Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
DOI : 10.47974/JIOS-2479

Deepfake detection using multiple deep learning architectures and HO-PLO hybrid bio-inspired optimization

Ahmed Saad Al-Qazzaz [†]
Pedram Salehpour *
Hadi S. Aghdasi [§]
Department of Computer Engineering
Faculty of Electrical and Computer Engineering
University of Tabriz
Tabriz
Iran

Abstract

Deepfakes pose an increasing threat to digital media authenticity, public trust, and forensic reliability because recent generative models can create highly realistic manipulated facial content. Although contemporary deepfake detectors frequently report high accuracy under within-dataset evaluation, their performance remains sensitive to architecture selection and hyperparameter configuration, while evidence regarding their stability, computational feasibility, and generalization is often limited. This study proposes a comprehensive deepfake detection framework that evaluates ten diverse deep learning architectures, covering convolutional neural networks, Vision Transformers, self-supervised models, multiscale networks, and hybrid CNN-Transformer designs. The architectures are optimized using a novel hybrid bio-inspired algorithm that combines Hippopotamus Optimization for global exploration with Polar-Magnetic-Field-guided Local Optimization for fine-grained exploitation, referred to as HO-PLO. An adaptive switching mechanism and a receiver operating characteristic area-under-the-curve-based fitness function are employed to guide the hyperparameter search. The framework was evaluated on two complementary image datasets: the Real and Fake Face Detection dataset containing 2,041 images and the 140k Real and Fake Faces dataset containing 140,000 images. Under five-fold cross-validation, CoAtNet-IF achieved the highest accuracy on Dataset 1 at 99.55 ± 0.41%, whereas ConvNeXt-FD achieved the highest accuracy on Dataset 2 at 99.62 ± 0.24%.

[†] *ORCID-ID:* **0009-0006-3837-6048**
* *ORCID-ID:* **0000-0002-1300-7848**
[§] *ORCID-ID:* **0000-0003-1613-7370**

[†] *E-mail:* **Msc.ahmedalqazaz@gmail.com**
* *E-mail:* **psalehpoor@tabrizu.ac.ir (Corresponding Author)**
[§] *E-mail:* **aghdasi@tabrizu.ac.ir**

© Taru Publications

For most representative architectures, HO-PLO outperformed the individual HO and PLO optimizers, with an improvement of up to 1.67 percentage points over HO. Paired statistical tests, effect-size analysis, ablation experiments, convergence analysis, and computational evaluation further demonstrate the stability and practical efficiency of the framework. The principal contribution of this study is therefore not limited to a single high-performing detector; it provides a unified architecture-optimizer evaluation framework that identifies effective combinations of modern deep networks and adaptive bio-inspired optimization. The reported findings establish strong within-dataset image-classification performance, while direct cross-dataset and video-level generalization remain important directions for future investigation.

Subject Classification: *68T07, 68T45, 68U10.*

Keywords: *Deepfake detection, Deep learning, Transformer, HO-PLO optimization, Hippopotamus optimizer, Polar local optimization, Hyperparameter optimization.*

1. Introduction

The increasing popularity of digital communication technologies and the rapid rise of social media as a dominant platform has fundamentally changed the way multimedia content is produced, published, and consumed. The proliferation of low-cost mobile devices such as smartphones, tablets, and digital cameras has enabled users to generate and share enormous volumes of video and image content across global networks [1]–[3]. However, this openness has also facilitated malicious manipulation of visual and audio content through deep learning technologies, giving rise to the phenomenon known as deepfakes. These are highly realistic yet artificial media artifacts produced through methods such as autoencoders and generative adversarial networks (GANs) [4]–[7]. Their hyper-realism renders them difficult to detect, particularly by the human eye, and therefore poses a significant threat to the integrity of digital media and public trust.

Deepfakes have been exploited for a range of abusive purposes, including fabricated political speeches, defamation campaigns, financial fraud, and information warfare. For example, fabricated videos may replicate purported military offensives or synthesize satellite imagery of nonexistent objects, disrupting intelligence operations and international relations [8], [9]. Additionally, impersonation on social networks has become widespread, with misinformation, identity theft, and non-consensual intimate imagery increasingly produced through fake face-swaps and distorted media [10]–[12]. Such abuse can have severe reputational consequences and a misleading impact on public opinion [13], [14].

Deepfakes are technically grounded in computer vision and deep learning algorithms for facial manipulation. Common categories include expression swapping, full-face synthesis, attribute modification, and identity replacement [15]–[17]. These manipulations are trained on massive collections of video and image data, and they become especially dangerous when applied to high-profile individuals such as politicians or celebrities, for whom abundant reference

imagery is publicly available [18]–[20]. As GAN architectures become more advanced, traditional rule-based or visual-inspection methods of detecting fake content become progressively less effective.

