and magenta hues. Because hue is circular, negative values wrap around (–h ≡ 1–h). These colors were specifically chosen because, unlike blues or greens, they do not trigger automatic white-balancing on common webcams. Saturation and value are kept constant at 1, ensuring isoluminant color modulation where brightness remains unchanged, and HSV is converted to RGB using standard routines (e.g., Python's *colorsys.hsv_to_rgb*).

Figure 2 illustrates this dynamic illumination process, showing the time-varying hue of the uniform light source (top) and nine simulated 3-D renderings of a face illuminated at different hue values (bottom). The simulation assumes a monochromatic facial reflectance and a 1:1 ratio of active to ambient light, with saturation boosted by 50% for visualization.

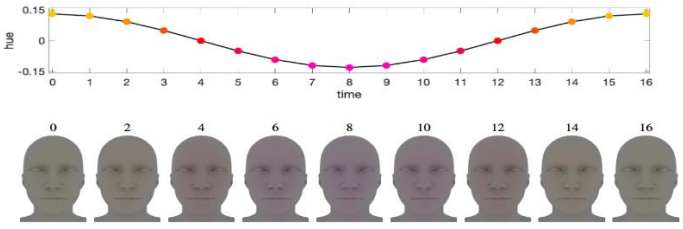

Figure 3. Dynamic change in the hue of the uniform-color area light source over the 16-second sequence (top), and nine simulated 3-D face renderings illuminated with matching hue values at distinct time points (bottom).

Face localization is performed using Dlib's 68-landmark detector. An elliptical mask is fitted using keypoints on the chin, nose bridge, and cheekbones to define the stable facial region used for hue measurement. The system then addresses the challenge of

separating ambient lighting from the artificially modulated illumination. Using a simplified Lambertian reflectance model, the authors assume the face under normal conditions is illuminated by nondirectional white light. The contribution of the active illumination is extracted by dividing each pixel's RGB values by the baseline facial color measured before the illumination sequence begins. This isolates the true hue response caused only by the modulated light source, removing environmental effects. For each frame, the RGB facial pixels (after baseline division) are converted to HSV, and hue values are averaged using circular statistics to account for hue periodicity:

$$\tilde{H}(t) = \frac{1}{2\pi}\text{atan2}\left(\sum_{i=1}^{n}\sin\left(2\pi h_i(t)\right), \sum_{i=1}^{n}\cos\left(2\pi h_i(t)\right)\right)$$

The measured facial hue $\tilde{H}(t)$ is then compared to the expected illumination sequence $H(t)$ using the Pearson circular correlation coefficient, where values near 1 indicate perfect temporal synchronization. Because the method is computationally efficient, the system can validate a 30-frame pattern—at 30 fps—with only a one-second delay.

Under ideal conditions, the method achieved a correlation of 0.99 in simulations. Real-world tests across 15 participants reached correlations of 0.93 using larger illumination areas, with performance decreasing proportionally for smaller sources. In contrast, deepfake videos showed negligible correlation values (average 0.09), exposing their inability to mimic real-time illumination responses. The authors further evaluated adversarial scenarios and found that even small temporal reproduction delays (as little as 2/30 seconds) reduced correlation below 0.5, highlighting a key temporal vulnerability in current generative models.

To further evaluate the robustness of the approach, the authors conducted extensive simulation experiments using the physically based renderer Mitsuba [51]. The scene consisted of a camera with a 90° field of view and a 3-D head model exhibiting Lambertian reflectance and a neutral skin tone, positioned 2 ft in front of the camera. Illumination included a unit-value ambient light and an area light of size 9×99 \times 99×9 in placed beside the camera to mimic the behavior of an illuminated computer display. The simulations systematically varied key parameters—including skin tone (Figure 4(a)), head-to-camera distance of 16, 20, 24, 30, and 36 inches (Figure 4(b)), active light source size of 5×55 \times 55×5, 7×77 \times 77×7, 9×99 \times 99×9, 11×1111 \times 1111×11, and 13×1313 \times 1313×13 inches (Figure 4(c)), and ambient light intensity levels of 0.0, 0.5, 1.0, 1.5, and 2.0 (Figure 4(d)). Across all simulated imaging configurations, the average correlation between the measured hue and the induced illumination hue was 0.99, with a minimum correlation of 0.988, demonstrating that the proposed technique is highly robust under idealized assumptions before transitioning to real-world validation.

Figure 4. A representative sample of our simulated data set with varying: (a) skin tone; (b) head-to-camera distance (increasing from left to right); (c) size of the active light source (increasing from left to right); and (d) intensity of ambient light (increasing from left to right)

The real-world data set was recorded from 15 users with diverse skin tones and across a variety of environments. Participants were positioned approximately 24 inches from both the display and the camera, while the size of the active illumination ranged from 13×1313 \times 1313×13 to 3×33 \times 33×3 inches in 2-inch increments. Each recording included two full cycles of the hue pattern shown previously in Figure 3; with the display and camera synchronized at 30 Hz, the entire active-illumination sequence was visible for only one second. As shown in the upper panel of Figure 5, the correlation between the measured facial hue and the induced illumination hue decreases systematically with diminishing illumination size. At the largest illumination size of 13 inches, the average correlation was 0.93. As the illumination size decreased to 11, 9, 7, 5, and 3 inches, the average correlations progressively declined to 0.92, 0.85, 0.83, 0.68, and 0.33, respectively. For comparison, the authors also measured the correlation between the expected illumination pattern and facial appearance in the absence of active illumination—a condition representative of deepfakes that do not transfer environmental lighting. Across all 15 users, this baseline condition produced an average correlation of 0.09, with a variance of 0.01, and a maximum value of 0.34. Notably, when the active light source exceeded 5×55 \times 55×5 inches, the vast majority of correlations were greater than 0.5, providing a clear separation between real illumination responses and synthetic content.

