A Robust Deep Learning Framework for Face Anti-Spoofing Detection Using Compact Convolutional Neural Networks

Pandiripalli Sai Ganesh[1], Palleti Sasidhar[2], Kunchakuri Dheeraj[3], Kusuri Karthik[4]
[1,2,3,4]*Department of CSE KL University, Vaddeswaram, Guntur, Andhra Pradesh, India*

***Abstract*—Face recognition systems have become ubiquitous in our daily lives—from unlocking smartphones with a glance to passing through automated border control gates at international airports. However, this convenience comes with a dangerous vulnerability: presentation attacks, commonly known as spoofing. Attackers can bypass these systems using something as simple as a printed photograph or as sophisticated as a hyper-realistic 3D silicone mask or a deepfake video generated by artificial intelligence. This paper presents a comprehensive study and a practical solution for face anti-spoofing using a compact Convolutional Neural Network (CNN) designed specifically for resource-constrained environments like smartphones and edge devices. We trained and rigorously evaluated our model on two benchmark datasets: the CASIA Face Anti-Spoofing Database (CASIA-FASD) and the Replay-Attack Database. On the CASIA dataset, our model achieves a training accuracy of 93.37% and a validation accuracy of 91.20%. More importantly, it demonstrates a Bona Fide Presentation Classification Error Rate (BPCER) of just 1.62%, meaning that genuine users are almost never incorrectly rejected. However, our Attack Presentation Classification Error Rate (APCER) of 27.54% reveals that sophisticated attacks—particularly high-quality video replays and 3D masks—remain challenging. Our model contains only 405,377 parameters, making it lightweight enough to run in real-time on devices with limited computational resources. Beyond the technical results, this paper offers an honest, human-centered discussion of the real-world challenges that are often overlooked in academic literature: why a liveness system trained on English lip movements fails for a Tamil or Mandarin speaker, why iris recognition is impractical on most smartphones due to resolution limits, and why asking a person with facial paralysis to" smile for the camera" is not just annoying but exclusionary. We also evaluate our model on the Replay-Attack dataset to test cross-database generalization, revealing significant performance degradation (accuracy dropping from 91.20% to 67.5%), highlighting the critical challenge of domain shift in real-world deployment. We conclude with a forward-looking roadmap that includes multimodal fusion (combining visible light with thermal and depth sensors), few-shot learning for adapting to new attack types with minimal examples, and privacy-preserving architectures that never transmit raw face images off the user's device.**

I. INTRODUCTION

A. The Silent Revolution of Facial Recognition

Imagine waking up in the morning, picking up your smart-phone, and having it unlock instantly just by looking at it. No password to type, no fingerprint to press, no pattern to draw. You walk to your front door, and it opens automatically because it recognizes your face. You board a flight at Indira Gandhi International Airport in Delhi, and you walk through a biometric e-gate that verifies your identity in less than five seconds without showing your boarding pass or passport. This is not science fiction—this is the reality of 2026.

Facial recognition technology has quietly integrated itself into the fabric of modern life. According to a 2025 report by Markets and Markets, the global facial recognition market was valued at approximately $8.5 billion in 2024 and is projected to reach $15.7 billion by 2030, growing at a compound annual growth rate (CAGR) of 14.8%. In India alone, the Unique Identification Authority of India (UIDAI) has enrolled over 1.4 billion people with biometric data including face, fingerprint, and iris scans under the Aadhaar program. The Delhi airport's DigiYatra system processes over 60,000 passengers daily using facial recognition, reducing average boarding time from 45 seconds to just 5 seconds per passenger.

Banks are increasingly adopting facial recognition for high-value transactions. In 2025, the State Bank of

India reported that over 40 million customers had enabled facial authentication for mobile banking transactions. Smartphone manufacturers—Apple, Samsung, Google, OnePlus, Xiaomi—have all integrated face unlock features into their flagship devices. Even your local supermarket might use facial recognition to identify loyal customers and offer personalized discounts as you walk in. The technology has truly become invisible and ubiquitous. And that is precisely the problem.

B. The Hidden Vulnerability: How Attackers Fool the System

Every rose has its thorns. Despite its sophistication and convenience, facial recognition has a fundamental weakness: a camera cannot inherently distinguish between a real, live human face and a clever imitation. This vulnerability has given rise to a thriving underground industry of spoofing attacks. Let us walk through the most common attack vectors, from the simple to the terrifyingly sophisticated.

1) The Printed Photo Attack: Simplicity Itself:

The simplest attack requires nothing more than a printer, some paper, and a photograph of the target. An attacker can find a high-resolution photo of the victim on social media Facebook, Instagram, LinkedIn print it out, and hold it in front of the camera. Older facial recognition systems, especially those deployed before 2018, are often completely fooled by this attack. The camera sees a face that matches the enrolled template, and the system grants access.

In 2019, researchers at the University of Toronto tested 12 commercial smartphones with face unlock features. They found that 9 out of 12 could be unlocked using a printed photograph. Even today, budget smartphones under 10,000 often use basic 2D face detection that remains vulnerable to print attacks.

2) The Video Replay Attack: Adding Motion:

Attackers quickly realized that static photos could be defeated by asking the user to blink or move their head. So, they evolved. In a video replay attack, the attacker records a short video of the legitimate user perhaps from a video call, a YouTube video, or a hidden camera. This video might include the user blinking, smiling, turning their head, or even speaking. The attacker then plays this video on a smartphone or tablet screen and presents it to the authentication camera.

Modern displays have made this attack even more effective. OLED screens with 120 Hz or 240 Hz refresh rates eliminate the flicker that older detection methods relied on. High-brightness displays (1000+ nits) can match real-world lighting conditions. Some attackers have even been known to use curved displays to better simulate the 3D structure of a face.

3) The 3D Mask Attack: The Next Level:

This is where things get genuinely concerning. Using a 3D printer and silicone casting materials, an attacker can create a wearable mask that looks almost identical to the target person. The process typically involves:

1) Obtaining multiple high-resolution photos of the target from different angles.
2) Using photogrammetry software to reconstruct a 3D model of the target's face.
3) 3D printing a mold of the face using resin or PLA filament.
4) Casting the mask using medical-grade silicone that mimics skin texture, flexibility, and even subsurface scattering.
5) Adding details like painted eyes, implanted hairs, and cosmetic makeup.

The cost for a high-quality 3D mask can range from $500 to $5,000, but for a determined attacker targeting a high-value account or facility, this is a trivial investment. In 2023, a cybersecurity firm demonstrated that a $2,000 silicone mask could fool 80% of commercial face recognition systems.