In response, deep-learning-based detection models have been trained to identify artifacts and inconsistencies introduced during deepfake synthesis. Transfer learning (TL) has demonstrated substantial potential for improving detection accuracy through pretrained models such as Inception, ResNet, and Xception [21]–[23]. TL is effective at capturing domain-invariant spatial features, enabling faster training and higher accuracy on smaller datasets. Nevertheless, most current models continue to struggle with generalization to unseen manipulations and remain difficult to interpret, particularly when applied to unfamiliar data or adversarial conditions [24]. Large benchmarks such as FaceForensics++ [25], Celeb-DF [26], and DFDC [27] have enabled this progress and have also exposed the cross-dataset generalization gap.

Although detection accuracy has improved considerably, several critical limitations remain. First, most high-performing models function as black boxes that provide little insight into their prediction processes [28], [29]. This opacity restricts their applicability in sensitive domains such as legal proceedings and digital forensics, where accountability is paramount. Second, existing work has insufficiently explored the combination of complementary metaheuristic optimization techniques to enhance detection performance. Finally, although interpretability tools such as LIME (Local Interpretable Model-Agnostic Explanations) exist, their application within the deepfake-detection domain has so far been superficial [30].

Recent research published in 2026 indicates that deepfake-detection evaluation is shifting from the pursuit of near-perfect accuracy on a single benchmark toward more demanding assessments involving cross-dataset generalization, parameter efficiency, temporal modeling, and robustness to previously unseen manipulations. Yermakov et al. [31] introduced GenD, a parameter-efficient detector that adapts only a small proportion of a pretrained vision encoder and reported a mean cross-dataset AUROC of 91.6% across 14 benchmark datasets, demonstrating that broad training-data diversity and paired real-fake samples are important for reducing shortcut learning.

Petmezas [32] conducted a unified cross-dataset comparison of Transformer and CNN architectures for video deepfake detection. TimeSformer obtained 78.4% accuracy, an AUC of 0.801, and an F1-score of 77.0% under a challenging evaluation protocol involving limited target-domain fine-tuning, illustrating that performance under unseen-domain conditions can be substantially lower than that reported by within-dataset studies.

Recent image-based research is also directly relevant to the datasets considered in the present study. Yesugade and Jadhav [33] evaluated Xception, DenseNet121, ResNet101, Vision Transformer, and ensemble models on a balanced dataset of 140,000 real and synthetic facial images. ResNet101 obtained 94.67% accuracy and an AUC of 0.987, while the ensemble achieved a recall of 96.60%, with the study emphasizing robustness and cross-dataset validation rather than a single accuracy figure.

Despite these advances, the literature still lacks a controlled investigation of how a hybrid global-local metaheuristic optimizer interacts with a broad and architecturally diverse collection of deepfake detectors under an identical preprocessing, training, and validation protocol. Most previous studies optimize or compare only one or a small number of architectures, making it difficult to distinguish improvements attributable to network design from those attributable to hyperparameter selection. The present work addresses this gap by jointly studying ten deep learning architectures and three optimizer configurations—HO, PLO, and the proposed HO-PLO—under a common experimental framework.

To address these research gaps, this study proposes a novel deepfake detection framework that integrates two core components:

1. *Multiple deep learning architectures.* Ten state-of-the-art models spanning transformers, CNNs, and hybrid architectures are employed, including DeiTFake, HDFF, SS-FDD, Swin-TF, CoAtNet-IF, DaViT-DD, EfficientNetV2-F, ConvNeXt-FD, MS-TAD, and Hybrid-CNN-ViT.
2. *HO-PLO hybrid optimization.* A novel combination of Hippopotamus Optimization (HO) for global exploration and Polar-Magnetic-Field-guided Local Optimization (PLO) for fine-grained local search, guided by a ROC-AUC-based fitness function.