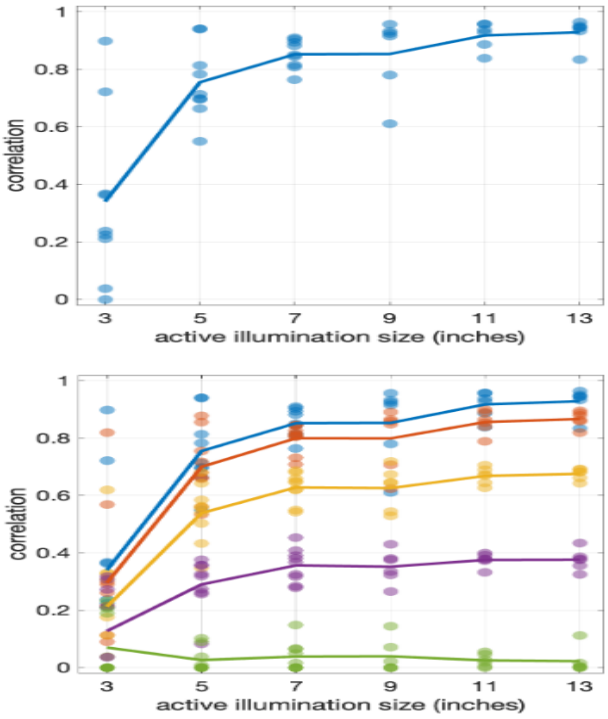

Figure 5. Correlation of Facial Hue Under Varying Illumination Sizes and Temporal Shifts

4. Real-Time Detection Methods

The growing need for immediate verification of visual authenticity has driven the development of real-time deepfake detection systems. Unlike offline forensic methods that analyze pre-recorded media, real-time approaches must operate with minimal latency and often rely on physically interpretable cues—such as light distribution, motion coherence, or temporal illumination consistency—to differentiate between genuine and synthesized faces. Among these, light-based detection techniques have emerged as a powerful and computationally efficient direction for ensuring authenticity during live video interactions.

Hui Guo, Xin Wang, and Siwei Lyu (2022) [52] introduced an active forensic method for real-time deepfake detection in video-conferencing environments based on corneal reflection analysis, enabling user authentication without requiring any specialized hardware. The method operates by briefly displaying a high-contrast geometric probing pattern—such as a diamond—on the participant's screen while the webcam captures the reflections forming on the user's eyes. Because real eyes naturally reflect the displayed pattern through physical light interaction, genuine participants exhibit clear and consistent corneal reflections, whereas deepfake avatars fail to do so due to the absence of real optical surfaces capable of producing physically accurate reflections. The overall pipeline of the proposed system is illustrated

in Figure 6. In a typical video-conference setting, a user sits in front of a laptop while the webcam records facial imagery (Fig. 6a). Upon initiating verification, the host displays the probing pattern on the shared screen. A real user will present a corresponding reflection in the cornea, while a real-time deepfake will not. The system first detects the face and extracts facial landmarks using Dlib [18] (Fig. 6b), then localizes the eye regions (Fig. 6c). Next, iris segmentation is performed using an edge detector followed by a Hough transform (Fig. 6d). The segmented iris images are passed to a template-matching module where the extracted reflection is compared with the original probing pattern (Fig. 6e). A match indicates a real person, while a lack of reflection—or very low similarity—suggests a possible real-time deepfake impersonation.

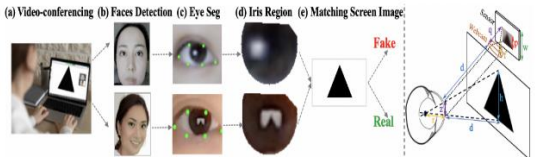

Figure 6. Overall Pipeline of the Proposed Method (Left) and Geometric Model for Estimating the Probing Pattern Size on the Camera Sensor (Right)

A key technical challenge is determining whether the probing pattern displayed on the screen forms a sufficiently large and detectable reflection on the cornea. The authors address this by modeling a standard video-conference scenario to estimate the expected pixel size of the reflection. Using an idealized geometric model, they assume a probing pattern of height $h \approx 14.5$ cm (70% of a 21.24-cm screen), a viewing distance of $d = 30\text{cm}$, an eyeball radius of $r = 1.25$ cm, a webcam sensor height of $w = 0.45$ cm, sensor resolution of $M = 720$ pixels, and a focal length of $f = 0.5\text{cm}$. Through geometric projection and imaging equations, the estimated size of the corneal reflection in the camera sensor is approximately:

$$\left(\frac{hrfM}{wd(d-f)}\right)^2 \approx 256 \text{ pixels}$$

However, directly comparing RGB values between the pattern and the reflection is unreliable due to ambient light variations and color-gamut differences between screen and camera. As shown in Figure 7, the authors therefore binarize the extracted corneal reflection and use high-contrast probing patterns (e.g., saturated shapes on a white background) to achieve robust shape preservation. Automatic thresholding is applied to generate binary masks, after which multi-scale templates are created to handle changes in head-to-screen distance.

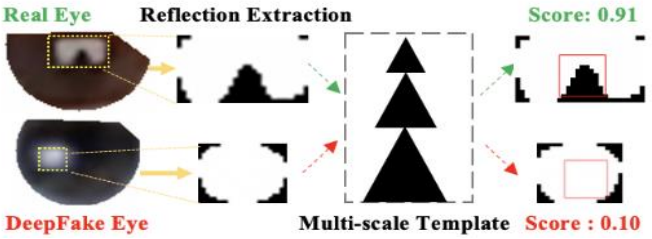

Figure 7. Overview of the probing pattern matching. The multi-scale templates are generated using the scaled probing pattern.