4) The Deepfake Attack: The Digital Nightmare:

The newest and most alarming threat is entirely digital. Deep-fake technology, powered by Generative Adversarial Networks (GANs) and more recently by diffusion models like Stable Diffusion and Mid journey, can generate photorealistic artificial faces or manipulate existing videos to alter identity and expressions.

Consider this scenario: An attacker obtains a single photograph of the target from social media. Using a deepfake generation tool like StyleGAN3 or Diff AE, they can generate a short video of that person speaking, blinking, and turning their head. The video never existed in reality, but it looks completely

authentic. This synthetic video is then replayed to the face recognition system. The system sees a face that matches the enrolled template and observes natural motion. It grants access.

In 2024, a study by the University of California, Berkeley tested state-of-the-art face recognition systems against deep-fake videos. They found that 65% of deepfakes successfully evaded detection. Even more concerning, when the deepfake included the person's actual voice (synthesized from a few seconds of audio), the success rate rose to 82%.

C. Why Liveness Detection Matters

This is where liveness detection enters the picture as the essential defense mechanism. Liveness detection is not about recognizing *who* the person is—that is the job of face recognition. Liveness detection asks a different, more fundamental question:" Is *this a live, physically present human being, or is it a clever imitation?"*

Liveness detection methods fall into two broad categories:

1) *Active Liveness Detection:* Active methods require the user to perform a specific action on command.

The system might ask you to:

- Blink twice in rapid succession
- Smile naturally
- Turn your head to the left and then to the right
- Read aloud a randomly generated sequence of digits
- Follow a moving dot on the screen with your eyes

The system verifies that the action is performed correctly and naturally. Active methods are highly effective against static prints and simple video replays. However, they have significant drawbacks: they are annoying (no one likes being asked to perform parlor tricks to access their own phone), they are slow (adding 2-5 seconds to every authentication), and they are exclusionary (people with facial paralysis cannot smile, people with motor disorders cannot turn their head precisely, people with visual impairments cannot follow a moving dot).

2) Passive Liveness Detection:

Passive methods require no user cooperation whatsoever. The system simply analyzes the single image or short video stream for subtle cues that indicate liveness:

- Skin texture analysis: Real skin has pores, fine hairs, and subtle color variations that are extremely difficult to reproduce in prints or screens.
- Specular reflection analysis: The way light reflects off the skin, especially around the nose and forehead, follows physical laws that printed or screen-reflected light does not.
- Micro-movement analysis: Even when a person tries to stay perfectly still, there are microscopic movements—tiny sways, breathing, pulse-induced vibrations—that are detectable.
- Remote Photoplethysmography (rPPG): The human heart pumps blood with each beat, causing subtle color changes in the face (the face becomes slightly redder during systole and slightly paler during diastole). These changes can be extracted from a standard video camera and used to measure pulse. A print or mask has no pulse.

Passive methods are more user-friendly and accessible. They work in the background, invisible to the user. However, they are technically more challenging to implement and can be fooled by sufficiently sophisticated spoofs.

D. Problem Formulation

We formalize the face anti-spoofing problem as a binary classification task. Given an input face image $\boldsymbol{I} \in \mathrm{R}^{H\times W\times 3}$ (height H, width W, three color channels), we seek a function f_θ: $\mathrm{R}^{H\times W\times 3} \to [0, 1]$ parameterized by θ that estimates the probability that the image depicts a live human face. The decision rule with threshold τ (typically set to 0.5) is:

$$\text{Decision}(I) = \begin{cases} \text{Live (Bona Fide)} & \text{if } f_\theta(I) \geq \tau \\ \text{Spoof (Attack)} & \text{otherwise} \end{cases}$$

The optimal classifier minimizes the trade-off between two error types:

- False Acceptance (Type I Error):

Classifying a spoof attack as live. This is measured by the Attack Presentation Classification Error Rate (APCER).

- False Rejection (Type II Error):

Classifying a live user as a spoof. This is measured by the Bona Fide Presentation Classification Error Rate (BPCER).

In security-critical applications, we typically want

a very low APCER (we cannot afford to let attackers in), even if that means accepting a slightly higher BPCER (occasionally inconveniencing legitimate users). In user-experience-critical applications like smartphone unlock, we want a very low BPCER (users hate being locked out of their own phones), even if that means a slightly higher APCER.

E. Our Contributions

This paper presents the following contributions:

1) A compact CNN architecture for face anti-spoofing with only 405,377 parameters, making it suitable for real-time inference on resource-constrained edge devices.
2) Comprehensive evaluation on two benchmark datasets: CASIA-FASD (12,000 images) and Replay-Attack (1,300 videos), following ISO/IEC 30107-3 standardized metrics.
3) Cross-database generalization analysis, revealing that a model trained on CASIA-FASD achieves only 67.5% accuracy on Replay-Attack, highlighting the domain shift problem.
4) Human-centered discussion of real-world challenges of-ten ignored in academic literature: language dependency in lip-sync detection, resolution constraints for iris methods, and accessibility barriers for users with disabilities.
5) A practical roadmap for future work including multi-modal fusion, few-shot learning for novel attack adaptation, and privacy-preserving architectures.

F. Paper Organization

The remainder of this paper is structured as follows. Section II provides a comprehensive review of related work, presented in a way that is accessible to readers new to the field while still valuable to experts. Section III details our methodology, including the datasets (CASIA-FASD and Replay-Attack), preprocessing pipeline, network architecture, and training protocol. Section IV presents our experimental results, including per-dataset performance, cross-database generalization, and ablation studies. Section V offers an honest discussion of limitations and research gaps that are often glossed over. Section VI proposes future research directions that we believe are most promising. Section VII concludes the paper with key takeaways for researchers and practitioners.

II. RELATED WORK: LEARNING FROM THE PAST TWO DECADES

The field of face anti-spoofing has evolved significantly over the past twenty years. Understanding this evolution is essential for appreciating where we stand today and where we need to go.

A. The Early Days: Handcrafted Features (2004-2014)

In the early 2000s, researchers noticed something interesting: printed photographs and screen replays looked different from real faces under microscopic analysis. Real skin has a certain texture—pores, fine hairs, subtle color variations—that is extremely difficult to reproduce perfectly, especially with consumer-grade printers and displays. This observation led to the first generation of anti-spoofing methods based on handcrafted texture features.

1) Local Binary Patterns (LBP) and Its Variants:

The Local Binary Pattern (LBP) operator, originally developed for texture classification in the 1990s, became the workhorse of early anti-spoofing systems. LBP works by comparing each pixel to its eight neighbors, generating a binary code for each pixel. The histogram of these codes across the face image provides a texture signature that differs between live faces and prints.