The main contributions of this study are summarized as follows:

1. *Unified multi-architecture evaluation.* A controlled deepfake-detection framework evaluates ten architectures representing CNN, Transformer, self-supervised, multiscale, fusion, and hybrid CNN-Transformer paradigms under identical preprocessing and five-fold cross-validation protocols.
2. *Novel HO-PLO optimization strategy.* A hybrid bio-inspired optimizer combines the global exploration capacity of Hippopotamus Optimization with the fine-grained local exploitation of Polar-Magnetic-Field-guided Local Optimization, using an adaptive switching mechanism to balance both search behaviors.
3. *Architecture-aware optimizer analysis.* HO, PLO, and HO-PLO are compared on representative networks, showing that optimizer effectiveness is architecture-dependent, with gains reaching 1.67 percentage points over HO for CoAtNet-IF.
4. *Comprehensive experimental validation.* Performance is examined using accuracy, precision, recall, F1-score, ROC-AUC, five-fold cross-validation, paired statistical tests, effect sizes, ANOVA, Friedman testing, convergence analysis, and ablation experiments.
5. *Accuracy-efficiency assessment.* Training time, optimization cost, inference latency, and frames-per-second are evaluated in addition to predictive

performance, demonstrating that high detection performance can coexist with practical inference efficiency.

6. *Strong within-dataset results.* CoAtNet-IF achieves 99.55% accuracy on Dataset 1, while ConvNeXt-FD achieves 99.62% accuracy on Dataset 2, identifying effective architecture-optimizer combinations for image-based deepfake detection.

The remainder of this paper is organized as follows. Section 2 reviews related literature on deepfake detection. Section 3 presents the proposed HO-PLO methodology and model architectures. Section 4 reports the experimental setup and results. Section 5 discusses the findings, and Section 6 concludes the paper.

2. Related Work

Generative models have driven substantial progress in deepfake-detection research. Whereas earlier techniques emphasized handcrafted features and simple classifiers, more recent approaches rely on deep learning, attention mechanisms, and data augmentation. A recurring theme across this literature is the trade-off among accuracy, robustness across manipulation types, computational efficiency, and explainability.

2.1. *Transfer learning approaches*

Zanje, Bongale, and Dharrao [34] implemented three CNN-based transfer-learning models—VGG19, InceptionV3, and DenseNet201—for facial image-forgery classification. The best accuracy of 99% was obtained with DenseNet201, compared to InceptionV3 (94%) and VGG-19 (84%). Their experiments showed that transfer-learning models pretrained on large datasets can generalize to image-forgery detection with minimal fine-tuning.

Al-Dulaimi and Kurnaz [35] designed a hybrid CNN-LSTM model for precise deepfake-image detection, in which convolutional layers represented spatial features and LSTM layers captured temporal dependencies. Evaluated on the DFDC and Ciplab datasets, the model achieved 98.21% precision with an error rate of 0.26%.

2.2. *Bio-inspired optimization methods*

Alazwari et al. [36] proposed an Artificial Rabbits Optimization with Transfer Learning Deepfake Detection (AROTL-DFDBA) framework, combining a tuned DarkNet-53 network with Artificial Rabbits Optimization (ARO) for hyperparameter tuning, achieving 96.48% accuracy. Cunha et al. [37] proposed Particle Swarm Optimization (PSO)-based hyperparameter selection for video deepfake detection. Their PSO-tuned EfficientNet-GRU model achieved 95.17% accuracy, demonstrating the effectiveness of optimization-guided architectures. Al-Adwan et al. [38] developed a hybrid CNN-RNN model optimized by PSO, tested on the Celeb-DF and DFDC datasets, achieving maximum accuracies of 97.26% and 94.2%, respectively.

2.3. *Transformer-based approaches*

Recent developments in vision transformers demonstrate their utility for deepfake detection. ViT, DeiT, and Swin Transformer models have proven effective at learning long-range dependencies and fine-grained artifacts that distinguish real from fake faces. Karathanasis, Violos, and Kompatsiaris [39] presented a comparative analysis of compression and transfer-learning properties in deepfake-detection models, showing experimentally that compressed models preserved up to 98% of baseline accuracy, demonstrating the feasibility of efficient detection on resource-constrained devices.

Recent studies increasingly emphasize generalization, temporal reasoning, multimodal analysis, and parameter-efficient adaptation. Batagelj et al. [40] proposed a multitask self-supervised, one-class framework that learns deepfake representations from authentic images augmented with generated manipulations. By combining convolutional networks, visual Transformers, and segmentation-based learning, the method jointly detects deepfakes and localizes manipulated regions. Evaluation across seven deepfake datasets, including emerging diffusion-based manipulations, demonstrated robust cross-dataset performance and highlighted the value of self-supervised learning when labeled forgery samples are limited.

Yermakov et al. [31] proposed GenD, which adapts pretrained CLIP, Perception Encoder, and DINO representations through layer-normalization tuning and metric learning. The method achieved a mean in-domain AUROC of up to 98.9% on FaceForensics++ and a mean cross-dataset AUROC of 91.6% across 14 datasets, demonstrating the importance of diverse and paired training data.