Template matching is performed using Normalized Cross-Correlation (NCC), with the matching response maps illustrated in Figure 8, where high NCC regions indicate successful localization of the reflected probing pattern. The NCC score is computed as:

$$NCC(u,v) = \frac{\sum_{x,y}\left[I(x,y)-\bar{I}_{u,v}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\sqrt{\sum_{x,y}\left[I(x,y)-\bar{I}_{u,v}\right]^2 \sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2}}$$

$I\,(x,y)$: pixel intensity of the input image

$t(x,y)$: pixel intensity of the template image

$\bar{I}$: mean intensity of the input image

$\bar{T}$: mean intensity of the template image

(x,y): Spatial coordinates of the pixels.

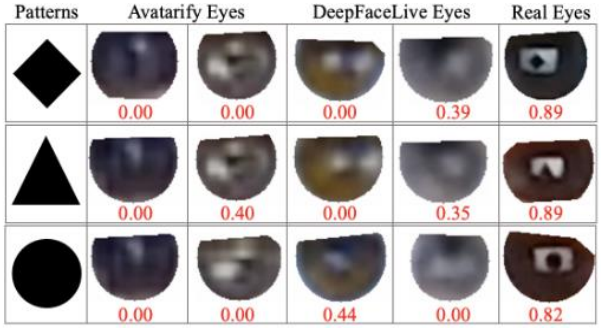

Figure 8. Probing Patterns (Left), Reflections in DeepFake Eyes Generated by Avatarify and DeepFaceLive (Middle), and Reflections in Real Eyes (Right)

To improve robustness against adversaries who might attempt to insert a predicted reflection into a deepfake stream, the authors recommend randomizing the probing pattern and embedding dynamic content such as timestamps. As further shown in Figure 9, color variations can significantly affect matching performance, highlighting the importance of binarization and contrast-based reflection extraction.

Figure 9. Effect of Probing Pattern Colour on Corneal Reflection and NCC Performance

This study proposes an active, illumination-based method for detecting real-time deepfake impersonation in video conferencing by analyzing corneal reflections. A high-contrast geometric probing pattern is briefly displayed on the user's screen, and genuine users produce clear reflections of this pattern in their eyes, while deepfakes fail to replicate these physical light interactions. Experiments using real Zoom recordings and simulated datasets show that real users yield high NCC correlation scores, whereas deepfake tools such as Avatarify and DeepFaceLive produce near-zero similarity. Additional evaluations demonstrate that the method is robust across different pattern shapes, colors, and indoor lighting conditions, with high-contrast designs providing the strongest reflections. Indoor light level changes have limited impact, as shown in Figure 10. Although the current implementation processes one frame every four seconds, optimization can enable real-time operation. Overall, the method offers a reliable, hardware-free biometric defense against real-time deepfake attacks.

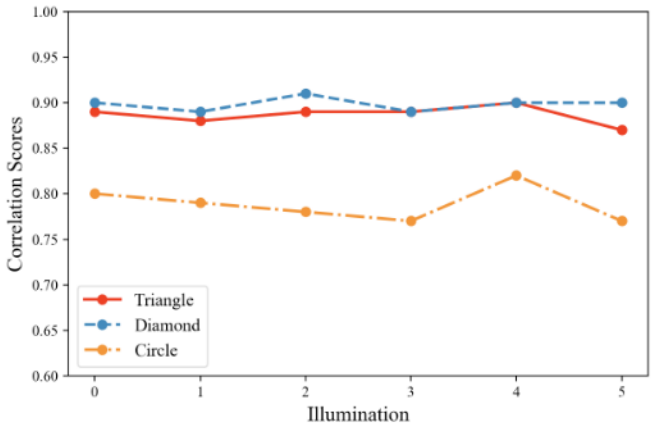

Figure 10. Influence of Indoor Illumination on Corneal Reflection Detection Performance

According to Zhixin Xie and Jun Luo (2024) [53], the SFake system works in three main steps, as illustrated in Figure 11. First, it generates a predefined or random vibration pattern and adjusts the camera's focal length while recording the face under controlled smartphone vibration. Next, it detects facial landmarks and analyzes gradient information to locate regions most responsive to the vibration pattern. Finally, it computes the variance of these regions across frames, filters noise, and determines authenticity by checking whether the extracted features correspond to the imposed vibration pattern.