Researchers developed several variants:

- Multi-scale LBP: Applies LBP at multiple scales to capture textures at different resolutions.
- LBP from Three Orthogonal Planes (LBP-TOP): Ex-tends LBP to video by considering the spatial planes (XY, XT, YT), capturing both spatial texture and temporal motion.
- Volume LBP (VLBP): A more general extension that considers a full 3D spatiotemporal volume.

Boulkenafet et al. [1] showed that analyzing color textures across multiple color spaces (RGB, HSV, YCbCr) significantly improved detection accuracy. Their approach achieved an average classification error rate of around 15% on the CASIA dataset, which was state-of-the-art at the time.

2) Histogram of Oriented Gradients (HOG):

Originally developed for pedestrian detection, the

Histogram of Oriented Gradients (HOG) descriptor was adapted for anti-spoofing. HOG computes the distribution of gradient orientations (edge directions) in local regions of the image. The intuition is that printed photos have different edge characteristics than real faces—the boundaries between the photo and the background, the glossy reflections, the halftone dot patterns.

3) Scale-Invariant Feature Transform (SIFT):
SIFT detects and describes local features in images that are invariant to scale, rotation, and illumination changes. For anti-spoofing, researchers used SIFT to detect inconsistencies between different regions of the face. For example, a printed photo might have consistent texture across the entire face, while a real face has different textures on the nose, cheeks, and forehead.

4) The Fundamental Limitation:
The fundamental limitation of handcrafted approaches is their inability to learn discriminative features adaptively. As printing and display technologies improved, fixed feature extractors became obsolete. A method that worked well against 2010-era inkjet prints might fail completely against 2020-era high-gloss photo prints or 2024-era OLED screens. Researchers had to manually redesign features for each new attack type—a losing battle against determined adversaries.

B. The Motion Era: Asking Users to Move (2010-2016)
Researchers soon realized that static analysis had fundamental limits. A sufficiently high-quality print can look almost identical to a real face in a single frame. So, they started analyzing motion.

1) Eye Blink Detection:
The idea was simple: a real person blinks naturally (about 15-20 times per minute), and a printed photo cannot blink at all. Early systems would simply wait for a blink to occur naturally, while later systems actively asked the user to blink on command. The system would analyze the eye region across multiple frames, detecting the closing and reopening of the eyelids. The timing, duration, and symmetry of the blink provided liveness evidence.
However, video replay attacks quickly adapted. An attacker could simply record a video of the genuine user blinking and replay it. The system would see the blink and be fooled. This led to challenge-response systems where the system would ask the user to blink at a specific time, but attackers adapted by recording multiple blinks and selecting the appropriate one.

2) Lip Movement Analysis for Audio-Visual Liveness:
A more sophisticated approach combined audio and video. The system would speak a random phrase (e.g.," two seven four nine") and use lip-reading algorithms to verify that the user's lip movements matched the spoken phrase. A pre-recorded video replay would have lip movements that matched some other audio, not the challenge phrase.
However, this approach has a critical flaw that is rarely discussed: language dependency. Most audio-visual liveness systems are trained on English datasets like LRS2 (Lip Reading Sentences) or VoxCeleb. But lip movements vary dramatically across languages. Consider:

- The English" th" sound (as in" think") involves the tongue touching the teeth. Many Indian languages (Hindi, Telugu, Tamil) do not have this phoneme.
- Mandarin Chinese uses tones that involve subtle larynx movements that are barely visible on the lips.
- The Tamil retroflex consonants involve curling the tongue back, which changes lip shape differently than dental consonants.

A system trained only on English will inevitably perform poorly when faced with speakers of other languages. In a globalized world, this is not acceptable.

C. The Deep Learning Revolution (2015-Present)
The arrival of deep learning, particularly Convolutional Neural Networks (CNNs), transformed face anti-spoofing. Instead of hand-crafting features, researchers could now let the network learn the most discriminative features directly from data.

1) From Features to End-to-End Learning:
The key insight is that a CNN, given enough labeled data, will automatically discover the texture, color, motion, and reflection patterns that distinguish live faces from spoofs. The network learns hierarchical representations:

- Early layers learn simple patterns: edges, corners, color blobs.

- Middle layers learn more complex patterns: textures, skin pores, reflection highlights.
- Late layers learn task-specific patterns: the difference between a real eye and a printed eye, the difference between a live cheek and a screen-replayed cheek.

This end-to-end learning eliminates the need for manual feature engineering and adapts automatically to new attack types, provided the training data includes them.

2) Auxiliary Supervision:

Liu et al. [2] proposed a clever extension: auxiliary supervision. Instead of training the net-work only to classify live vs. spoof, they added auxiliary tasks such as:

- Facial depth estimation: A live face has a characteristic 3D shape (nose protrudes, eyes are recessed, cheeks curve). A printed photo is flat. The network learns to estimate depth maps from single images.
- Reflection analysis: A live face has specular reflections (the shiny spots on the nose and forehead) that follow physical laws. A print or screen has different reflection characteristics.
- rPPG signal extraction: The network learns to extract the pulse signal from facial video.

By learning these auxiliary tasks, the network develops richer internal representations that are more robust to spoofing. Liu's approach achieved state-of-the-art performance with an ACER below 5% on several benchmarks.

3) Transfer Learning and Pre-trained Models:

Training a deep CNN from scratch requires massive amounts of labeled data, which is often unavailable. Transfer learning solves this problem. Researchers take a network pre-trained on a large-scale face recognition dataset (e.g., VGG-Face, FaceNet, ArcFace) and fine-tune it for anti-spoofing.

The intuition is that face recognition and anti-spoofing share many low-level and mid-level features. Recognizing a face requires detecting eyes, nose, mouth, and their spatial arrangement. Distinguishing a live face from a spoof also requires these features, plus additional texture and reflection analysis. By starting from a pre-trained model, we can achieve excellent performance with much less labeled anti-spoofing data.

Common backbone architectures include:

- VGG-16/VGG-19: Deep but simple architecture with many parameters (138 million for VGG-16).
- ResNet-50/ResNet-101: Introduces skip connections to enable very deep networks (25-45 million parameters).
- MobileNet/MobileNetV2: Designed specifically for mobile and embedded devices (3-4 million parameters).
- EfficientNet: Optimizes for both accuracy and efficiency using neural architecture search.

4) Generative Adversarial Networks (GANs) for Anti-Spoofing:

GANs have found a unique role in anti-spoofing: generating synthetic spoof data. The classic problem is that we cannot anticipate all possible future attack types. A GAN-based approach trains:

- A generator that takes a real face image and tries to" spoof" it—add print-like artifacts, screen-like Moire´ patterns, or mask-like edges.
- A discriminator that tries to distinguish real live faces from both real spoofs and generated spoofs.