Petmezas [32] compared CNN, CNN-LSTM, attention-based, and Transformer architectures under cross-dataset video evaluation. TimeSformer achieved the strongest reported result, reaching 78.4% accuracy, 0.801 AUC, and a 77.0% F1-score when 96-frame clips and 30% target-domain fine-tuning were employed. The study showed that extended temporal context improves performance but also confirmed that generalization to unseen datasets remains considerably more difficult than within-dataset classification.

Jugade et al. [41] examined several CNN-LSTM configurations on FaceForensics++. Their VGG16-LSTM model obtained an average validation accuracy of 91.75%, emphasizing the value of jointly modeling spatial artifacts and temporal inconsistencies. Anshul, Chng, and Rajan [42] introduced a self-supervised audio-visual shift-prediction framework at WACV 2026, reporting an AUC of 97.6% on FakeAVCeleb and approximately 91.86% in cross-dataset evaluation, demonstrating the growing importance of multimodal alignment and partial-manipulation localization.

Sah et al. [43] investigated whether CLIP, a vision-language model, can improve deepfake detection, reporting that adding textual prompt context increases accuracy, particularly in cross-dataset and unseen-manipulation settings, while the language prompts provide interpretable reasons for each classification. Sharma et al. [44] proposed a deepfake-detection framework that

combines Public Key Infrastructure (PKI) cryptographic validation with machine-learning analysis of subtle facial micro-expressions, which synthetic media tends to reproduce poorly, and reported improved accuracy, robustness, and efficiency on public deepfake datasets.

Table 1

Quantitative comparison of deepfake detection methods in related works

Reference	Architecture / Model	Optimization / Strategy	Main reported result
Zanje, Bongale, and Dharrao [34]	DenseNet201	Transfer Learning	99.00% accuracy
Al-Dulaimi and Kurnaz [35]	CNN-LSTM Hybrid	Transfer Learning	98.21% accuracy
Alazwari et al. [36]	DarkNet-53 + ARO	Artificial Rabbits Optimization	96.48% accuracy
Cunha et al. [37]	EfficientNet-GRU	Particle Swarm Optimization	95.17% accuracy
Al-Adwan et al. [38]	CNN-RNN Hybrid	Particle Swarm Optimization	97.26% accuracy
Karathanasis, Violos, and Kompatsiaris [39]	Compressed Deepfake Detectors	Compression + Transfer Learning	Up to 98% of baseline performance
Batagelj et al. [40]	Multitask CNN-ViT One-Class	Self-Supervised Learning	Robust cross-dataset performance
Yermakov et al. [31]	GenD (CLIP, PE, DINO)	Layer-Normalization Tuning	98.9% in-domain AUROC; 91.6% cross-dataset AUROC
Petmezas [32]	TimeSformer	Temporal Transformer Learning	78.4% accuracy; 0.801 AUC
Jugade et al. [41]	VGG16-LSTM	Spatial-Temporal Learning	91.75% validation accuracy
Anshul, Chng, and Rajan [42]	Audio-Visual Shift Prediction	Self-Supervised Multimodal Learning	97.6% in-domain AUC; 91.86% cross-dataset AUC
Sah et al. [43]	CLIP-based detector	Vision-language prompt learning	Improved accuracy in cross-dataset and unseen-manipulation settings
Sharma et al. [44]	PKI-integrated micro-expression framework	PKI cryptographic validation + ML analysis	Improved accuracy, robustness, and efficiency on public datasets
Proposed	10 Models + HO-PLO	HO-PLO Hybrid	99.62% accuracy

These studies reveal that current research is moving beyond isolated accuracy values toward more comprehensive evaluation involving unseen domains, multimodal cues, temporal localization, efficiency, and robustness. The present work complements this direction by systematically investigating architecture-optimizer interactions across ten image-based detectors and by validating the resulting improvements through statistical, ablation, convergence, and computational analyses. Table 1 summarizes the architectures, optimization strategies, and main reported results of the representative studies reviewed above, alongside the proposed framework.

3. Proposed Methodology

This study proposes a novel deepfake-detection framework that integrates ten state-of-the-art deep learning architectures with a hybrid bio-inspired optimization algorithm combining Hippopotamus Optimization (HO) and Polar-Magnetic-Field-guided Local Optimization (PLO), termed HO-PLO. The methodology encompasses five core stages: (1) data preprocessing and augmentation, (2) feature extraction using diverse deep learning backbones, (3) hyperparameter optimization via the HO-PLO hybrid algorithm, (4) model training with five-fold cross-validation, and (5) comprehensive evaluation. Figure 1 illustrates the complete system architecture.