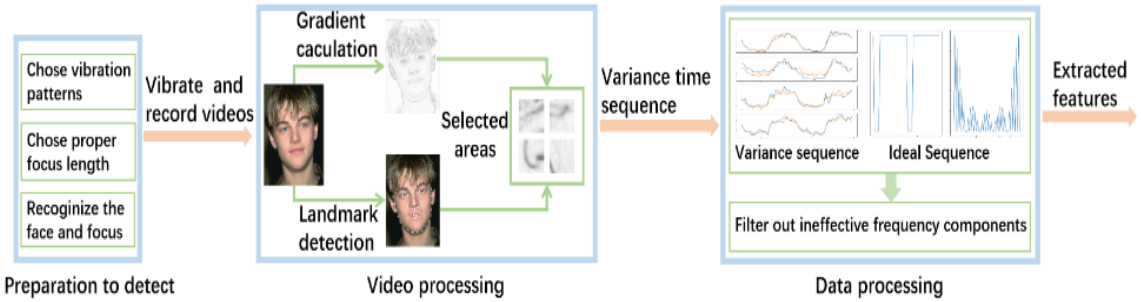

Figure 11. The workflow of the SFake

SFake configures its vibration patterns by adjusting the vibration period and duty cycle, expecting the resulting variance sequence to follow these patterns. Although the variance generally reflects the vibration behavior, its sensitivity is limited—for example, small duty-cycle changes (0.50 → 0.51) may not significantly affect the variance. To study this relationship, vibration periods from 1–5 seconds and duty cycles from 0–1 (step 0.05) are tested, and a video is recorded for each pattern. The variance sequence is computed frame by frame, and its middle value is used to estimate the duty cycle. Figure 12 (a) shows that the duty-cycle trends are mostly preserved but not fully accurate for short periods (1–2 seconds). Figure 12 (b) confirms that the variance period consistently matches the vibration period. Therefore, SFake adopts three coarse duty-cycle options—0.2, 0.5, and 0.8—which create clearly distinguishable blur levels, and does not restrict the period to maintain pattern diversity. Android's Vibrator and VibrationEffect interfaces enable generating these controllable vibration patterns. Because vibration-induced blur depends on proper camera focus, SFake first detects the face using Dlib and adjusts the camera to focus on that region. Increasing the focal length enlarges the Circle of Confusion and strengthens blur but narrows the field of view, so a practical value of about 50 mm is chosen to balance blur intensity and facial coverage. In Android, the focus distance is set via LensFocusDistance in CaptureRequest.Builder, and the facial area is selected using a MeteringRectangle before recording.

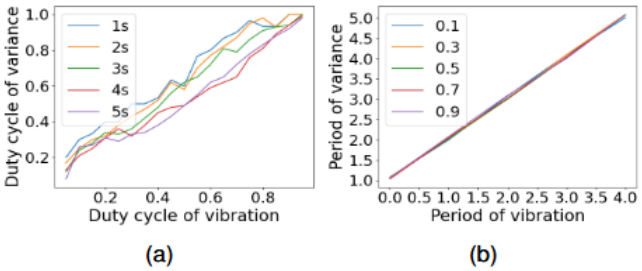

Figure 12. (a) The relationship between the duty cycle of variance and vibration across different periods. (b) The relationship between the period of variance and vibration across different duty cycles.

The fundamental idea of SFake is to determine video authenticity by comparing the vibration pattern with the corresponding changes in blurriness. To measure this blurriness accurately, SFake computes the image gradient but only in selected representative regions to reduce computational cost. Blurriness caused by vibration mainly appears along strong edges in the image. Although variance can be used to measure blur under good shooting conditions, poor lighting and sensor noise introduce fluctuations inside flat color regions, making variance unreliable. To reduce this noise, SFake applies gradient processing. The gradient at pixel location (i,j) is computed using a local 3×3 neighborhood as:

$$g[i,j] = \frac{1}{9} \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} |f[u,v] - f[i,j]|,$$

Where f[i,j] is the pixel value at position (i,j), and g[i,j] is the resulting gradient magnitude. Gradients smaller than one-tenth of the maximum gradient are set to zero to eliminate noise. As shown in Figure 13(a), the gradient image highlights meaningful edges, and Figure 13(b) shows that the variance sequence computed after gradient processing contains significantly less noise and matches the vibration pattern more clearly.

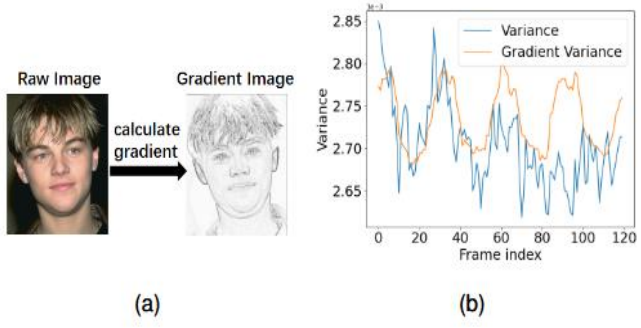

Figure 13. (a) The original image and the gradient image.

Because calculating gradients for the entire frame is expensive and may include irrelevant changes such as background motion or facial expressions, SFake selects

only informative areas. Since vibration-induced blur is more pronounced around facial features, SFake uses Dlib's 68 facial landmarks to locate potential regions (Figure 14(a)). Areas involving the eyes and eyebrows are excluded to avoid blinking interference. From the remaining landmark-based rectangles, SFake computes the average gradient value and selects the top nnn regions with the highest gradients for variance analysis. These selected regions are fixed throughout the entire video because the smartphone remains still during detection. To demonstrate the effectiveness of this selection method, a 4-second video is analyzed using both a random 50×50 patch and a patch chosen through the landmark-based approach. As illustrated in Figure 14(b), the area selected by the algorithm aligns with the vibration pattern much more accurately, confirming the advantage of the gradient-based landmark selection strategy.

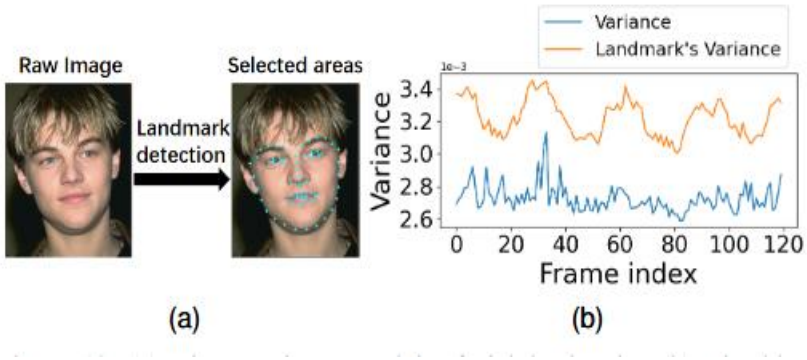

Figure 14. (a) The raw image and its facial landmarks. (b) The blue waveform is the variance sequence of the area randomly selected, and the orange waveform is that selected by landmark detection.