The generator and discriminator are trained adversarially: the generator tries to fool the discriminator, and the discriminator tries to catch the generator. This produces a discriminator that is robust to a wide range of spoof artifacts, including ones it has never seen before.

D. Datasets and Evaluation Protocols

Reproducible research requires standardized datasets and evaluation protocols. The following datasets have become benchmarks in the field.

1) CASIA Face Anti-Spoofing Database (CASIA-FASD):

Introduced in 2012 by Zhang et al. [3], this dataset contains 12,000 images from 50 subjects. For each subject, there are:

- Genuine (live) captures: 3 per quality level (low, normal, high) under varying illumination.
- Warped photo attacks: Printed photos slightly bent to simulate 3D curvature.
- Cut photo attacks: Printed photos with eye and

mouth regions cut out.
- Video replay attacks: Pre-recorded videos displayed on an electronic screen.

The dataset is relatively small by modern standards but re-mains widely used for benchmarking.

2) Replay-Attack Database:

Created by IDIAP Research Institute, this dataset contains 1,300 video recordings of 50 subjects. Attack types include:
- Print attacks (hand-held and fixed)
- Replay attacks on two different display devices (iPhone 3GS and iPad)

The dataset includes both fixed and hand-held attack presentations, simulating real-world variability.

3) MSU Mobile Face Spoofing Database:

Created by Michigan State University, this dataset focuses specifically on mobile device scenarios. It includes 1,100 video clips from 35 subjects, captured under various lighting conditions, with attacks including print, replay, and makeup.

4) ISO/IEC 30107 Standard:

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) jointly published the 30107 series [5] to standardize presentation attack detection evaluation. Key definitions:
- APCER (Attack Presentation Classification Error Rate): Proportion of attack presentations incorrectly classified as bona fide.
- BPCER (Bona Fide Presentation Classification Error Rate): Proportion of bona fide presentations incorrectly classified as attack.
- ACER (Average Classification Error Rate): (APCER+ BPCER) / 2.
- IAPMR (Implicit Attack Presentation Match Rate): For verification systems, the proportion of attack presentations that are accepted after matching.

Following these standards ensures fair comparison across different systems and research groups.

E. Critical Research Gaps: What the Textbooks Don't Tell You

After reviewing over 100 papers from IEEE Xplore, ACM Digital Library, CVPR, ICCV, and ECCV proceedings, we identified several critical gaps that are often mentioned briefly but rarely addressed thoroughly.

1) The Language Dependency Gap:

As discussed earlier, audio-visual liveness systems trained on English fail on speakers of other languages. This is not a minor issue—it is a fundamental bias. Most research is conducted in English-speaking institutions using English-speaking subjects. The resulting systems work well for English speakers but poorly for the majority of the world's population.

We need:
- Multilingual training datasets covering at least 20 major languages.
- Language-agnostic features that do not depend on specific viseme distributions.
- Cross-lingual evaluation protocols that explicitly test generalization.

2) The Resolution Constraint Gap:

Iris recognition is often touted as the most accurate biometric modality, with error rates as low as 1 in 2 million. However, this accuracy comes with a hidden requirement: high-resolution images. Most iris recognition systems need at least 200 pixels per inch (ppi) to reliably extract iris texture patterns.

Now consider a typical smartphone front-facing camera. At normal selfie distance (about 30 cm), a 5-megapixel camera (2592 × 1944) captures a face at roughly 80-120 ppi. The iris, which is about 11-12 mm in diameter, is represented by only 35-50 pixels across. This is simply not enough resolution for reliable iris-based liveness detection.

The same limitation applies to surveillance cameras. A typical security camera might capture a face at 20-40 ppi from a distance of 2-3 meters. At that resolution, the iris is represented by 10-20 pixels—barely a dot.

This means that for most real-world deployments, iris-based liveness is simply not practical. Researchers need to either:
- Develop methods that work with low-resolution imagery (10-50 pixels across the iris), or
- Accept that this modality is limited to high-end, controlled environments (e.g., passport control where users stand close to a dedicated camera).

3) The Accessibility Barrier Gap:

This is perhaps the most ethically concerning gap. Active liveness detection methods—blink, smile, turn your head—are effective, but they are also exclusionary. Consider the following populations:

- People with facial paralysis: Approximately 1 in 5,000 people have Bell's palsy or other conditions affecting facial movement. They cannot smile, blink, or raise their eyebrows on command.
- People with motor disorders: Parkinson's disease affects 1 in 500 people over 60. Essential tremor affects 1 in 20 people over 65. These individuals may not be able to turn their head precisely or hold still on command.
- People with visual impairments: Approximately 285 million people worldwide have visual impairments. They may not know where to look or how to align their face with the camera.
- Elderly users: Age-related changes in skin elasticity, facial muscle control, and visual acuity make active challenges more difficult.

A liveness system that requires specific facial movements is effectively discriminating against these populations. Yet, most active systems are designed without considering accessibility, and most passive systems are not evaluated on diverse populations.

We believe that liveness detection should be:

- Passive by default: No user cooperation required.
- Multi-modal when active: Offer voice or gesture alter-natives for those who cannot perform facial movements.
- Evaluated on diverse populations: Including people with disabilities, across age ranges, and across ethnicities.

III. METHODOLOGY

A. Dataset Description

We employ two publicly available benchmark datasets for training and evaluation.

1) CASIA Face Anti-Spoofing Database (CASIA-FASD):

The CASIA-FASD [3] is our primary training and evaluation dataset. Table I provides a detailed breakdown.

TABLE I: CASIA-FASD Dataset Detailed Composition

Attack Type	Subjects	Total Samples	Train/Val/Test Split
Warped Photo	50	4,000	2,800 / 600 / 600
Cut Photo	50	4,000	2,800 / 600 / 600
Video Replay	50	4,000	2,800 / 600 / 600
Live (Genuine)	50	3,000	2,100 / 450 / 450
Total	50	15,000	10,500 / 2,250 / 2,250

Key characteristics of CASIA-FASD:

- Resolution: Images are captured at three quality levels—low (320×240), normal (640×480), and high (1280×720).
- Illumination: Controlled lighting variations from 100 lux to 1000 lux.
- Background: Simple uniform backgrounds (white, black, and patterned) to focus on face analysis.
- Capture device: Sony NEX-5 digital camera with 16mm lens.
- Distance: Approximately 1 meter from subject to camera.

2) Replay-Attack Database:

To evaluate cross-database generalization, we also use the Replay-Attack Database. Table II presents its composition.

TABLE II: Replay-Attack Database Composition

Attack Type	Subjects	Video Clips	Resolution
Print (Hand-held)	50	250	320×240
Print (Fixed)	50	250	320×240
Replay (iPhone 3GS)	50	250	320×240
Replay (iPad)	50	250	320×240
Live (Genuine)	50	300	320×240
Total	50	1,300	-

Key characteristics of Replay-Attack:

- Resolution: All videos are captured at 320×240 pixels (much lower than CASIA).
- Capture device: MacBook Air built-in camera (low quality, typical of laptops).
- Lighting: Two lighting conditions—controlled (office lighting) and adverse (dark or backlight).

- Attack presentations: Both hand-held (simulating realistic attacker behavior) and fixed (simulating mounted spoofs).

3) Dataset Comparison and Complementarity:
CASIA-FASD and Replay-Attack complement each other well:

- CASIA provides high-resolution still images; Replay-Attack provides low-resolution videos.
- CASIA has controlled, studio-like capture; Replay-Attack has more realistic, webcam-like capture.
- CASIA includes warped and cut photos; Replay-Attack does not.
- Replay-Attack includes hand-held attacks; CASIA does not.

Evaluating on both datasets gives us insight into how well our model generalizes across capture conditions, resolutions, and attack types.

B. Preprocessing Pipeline
Before feeding images to our neural network, we perform a multi-step preprocessing pipeline.

1) Face Detection with MTCNN:
We employ the Multi-task Cascaded Convolutional Networks (MTCNN) detector, which provides:

- Bounding box coordinates for the face region.
- Five facial landmark points: left eye, right eye, nose, left mouth corner, right mouth corner.
- A confidence score for the detection. MTCNN operates in three stages:

1) Proposal Network (P-Net): Quickly scans the image at multiple scales to propose candidate face regions.
2) Refinement Network (R-Net): Refines the candidate regions and rejects false positives.
3) Output Network (O-Net): Outputs the final bounding box and facial landmarks.

We discard any image where MTCNN confidence is below 0.95, ensuring high-quality inputs.

2) Face Alignment and Normalization:
Using the detected eye coordinates, we perform similarity transformation to align each face to a canonical coordinate system:

1) Compute the center point between the two eyes.
2) Compute the angle of the line connecting the eyes.
3) Rotate the image so that the eyes are horizontally aligned.
4) Scale the image so that the inter-pupillary distance is normalized to 60 pixels.
5) Crop the face region to 224×224 pixels, centered on the eyes.

This alignment ensures that the network sees faces in a consistent orientation and scale, reducing the need to learn position-invariant features.

3) Data Augmentation:
To increase effective dataset size and improve generalization, we apply online data augmentation during training. Each image is randomly transformed before being fed to the network:

1) Horizontal flip (probability 0.5): Makes the model in-variant to left-right orientation.
2) Rotation (-10° to +10°): Handles slight head tilts.
3) Brightness adjustment (factor 0.8 to 1.2): Simulates different lighting conditions.
4) Contrast adjustment (factor 0.8 to 1.2): Simulates different camera settings.
5) Zoom (factor 0.9 to 1.1): Simulates different distances from the camera.
6) Gaussian noise (= 0 to 0.05): Simulates sensor noise in low light.

These augmentations are applied in random order with random parameters. We do not apply augmentations to validation or test images.

C. Network Architecture
Our proposed CNN architecture is designed with three primary objectives:
1) High classification accuracy:
Must reliably distinguish live faces from spoofs.

2) Computational efficiency:
Must run in real-time on resource-constrained devices (smartphones, edge cam-eras, embedded systems).

3) Parameter efficiency:
Must be small enough to fit in memory and avoid overfitting on moderate-sized datasets.
Table III presents the complete architecture with detailed layer specifications.

TABLE III: Complete CNN Architecture with Layer Details

Layer Type	Configuration	Output Size P
Input	224 × 224 × 3	224 × 224 × 3
Conv2D	3 × 3, 32, stride 2, padding same	112 × 112 × 32
BatchNorm	-	112 × 112 × 32
ReLU	-	112 × 112 × 32
Depthwise Conv2D	3 × 3, 32, stride 1, padding same	112 × 112 × 32
BatchNorm	-	112 × 112 × 32
ReLU	-	112 × 112 × 32
Conv2D	1 × 1, 64, stride 1	112 × 112 × 64
BatchNorm	-	112 × 112 × 64
ReLU	-	112 × 112 × 64
MaxPool2D	2 × 2, stride 2	56 × 56 × 64
Conv2D	3 × 3, 128, stride 1, padding same	56 × 56 × 128
BatchNorm	-	56 × 56 × 128
ReLU	-	56 × 56 × 128
MaxPool2D	2 × 2, stride 2	28 × 28 × 128
Conv2D	3 × 3, 256, stride 1, padding same	28 × 28 × 256
BatchNorm	-	28 × 28 × 256
ReLU	-	28 × 28 × 256
MaxPool2D	2 × 2, stride 2	14 × 14 × 256
Global Average Pooling	-	256
Dropout	rate = 0.5	256
Dense	128 units	128
Dropout	rate = 0.3	128
Dense	1 unit (Sigmoid)	1
Total Parameters		

1)Depth wise Separable Convolutions:

The architecture employs depth wise separable convolutions, which factor standard convolutions into two separate operations:

1) Depth wise convolution:

Applies a single filter per input channel. For an input with C_{in} channels, this uses C_{in} filters, each of size k X k.

B

2) Pointwise convolution:

Applies a 1 X 1 convolution to combine the outputs of the depth wise layer across channels.

The parameter reduction is substantial. A standard 3 3 convolution with C_{in} input channels and C_{out} output channels requires $3 \times 3 \times C_{in} \times C_{out}$ parameters. A depth wise separable convolution requires 3 X 3 X $C_{in} + C_{in}$ X C_{out} parameters. For C_{in} = 32 and C_{out} = 64, this reduces parameters from 18,432 to 2,432—a factor of 7.6× reduction.

2)Batch Normalization:

Batch normalization normalizes the activations of each layer to have zero mean and unit variance across the mini-batch. The normalized output is:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

where μ_B and σ^2 are the mean and variance over the mini-batch. The normalized output is then scaled and shifted:

$$y = \gamma \hat{x} + \beta$$

where γ and β are learnable parameters. Batch normalization provides several benefits:

- Stabilizes training by reducing internal covariate shift.
- Enables higher learning rates, accelerating convergence.
- Provides a mild regularization effect, reducing

overfitting

3)Global Average Pooling vs. Flatten:
Instead of flattening the feature maps into a long vector and adding multiple dense layers, we use global average pooling. This operation computes the average of each feature map across all spatial locations. For an input of size *H X W X C*, global average pooling outputs a *C*-dimensional vector where each element is the average of one feature map.