The variance sequence of the selected region usually reflects the vibration pattern, but it can be distorted by factors such as lighting changes or slight movements. In a 4-second indoor test video with gradual illumination variation, the raw variance sequence (blue curve in Figure 15(a)) fails to show the expected 1-second, 0.5-duty-cycle pattern. To isolate the vibration component, the ideal square-wave variance sequence is transformed into the frequency domain, and the top 80% of its non-zero frequency components are identified. The actual variance sequence is then filtered to retain only these frequencies. The filtered result (orange curve in Figure 15(a)) clearly matches the vibration period of 30 frames and a 0.5 duty cycle, and is used as a 120-dimensional feature. For comparison, the same process applied to a generated fake video produces neither a meaningful raw nor filtered pattern, as shown in Figure 15(b), demonstrating that the extracted features reliably distinguish real from fake videos.

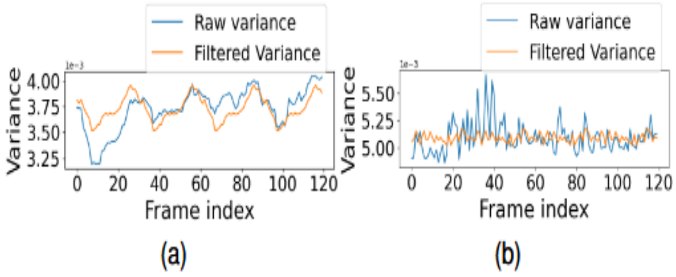

Figure 15. (a) The raw and filtered variance sequence of the real video. (b) The raw and filtered variance sequence of the fake video.

SFake shows strong performance across all evaluated deepfake sources, consistently achieving detection accuracies above 95%. It performs best on DeepFaceLive, reaching nearly 99% accuracy, since its classifier was trained on this type of fake data. Even in challenging cases such as RemakerAI—where other methods struggle due to compression artifacts—SFake still maintains high accuracy. To understand why SFake performs well, Figure 16 compares the variance sequences of real and fake videos under the same vibration pattern. The real video displays consistent drops every 30 frames, matching the imposed 1-second, 0.5-duty-cycle vibration. Fake videos, however, show irregular fluctuations with no periodic pattern, indicating that deepfake generators cannot reproduce the fine-grained blurriness changes caused by physical vibration. This clear separation makes classification straightforward.

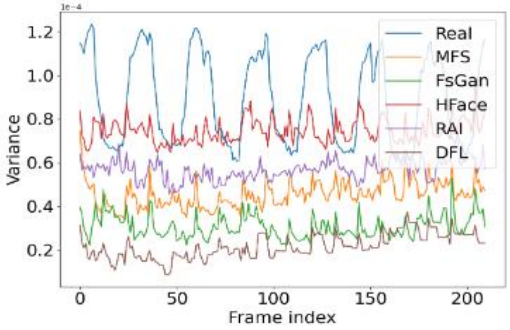

Figure 16. We extract the same 50x50 area from one real video and its corresponding five fake videos to calculate the variance sequences.

In terms of efficiency, SFake is significantly lighter than existing deepfake detectors. As shown in Figure 17, SFake requires less than 450 MB of memory—far lower than other methods—and processes a 4-second clip in about 4.5 seconds. Most of the computation comes from gradient extraction, but parallel processing keeps the runtime low.

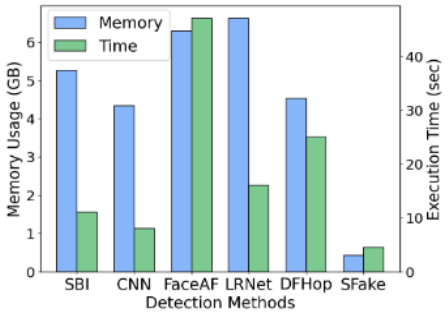

Figure 17. The computation performance of the different detection methods.

SFake also remains robust under various real-world conditions. Figure 18(a) shows that it performs well even under relatively low lighting, with accuracy dropping only in very dark environments. Figure 18(b) indicates that high resolution (1920×1080) is important, as lower resolutions suppress the subtle blur caused by vibration. According to Figure 18(c), the optimal shooting distance is between 20 and 60 cm, matching typical smartphone use. Figure 18(d) demonstrates that both 4-second and 8-second detection windows maintain high accuracy, while other durations require retraining. Finally, Figure 18(e) shows that a zoom factor between 1.6 and 2 yields the best results, providing enough magnification for vibration-induced blur without overwhelming the image

.

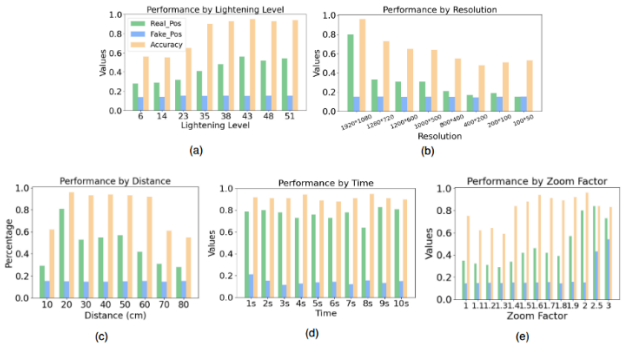

Figure 18. The impact of (a) lightening level, (b) detection time, (c) zoom factor, (d) resolution, and (e) shooting distance on accuracy and POS values for real and fake videos.

Table 1. Comparative Analysis of Real-Time Deepfake Detection Methods

Algorithm & Study	Objective	Dataset	Processing Method	Results
Gerstner et al. (2022) – *Active Illumination via Hue Modulation*	To verify facial authenticity by measuring whether the skin's chromatic response follows a controlled, time-varying illumination pattern.	• Real-world recordings from 15 participants under varying lighting, distances, and display sizes. • Physically based simulations using the Mitsuba renderer with multiple skin tones and illumination parameters.	•Display sinusoidally modulated hue • Face masking with Dlib landmarks • HSV conversion & circular hue averaging • Compare measured vs expected hue using circular correlation	• Simulation correlation ≈ 0.99. • Real-world correlation ≈ 0.93 with large illumination sources. • Deepfake videos produce correlation ≈ 0.09. • Method achieves clear separation between real and synthetic content.
Guo, Wang & Lyu (2022) – *Corneal Reflection Pattern Matching*	To detect deepfake impersonation by analyzing the presence, shape, and temporal consistency of corneal reflections generated by an on-screen probing pattern.	• Real video-conference recordings (Zoom). • Fake videos produced using Avatarify and DeepFaceLive. • Multiple indoor lighting scenarios, pattern shapes, and colours.	• Display high-contrast probing pattern • Detect eyes using Dlib • Iris segmentation + Hough transform • Template matching using NCC	• Real users show high NCC similarity. • Deepfakes show near-zero NCC similarity. • Robust to lighting and pattern variations. • Provides reliable real-time identity verification without special hardware.
Xie & Luo (2024) – SFake (Vibration-Induced Blurriness)	To authenticate videos by examining whether the temporal blur pattern matches a predefined smartphone vibration signal.	• Real smartphone vibration recordings. • Fake videos generated by DeepFaceLive, RemakerAI, and additional deepfake tools. • Tests across lighting, distance, resolution, and zoom conditions.	• Generate vibration pattern • Landmark-based region selection • Gradient & variance calculation • FFT filtering & 2-layer NN classifier	• Overall detection accuracy >95%. • DeepFaceLive accuracy: 98.8%. • RemakerAI accuracy: 96.8%. • Very lightweight: <450 MB RAM, ~4.5 seconds processing. • Real videos show periodic variance drops; fake videos show random noise patterns.

5. Challenges and Limitations

Although deepfake detection—particularly through deep learning and active-illumination methods—has achieved notable progress, several fundamental challenges continue to limit the reliability, generalizability, and deployment of these systems in real-world environments. These limitations can be grouped into five key areas:

5.1 Dataset and Training Limitations

Deepfake detection remains heavily dependent on supervised learning, yet the field suffers from a significant shortage of diverse, high-quality, and illumination-rich datasets. As reported in Zotov et al. (2020) [54] and Mitra et al. (2021) [55], most existing datasets exhibit:

- Imbalanced distributions of real vs. fake samples
- Limited representation of environmental variability (lighting, distance, motion)
- Unrealistic or low-quality synthetic faces
- Weak coverage of modern deepfake generation pipelines

Many benchmark datasets (e.g., DFDC, FaceForensics++) lack true illumination diversity, causing detectors to overfit to specific lighting conditions or compression levels. Additionally, training deep learning detectors on large-scale video datasets requires substantial computational power, which becomes a bottleneck for real-time systems.

5.2 Reliability and Generalization Issues

Despite strong performance in controlled environments, many detectors perform poorly on unseen manipulations, new generation techniques, or real-world media. Studies such as Xu et al. (2022) and Zhang (2022) [56][57] highlight key weaknesses:

- Low robustness to domain shifts (different codecs, cameras, color spaces)
- Vulnerability to adversarial attacks and subtle perturbations
- Poor performance on hybrid manipulations or partial forgeries
- Difficulty handling multi-face and fast-motion scenarios

These issues reveal that current binary classification frameworks are insufficient for complex, dynamic, or adaptive deepfake threats.

5.3 Lack of Explainability and Transparency

Most detection models function as opaque black-boxes, providing confidence scores without interpretable evidence. This presents major problems in high-stakes environments such as journalism, forensics, and legal investigations, where explainability is essential. The lack of:

- traceable decision rationale
- interpretable visual cues
- physically grounded verification signals

reduces user trust and limits the adoption of automated detectors in practical authentication workflows.

5.4 Real-World Operational Constraints

Methods relying on active illumination, corneal reflections, or fine-grained photometric cues require stable hardware, precise synchronization, and high-quality

imaging. In practice, consumer devices introduce several constraints:

- limited dynamic range, noisy sensors, and inconsistent frame rates
- autofocusing issues and motion blur
- unpredictable indoor/outdoor lighting environments
- occlusions from glasses, hair, or head movement

Furthermore, video streaming platforms apply heavy compression, downsampling, and metadata removal ("social media laundering"), which deteriorates the subtle signals required by photometric detectors.

5.5 Evaluation and Benchmarking Limitations

Most studies evaluate deepfake detection as a simple binary classification task. This fails to capture real-world complexities where:

- only specific regions or frames may be manipulated
- temporal consistency is critical
- localized authenticity analysis is required

Additionally, there is no standardized benchmark for evaluating real-time performance, such as latency, throughput (fps), or processing delay. The lack of standardized evaluation metrics makes it difficult to compare methods or assess whether proposed solutions are viable for real-time deployment.

6. Conclusion

This review highlights that real-time deepfake detection methods based on illumination cues—such as active lighting, corneal reflections, and vibration-induced blur—are highly effective because deepfake models cannot accurately mimic real physical light interactions. These techniques provide strong temporal consistency, interpretability, and reliability compared to traditional data-driven approaches. However, their performance is still affected by varying lighting conditions, hardware limitations, compression artifacts, and the lack of illumination-focused datasets. Future work should combine physical light modeling with advanced AI methods, such as YOLO and transformer-based detectors, to improve robustness and enable scalable, real-time deepfake authentication. Overall, integrating illumination-based cues with modern machine learning offers a promising direction for building reliable and explainable deepfake detection systems.

7. Future Directions

Future work will focus on creating a hybrid deepfake detection framework that combines physical illumination cues with advanced AI models. Research should improve data pre-processing through adaptive filtering, illumination balancing, and temporal alignment to handle challenging real-world conditions. Expanding datasets to include diverse lighting environments, facial angles, backgrounds, and motion variations will enhance model generalization. Additionally, integrating powerful deep learning architectures—such as YOLO, CNN–transformer hybrids, and feature-fusion

networks—will enable more robust, real-time, and accurate detection of both known and emerging deepfakes. Overall, merging physical-based analysis with data-driven methods represents a promising direction for future deepfake authentication systems.