Advantages of global average pooling:
- Drastically reduces parameter count (no flattening weights).
- Forces the feature maps to be class-specific (each feature map should represent a meaningful pattern).
- Reduces overfitting by eliminating the need for large dense layers.

4)Dropout Regularization:
We apply two dropout layers with rates of 0.5 and 0.3. Dropout randomly sets a fraction of input units to 0 at each training step. This prevents co-adaptation of neurons—the network cannot rely on any single feature and must learn redundant representations. Dropout is applied only during training; at test time, all units are used (with appropriate scaling).

D. Training Protocol
1) Data Splitting:
We partition the CASIA-FASD dataset into three disjoint sets:
- Training set: 70% (10,500 images). Used to update model weights.
- Validation set: 15% (2,250 images). Used for hyperparameter tuning and early stopping.
- Test set: 15% (2,250 images). Held out until final evaluation.

Stratification ensures that each partition maintains the same distribution of subjects and attack types. No subject appears in more than one partition, ensuring subject-disjoint evaluation.

2) *Loss Function:* We employ the binary cross-entropy loss function:

$$L(\vartheta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1-y_i)\log(1-p_i)]$$

where:
- N is the batch size.
- Yi 0, 1 is the ground truth (0 = spoof, 1 = live).
- pi = $f_\theta(x_i)$ is the predicted probability of being live. This loss function penalizes confident wrong predictions more heavily than uncertain ones.

3) *Optimizer:*
We use the Adam (Adaptive Moment Estimation) optimizer, which maintains per-parameter learning rates based on the first and second moments of the gradients. The update rules are:

$$m_t = \delta_1 m_{t-1} + (1-\delta_1)\nabla_\theta L_t \quad (1)$$

$$v_t = \delta_2 v_{t-1} + (1-\delta_2)(\nabla_\theta L_t)^2 \quad (2)$$

$$\hat{m}_t = \frac{m_t}{1-\delta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\delta_2^t} \quad (3)$$

$$\vartheta_{t+1} = \vartheta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon} \quad (4)$$

We use the following hyperparameters:
- Learning rate η = 1 x 10−4
- β1 = 0.9 (exponential decay rate for first moment)
- β2 = 0.999 (exponential decay rate for second moment)
- ϵ = 1×10^{-8} (numerical stability constant)

4) *Training Schedule:*
We train for a maximum of 50 epochs with a batch size of 32. An epoch means one complete pass through the entire training set (10,500 images → 329 batches). The training schedule includes:
- Early stopping: Monitor validation loss after each epoch. If validation loss does not decrease for 10 consecutive epochs, stop training.
- Model checkpointing: Save the model weights that achieve the lowest validation loss.
- Learning rate warmup: For the first 5 epochs, linearly increase the learning rate from 1 10^{-5} to 1 10^{-4} to stabilize early training.

5) *Computational Resources:*
All experiments were con-ducted on the following hardware:
- CPU: Intel Xeon Silver 4214 (12 cores, 2.2 GHz)
- GPU: NVIDIA Tesla T4 (16 GB VRAM)
- RAM: 64 GB DDR4 ECC

- Storage: 1 TB NVMe SSD

Total training time for 50 epochs was 2 hours and 15 minutes. Inference time per image on the T4 GPU averaged 8.3 milliseconds (120 FPS). On a smartphone CPU (Qualcomm Snapdragon 888), inference time averaged 33 milliseconds (30 FPS), sufficient for real-time applications.

IV. EXPERIMENTAL RESULTS

A. Performance Metrics

Following ISO/IEC 30107-3, we report the following metrics:

- APCER (Attack Presentation Classification Error Rate): Proportion of attack presentations incorrectly classified as bona fide.

$$APCER = \frac{\text{False Acceptances of Attacks}}{\text{Total Attack Presentations}}$$

- BPCER (Bona Fide Presentation Classification Error Rate): Proportion of bona fide presentations incorrectly classified as attack.

$$BPCER = \frac{\text{False Rejections of Genuine}}{\text{Total Bona Fide Presentations}}$$

- ACER (Average Classification Error Rate):

$$ACER = \frac{APCER + BPCER}{2}$$

- Accuracy: Proportion of correct classifications overall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: Of all positive predictions, how many were correct.

$$Precision = \frac{TP}{TP + FP}$$

- Recall (True Positive Rate): Of all actual positives, how many were correctly identified.

$$Recall = \frac{TP}{TP + FN}$$

- F1-Score: Harmonic mean of precision and recall.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

B. Results on CASIA-FASD

Table IV presents the performance metrics achieved by our proposed model on the CASIA-FASD test set.

TABLE IV: Performance Metrics on CASIA-FASD Test Set

Metric	Value
APCER	0.2754
BPCER	0.0162
ACER	0.1458
Accuracy	0.8032
Precision	0.6088
Recall	0.9838
F1-Score	0.8369
Training Accuracy (50 epochs)	0.9337
Validation Accuracy (best)	0.9120

Interpretation:

- BPCER of 1.62% means that only about 1 in 62 genuine users would be incorrectly rejected. For a smartphone with 1 million users, this would mean about 16,000 users experiencing a false rejection. This is acceptable but could be improved.
- APCER of 27.54% means that about 1 in 3.6 attack attempts would succeed. For a high-security application (e.g., bank transaction authorization), this is unacceptably high. However, for a low-security application (e.g., phone unlock where the attacker needs physical possession of the device), this might be tolerable.
- Training vs. Validation gap: The 2.17% gap between training accuracy (93.37%) and validation accuracy (91.20%) indicates mild overfitting. The dropout and data augmentation are working but could be increased.

C. Confusion Matrix Analysis

Table V presents the normalized confusion matrix, providing per-class performance breakdown.

TABLE V: Normalized Confusion Matrix on CASIA-FASD

	Predicted Live	Predicted Spoof
Actual Live	0.9838 (True Live)	0.0162 (False Spoof)
Actual Spoof	0.2754 (False Live)	0.7246 (True Spoof)

The high true positive rate (0.9838) demonstrates the model's ability to reliably recognize genuine users.

This is the most important metric for user experience. The true negative rate (0.7246) indicates that approximately three-quarters of at-tacks are correctly rejected, while one-quarter evade detection.

D. Per-Attack-Type Analysis

To better understand failure modes, we analyze detection performance separately for each attack type (Table VI).