REFERENCES

[1] Çeçen M, Karaköse M. A Deepfake Image Detection Approach Based on YOLOv3. In: 2th International Conference on Advances and Innovations in Engineering; 21-23 September 2023. pp. 10-18.

[2] Franklin RJ, Mohona. Traffic Signal Violation Detection using Artificial Intelligence and Deep Learning. In: International Conference on Communication and Electronics Systems; 10-12 June 2020. pp. 839 - 844.

[3] İlhan İ., Balı E., Karaköse M. An Improved DeepFake Detection Approach with NASNetLarge CNN. In: IEEE International Conference on Data Analytics for Business and Industry; 25-26 October 2022. pp. 598-602.

[4] Seow JW, Lim MK, Phan R, Liu J. A comprehensive overview of Deepfake: Generation, detection, datasets, and opportunities. Elsevier Neurocomputing 2022; 513: 351–371.

[5] İlhan İ, Karaköse M. Derin Sahte Videoların Tespiti ve Uygulamaları için Bir Karşılaştırma Çalışması. Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi 2021; 8(14): 47-60.

[6] John J, Sherif B. Comparative Analysis on Different DeepFakeDetection Methods and Semi Supervised GAN Architecture for DeepFake Detection. In: Proceedings of the Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud); 10-12 November 2022.

[7] *FaceApp*. Accessed: Jan. 4, 2021. [Online]. Available: https://www. faceapp.com/

[8] *FakeApp*. Accessed: Jan. 4, 2021. [Online]. Available: https://www. fakeapp.org/

[9] Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, J. Ortega-Garcia, Deepfakes and beyond: a survey of face manipulation and fake detection. Inf. Fusion [Online] 64, 131 148 (2020). Available at: https://arxiv.org/pdf/2001.00179.pdf.

[10] Clark, Deepfakes algorithm nails Donald Trump in most convincing fake yet [Online]. TNW | Artificial-Intelligence (2018).

[11] Westerlund, The emergence of Deepfake technology: a review. Technol. Innov. Manag. Rev. [Online] 9(11) (2019).

[12] N. Kanwal, A. Girdhar, L. Kaur, and J. S. Bhullar, "Detection of Digital Image Forgery using Fast Fourier Transform and Local Features," in 2019 International Conference on Automation, Computational and Technology Management (ICACTM), London, United Kingdom: IEEE, Apr. 2019, pp. 262–267. doi: 10.1109/ICACTM.2019.8776709.

[13] L. Verdoliva, "Media Forensics and DeepFakes: An Overview," IEEE J. Sel. Top. Signal Process., vol. 14, no. 5, pp. 910–932, Aug. 2020, doi: 10.1109/JSTSP.2020.3002101.

[14] "A. van den Oord, Y. Li, O. Vinyals, Representation... - Google Scholar." Accessed: Jun. 07, 2024.

[15] GoodfellowIan et al., "Generative adversarial networks," Communications of the ACM, Oct. 2020, doi: 10.1145/3422622.

[16] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders," MAL, vol. 12, no. 4, pp. 307–392, Nov. 2019, doi: 10.1561/2200000056.

[17] "Jianchang Mao and Anil K Jain. Texture classification... - Google Scholar." Accessed: Jun. 07, 2024.

[18] Azawi, Raghad Majeed, Ibrahim Tariq Ibrahim, and Israa Mishkhal. "A Hybrid Detection System of Heart Disease by Using Machine Learning Techniques." Journal homepage: https://ijas. uodiyala. edu. iq/index. php/IJAS/index ISSN 3006: 5828.

[19] "Adversarial-learning-based image-to-image transformation: A survey - ScienceDirect." Accessed: Jun. 07, 2024.

[20] J. W. Seow, M. K. Lim, R. C. W. Phan, and J. K. Liu, "A comprehensive overview of Deepfake: Generation, detection, datasets, and opportunities," Neurocomputing, vol. 513, pp. 351–371, Nov. 2022, doi: 10.1016/j.neucom.2022.09.135.

[21] "FaceApp Inc, Faceapp (2016). https://www.faceapp.com/. - Google Scholar." Accessed: Jun. 08, 2024.

[22] shaoanlu, shaoanlu/faceswap-GAN. (Mar. 09, 2024). Jupyter Notebook. Accessed: Mar. 11, 2024.

[23] "Zao Asian Cafe," App Store. Accessed: Mar. 11, 2024.

[24] J. He, J. Zheng, Y. Shen, Y. Guo, and H. Zhou, "Facial Image Synthesis and Super-Resolution

With Stacked Generative Adversarial Network," Neurocomputing, vol. 402, pp. 359–365, Aug. 2020, doi: 10.1016/j.neucom.2020.03.107.

[25] R. Natsume, T. Yatagawa, and S. Morishima, "RSGAN: Face Swapping and Editing using Face and Hair Representation in Latent Spaces," in ACM SIGGRAPH 2018 Posters, Aug. 2018, pp. 1–2. doi: 10.1145/3230744.3230818.

[26] R. Natsume, T. Yatagawa, and S. Morishima, "FSNet: An Identity-Aware Generative Model for Image-based Face Swapping," vol. 11366, 2019, pp. 117–132. doi: 10.1007/978-3-030-20876-9_8.

[27] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject Agnostic Face Swapping and Reenactment," Aug. 16, 2019, arXiv: arXiv:1908.05932. doi: 10.48550/arXiv.1908.05932.

[28] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping," Sep. 15, 2020, arXiv: arXiv:1912.13457. Accessed: Dec. 10, 2023. [Online]. Available: http://arxiv.org/abs/1912.13457.

[29] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT: IEEE, Jun. 2018, pp. 8789–8797. doi: 10.1109/CVPR.2018.00916.

[30] W. Wu, Y. Zhang, C. Li, C. Qian, and C. C. Loy, "ReenactGAN: Learning to Reenact Faces via Boundary Transfer," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 603–619. Accessed: Jun. 10, 2024.

[31] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-GAN: Unsupervised Video Retargeting," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 119–135. Accessed: Jun. 10, 2024.

[32] Y. Song, J. Zhu, D. Li, X. Wang, and H. Qi, "Talking Face Generation by Conditional Recurrent Adversarial Network," Jul. 25, 2019, arXiv: arXiv:1804.04786. doi: 10.48550/arXiv.1804.04786.

[33] S. Tripathy, J. Kannala, and E. Rahtu, "ICface: Interpretable and Controllable Face Reenactment Using GANs," in 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA: IEEE, Mar. 2020, pp. 3374– 3383. doi: 10.1109/WACV45572.2020.9093474.

[34] Y. Sun, J. Tang, Z. Sun, and M. Tistarelli, "Facial Age and Expression Synthesis Using Ordinal Ranking Adversarial Networks," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2960–2972, 2020, doi: 10.1109/TIFS.2020.2980792.

[35] Y. Wang, P. Bilinski, F. Bremond, and A. Dantcheva, "ImaGINator: Conditional Spatio-Temporal GAN for Video Generation," in 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA: IEEE, Mar. 2020, pp. 1149–1158. doi: 10.1109/WACV45572.2020.9093492.

[36] C. Fu, Y. Hu, X. Wu, G. Wang, Q. Zhang, and R. He, "High-Fidelity Face Manipulation With Extreme Poses and Expressions," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2218–2231, 2021, doi: 10.1109/TIFS.2021.3050065.

[37] Y. Didi, "Jiggy: Magic dance gif maker (2020)," URL https://apps. apple. com/us/app/jiggy-magic-dance-gifmaker/id1482608709.

[38] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, "Liquid Warping GAN: A Unified Framework for Human Motion Imitation, Appearance Transfer and Novel View Synthesis," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5904–5913. Accessed: Jun. 13, 2024.

[39] T. Xiao, J. Hong, and J. Ma, "ELEGANT: Exchanging Latent Encodings with GAN for Transferring Multiple Face Attributes," presented at the Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 168–184. Accessed: Jun. 13, 2024.

[40] T. Li et al., "BeautyGAN: Instance-level Facial Makeup Transfer with Deep Generative Adversarial Network," in Proceedings of the 26th ACM international conference on Multimedia, in MM '18. New York, NY, USA: Association for Computing Machinery, Oct. 2018, pp. 645–653. doi: 10.1145/3240508.3240618.

[41] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: Facial Attribute Editing by Only Changing What You Want," IEEE Transactions on Image Processing, vol. 28, no. 11, pp. 5464–

5478, Nov. 2019, doi: 10.1109/TIP.2019.2916751.

[42] M. Liu et al., "STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3673–3682. Accessed: Jun. 14, 2024.

[43] Y. Jo and J. Park, "SC-FEGAN: Face Editing Generative Adversarial Network With User"s Sketch and Color," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1745–1753. Accessed: Jun. 15, 2024.

[44] X. Nie, H. Ding, M. Qi, Y. Wang, and E. K. Wong, "URCA-GAN: UpSample Residual Channel-wise Attention Generative Adversarial Network for image-to-image translation," Neurocomputing, vol. 443, pp. 75–84, Jul. 2021, doi: 10.1016/j.neucom.2021.02.054.

[45] "Photo-realistic face age progression/regression using a single generative adversarial network - ScienceDirect." Accessed: Jun. 14, 2024.

[46] J. Guo and Y. Liu, "Attributes guided facial image completion," Neurocomputing, vol. 392, pp. 60–69, Jun. 2020, doi: 10.1016/j.neucom.2020.02.013.

[47] M. Afifi, M. A. Brubaker, and M. S. Brown, "HistoGAN: Controlling Colors of GAN-Generated and Real Images via Color Histograms," presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7941–7950. Accessed: Jun. 15, 2024.

[48] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4401–4410. Accessed: Jun. 05, 2024.

[49] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, Jun. 2020, pp. 8107–8116. doi: 10.1109/CVPR42600.2020.00813.

[50] C. R. Gerstner and H. Farid, Detecting Real-Time Deep-Fake Videos Using Active Illumination, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR), 2022, pp. 3928–3937.

[51] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wen-zel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. Transactions on Graphics (Proceedings of SIG GRAPH Asia), 38(6), Dec. 2019.

[52] H. Guo, X. Wang, and S. Lyu, Detection of Real-Time DeepFakes in Video Conferencing with Active Probing and Corneal Reflection, arXiv preprint arXiv:2210.12108, 2022.

[53] Z. Xie and J. Luo, SFake: Real-Time Deepfake Detection via Smartphone Vibration Probing, arXiv preprint arXiv:2404.01674, 2024.

[54] Zotov S, Dremliuga R, Borshevnikov A et al (2020) Deepfake detection algorithms: a meta-analysis. In: 2020 2nd symposium on signal processing systems. pp 43–48.

[55] Mitra A, Mohanty SP, Corcoran P et al (2021) A machine learning based approach for deepfake detection in social media through key video frame extraction. SN Comput Sci 2(2):1–18.

[56] Xu FJ, Wang R, Huang Y et al (2022) Countering malicious deepfakes: survey, battleground, and horizon. Int J Comput Vis. https:// doi. org/ 10. 1007/ s11263- 022- 01606-8.

[57] Zhang T (2022) Deepfake generation and detection, a survey. Multimed Tools Appl 81(5):6259–6276.