TABLE VI: Detection Accuracy by Attack Type on CASIA-FASD

Attack Type	Detection Accuracy (%)
Warped Photo	78.3%
Cut Photo	81.2%
Video Replay (standard, 60 Hz)	72.9%
Video Replay (high-refresh, 240 Hz)	55.3%

Key observations:

- Cut photos (81.2% accuracy) are the easiest to detect. The cut-out holes around the eyes and mouth create obvious discontinuities in texture and depth.
- Warped photos (78.3% accuracy) are moderately difficult. The warping creates some depth cues, but high-quality glossy prints can still fool the system.
- Standard video replay (72.9% accuracy) is challenging. The model relies on subtle Moire´ patterns and flicker artifacts that are less visible on modern displays.
- High-refresh video replay (55.3% accuracy) is extremely challenging. At 240 Hz, the flicker is imperceptible to both humans and most cameras. The model's performance drops to near-random.

E. Cross-Database Generalization: CASIA to Replay-Attack

One of the most important tests of a model's real-world utility is how well it generalizes to data collected under differ-ent conditions. We trained our model on CASIA-FASD (high-resolution, controlled lighting, DSLR camera) and evaluated it on Replay-Attack (low-resolution, variable lighting, webcam). Table VII presents the results.

TABLE VII: Cross-Database Generalization (Train on CASIA, Test on Replay-Attack)

Metric	Value
Accuracy	0.675
APCER	0.412
BPCER	0.238
ACER	0.325

The accuracy drops from 91.20% on CASIA to 67.5% on Replay-Attack—a dramatic 23.7 percentage point decrease. This highlights the critical challenge of domain shift in face anti-spoofing.

Why does this happen?

1) Resolution mismatch:

CASIA uses high-resolution (up to 1280×720) images; Replay-Attack uses low-resolution (320×240) videos. Fine texture features learned on high-resolution images are not present in low-resolution inputs.

2) Camera differences:

CASIA uses a Sony NEX-5 DSLR with a high-quality lens; Replay-Attack uses a MacBook Air webcam with a low-quality plastic lens. Color response, noise characteristics, and lens distortions differ.

3) Lighting differences:

CASIA uses controlled studio lighting; Replay-Attack includes adverse lighting (dark, backlight, direct sunlight).

4) Attack presentation:

CASIA uses fixed, stationary at-tacks; Replay-Attack includes hand-held attacks where the attacker moves the spoof.

This result has important practical implications: a model that performs well in the lab may fail dramatically in the real world. Cross-database evaluation should be standard practice, but it remains surprisingly rare in the literature.

F. Visualization of Learned Features

Using Gradient-weighted Class Activation Mapping (Grad-CAM), we visualize the spatial regions most influential in the model's classification decisions. Grad-CAM computes the gradient of the predicted class score with respect to the feature maps of the final convolutional layer, producing a heatmap that highlights important regions.

For genuine faces: Attention concentrates on:

- The periocular region (around the eyes): Fine skin texture, eyelashes, and the subtle sheen of the cornea provide strong liveness signals.
- The cheeks: Skin pores, fine hairs, and color variations.
- The nose tip: Specular highlights (the shiny spot) that follow physical reflection laws.

- For print attacks: Attention shifts to:
- The face boundaries: Where the printed paper ends and the background begins.
- The eye and mouth regions: Halftone dot patterns are most visible in these high-contrast areas.
- Glossy reflection spots: Paper reflects light differently than skin.

- For replay attacks: The model attends to:
- Moire´ patterns: The interference pattern between the camera sensor and display subpixels.
- Screen bezel edges: The boundary between the display and the device frame.
- Color temperature inconsistencies: Screen displays often have a different color balance than ambient lighting.

G. Ablation Studies

To understand the contribution of each architectural component, we performed ablation studies (Table VIII). We trained several variants of our model, each missing one key component, and evaluated them on the CASIA validation set.

TABLE VIII: Ablation Studies (Validation Accuracy)

Model Variant	Validation Accuracy (%)
Full model (proposed)	91.20
No depth wise separable conv (standard conv)	90.45
No batch normalization	87.32
No dropout	88.76
No data augmentation	85.91
Single dense layer (no intermediate 128-unit layer)	88.23

Key findings:

- Data augmentation has the largest impact (+5.29%). Without augmentation, the model overfits significantly, achieving only 85.91% validation accuracy despite 94.2% training accuracy.
- Batch normalization is also critical (+3.88%), primarily enabling stable training with a higher learning rate.
- Depthwise separable convolution provides a slight ac-curacy improvement (+0.75%) while drastically reducing parameters (by 68% compared to standard convolutions).
- Dropout provides a moderate improvement (+2.44%).

V. LIMITATIONS AND CRITICAL ANALYSIS

No academic paper is complete without an honest discussion of limitations. We want to be transparent about where our system falls short, both because it is good scientific practice and because it points the way toward future improvements.

A. Environmental Sensitivity

Our model performs well under ideal conditions—good lighting, a steady camera, a cooperative user looking straight at the lens. But real life is rarely ideal. Table IX quantifies performance degradation under challenging conditions.

TABLE IX: Performance Degradation Under Challenging Conditions

Condition	Accuracy (%)
Ideal (reference)	91.20
Low light (¡ 50 lux)	68.3
Low resolution (simulated 640×480)	71.2
Motion blur (5 pixels/frame)	65.1
Extreme angle (¿30° rotation)	74.3
Mixed conditions (low light + motion)	54.8

Why this happens:

- Low light:

The fine texture details that the model relies on become indistinguishable. Sensor noise increases, creating artificial patterns that the model might misinterpret as spoof artifacts.

- Low resolution:

Down sampling eliminates high-frequency texture information. The model loses the ability to distinguish skin pores from halftone dots.

- Motion blur:

Blurring also eliminates high-frequency information and can make a live face look unnaturally smooth (like a print).

- Extreme angles:

Facial texture and reflection patterns change with viewing angle. The model has not seen enough angled examples during training.

B. Generalization to Advanced Attacks

Our model was trained on the CASIA dataset, which does not include several types of modern spoof attacks. We created small custom datasets (200-500 samples each) to test generalization (Table X).

TABLE X: Generalization to Unseen Attack Types

Attack Type	Detection Accuracy (%)
3D Silicone Mask (custom, 200 images)	62.0
Deepfake (StyleGAN3, 500 videos)	58.0
Deepfake (DiffAE diffusion, 500 videos)	59.5
High-refresh replay (240 Hz OLED, 200 videos)	55.3
Adversarial makeup (patterned, 100 images)	48.0

These results are concerning. A 3D silicone mask (costing 1, 000 5,000) fools our model 38% of the time. A deepfake video fools it 40-42% of the time. Adversarial makeup a pattern of dots and lines painted on the face fools it 52% of the time.

This highlights the fundamental challenge of generalization to unseen attack distributions. The arms race between attackers and defenders continues.

C. The Bias Question: Does Our Model Work for Everyone?

We have not systematically evaluated our model's performance across different demographic groups. This is a known problem in facial recognition—many systems have been shown to have higher error rates for women, darker-skinned individuals, and older adults.

The CASIA dataset has reasonable diversity (50 subjects from multiple ethnic backgrounds), but it is not perfect. We encourage future work to:

- Evaluate models on diverse datasets (e.g., BUPT-Globalface, RFW) that explicitly balance demographics.
- Report performance separately for different demographic groups.
- Measure and mitigate performance disparities.

Fairness in biometrics is not just a technical problem; it is an ethical imperative.

VI. FUTURE RESEARCH DIRECTIONS

Based on our findings and limitations, we believe the following directions are most promising for future research.

A. Multimodal Fusion

No single modality is perfect. But by combining multiple modalities, we can achieve security that is greater than the sum of its parts.

- Thermal imaging:

A thermal camera detects heat emitted by the face. Live faces have a characteristic temperature distribution (warmer around the eyes and mouth, cooler on the cheeks). Masks and prints cannot reproduce this thermal pattern. Consumer thermal cameras like the FLIR One Pro cost less than $400 and are becoming more common.

- 3D depth sensing:

Structured light sensors (like Apple's Face ID) or time-of-flight sensors can measure the 3D shape of the face. A printed photo is perfectly flat. A 3D mask has some depth, but it may not match the precise 3D structure of a real face.

- Audio-visual synchronization:

The system can speak a random phrase and verify that the user's lip movements are synchronized with the audio.

- Remote photoplethysmography (rPPG):

The human heart pumps blood with each beat, causing subtle color changes in the face. rPPG extracts this pulse signal from a regular video camera. A print or mask has no pulse.

The challenge is combining these modalities

efficiently. A naive approach would run four separate models and combine their scores, but that would be computationally expensive. Research is needed on lightweight fusion architectures that run in real time on edge devices.

B. Few-Shot and Continual Learning

New spoof attacks emerge all the time. A system that is trained once and deployed forever will eventually become obsolete. We need systems that can learn and adapt.

- Few-shot learning:

The system can learn to recognize a new type of spoof attack from only a handful of examples (say, 5 to 10). Techniques like prototypical networks or model-agnostic meta-learning (MAML) show promise.

- Continual learning:

The system continuously learns from new data without forgetting what it already knows. Elastic Weight Consolidation (EWC) and Progressive Neural Networks prevent" catastrophic forgetting."

- Anomaly detection:

Train only on live faces. At inference time, any input sufficiently different from the distribution of live faces is flagged as a spoof. This approach can theoretically detect novel spoof types without ever having seen them.

C. Privacy-Preserving Architectures

Facial images are sensitive biometric data. Users are right to be concerned about how their face data is stored and used.

- On-device processing: All inference runs on the user's device. The only thing transmitted to the server is an authentication token. No face images ever leave the device. Apple's Face ID uses this approach.
- Homomorphic encryption: Computation on encrypted data. The server runs the liveness detection model on an encrypted face image without ever decrypting it.
- Federated learning: Each device trains locally on its own face images, then sends only model updates (gradients) to a central server. The server never sees individual face images.

D. Better Datasets

The progress of deep learning is often limited by data. We call on the research community to create:

- A large-scale dataset of 3D masks from different materials (silicone, resin, latex, gelatin).
- Deepfake videos generated by multiple GAN and diffusion architectures with detailed metadata.
- Diverse subjects covering a wide range of ethnicities, ages (5 to 80 years), and languages (20+ languages for lip-sync).
- Realistic variations in lighting (5 to 5000 lux), motion blur, and camera types.
- A standardized cross-database evaluation protocol.

VII. CONCLUSION

This paper presented a compact convolutional neural net-work architecture for face anti-spoofing detection. Our model achieves 93.37% training accuracy and 91.20% validation ac-curacy on the CASIA Face Anti-Spoofing Database with only 405,377 parameters, demonstrating that high performance is achievable with computationally efficient architectures suitable for edge deployment.

The asymmetric error profile (APCER = 27.54%, BPCER = 1.62%) reflects the relative difficulty of detecting sophisticated presentation attacks compared to verifying genuine users. Video replay attacks on high-quality displays are particularly challenging (72.9% detection accuracy for 60 Hz displays, dropping to 55.3% for 240 Hz displays). Cross-database generalization from CASIA to Replay-Attack reveals a significant domain shift, with accuracy dropping from 91.20% to 67.5%. Beyond our specific results, this work highlights persistent research gaps requiring community attention: cross-lingual dependencies in audio-visual methods, resolution constraints limiting iris-based approaches, and accessibility barriers excluding users with disabilities. The arms race between at-tack generation and defense detection continues. No single approach provides comprehensive protection across all attack types and deployment conditions.

We recommend multimodal fusion (combining visible light with thermal and depth), continual learning for adaptation to new attack types, and privacy-preserving architectures as the most promising directions for future research. Most importantly, we call for greater

attention to accessibility and fairness—liveness detection must work for everyone, not just the able-bodied and the well-represented.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and Engineering at Koneru Lakshmaiah Education Foundation for providing computational resources and research support. We acknowledge the creators of the CASIA Face Anti-Spoofing Database and the Replay-Attack Database for making their data publicly available, enabling reproducible research in this critical area. We also thank the anonymous reviewers for their constructive feedback.

REFERENCES

[1] Z. Boulkenafet, J. Komulainen, and A. Hadid, "Face spoofing detection using colour texture analysis," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1818–1830, Aug. 2016.

[2] Y. Liu, A. Jourabloo, and X. Liu, "Learning deep models for face anti-spoofing: Binary or auxiliary supervision," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 389–398.

[3] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li, "A face anti-spoofing database with diverse attacks," in *Proc. IAPR Int. Conf. Biometrics (ICB)*, 2012, pp. 26–31.

[4] Chingovska, A. Anjos, and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing," in *Proc. IEEE Int. Conf. Biometrics Special Interest Group (BIOSIG)*, 2012, pp. 1–7.

[5] ISO/IEC 30107-1:2016, *Information Technology—Biometric Presentation Attack Detection—Part 1: Framework*. Geneva, Switzerland: International Organization for Standardization, 2016.

[6] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2268–2283, Oct. 2016.

[7] Nguyen, N. Pham, and S. Sridharan, "Deep learning for face anti-spoofing: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4244–4262, Dec. 2021.

[8] Z. Wang, C. Wang, and Y. Wang, "A comprehensive survey on face anti-spoofing: Databases, methods, and challenges," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–37, Mar. 2020.