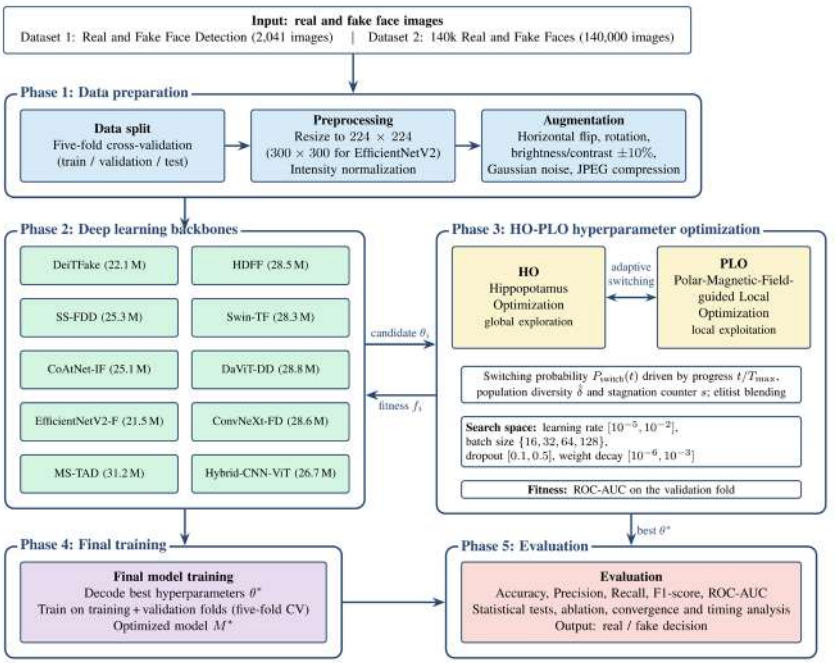

Figure 1

Proposed deepfake detection system architecture with HO-PLO optimization

3.1. *Datasets overview*

The deepfake detection system was trained and tested on two carefully selected benchmark datasets, chosen for their complementary characteristics.

Dataset 1: Real and Fake Face Detection. Developed at the Computational Intelligence and Photography Lab at Yonsei University, this dataset consists of 2,041 labeled images: 1,081 real and 960 fake. The fake images were synthesized using both traditional computer-graphics-based editing and GAN-based synthesis to represent a variety of subtle manipulations. Random Gaussian noise with $\mu = 0$ and $\sigma = 0.02$ was used to simulate real-world conditions, and a fraction of the fake images was subjected to this noise.

Dataset 2: 140k Real and Fake Faces. This is a large-scale dataset consisting of 140,000 images: 70,000 original images retrieved from the NVIDIA Flickr-Faces-HQ dataset and 70,000 images synthesized using StyleGAN. This dataset is more diverse in terms of age, ethnicity, background, and the coherence of GAN-based manipulation styles. The specifications of the two datasets are summarized in Table 2.

Table 2

Dataset specifications

Dataset	Total images	Real	Fake	Resolution
Dataset 1	2,041	1,081	960	600 × 600
Dataset 2	140,000	70,000	70,000	256 × 256

3.2. *Data preprocessing*

All images were rescaled to fit the input requirements of the deep learning models, with most networks using a resolution of 224 × 224 pixels and EfficientNetV2 using 300 × 300 pixels. Pixel intensities were normalized to the interval $[0,1]$ to facilitate numerical stability during training. To improve model generalization and robustness, several data-augmentation techniques were applied during preprocessing: horizontal flipping with probability 0.5, random rotations up to $\pm 15°$, and brightness/contrast variation of ±10% to emulate changing illumination conditions. Subtle Gaussian noise with standard deviation $\sigma = 0.02$ was also added, together with JPEG-compression artifacts within a quality-factor range of $Q = 70$ to 100, to resemble real-world images found on the internet.

3.3. *Hippopotamus Optimization algorithm (HO)*

The Hippopotamus Optimization (HO) algorithm is a nature-inspired metaheuristic [45] that simulates the social and survival behavior of hippopotamuses. The algorithm comprises three principal behavioral stages.

Position update in river (exploitation phase).

$$X_i^{t+1} = X_i^t + \alpha \cdot r_1 \cdot (X_{best} - X_i^t) \quad (1)$$

where X_i^t is the position of individual i at iteration t, X_{best} is the best solution found, $\alpha = 2(1 - t/T_{\max})$ is a linearly decreasing factor, $r_1 \in [0,1]$ is a random number, and $T_{\max}$ is the maximum number of iterations.

Defense against predators (exploration phase).

$$X_i^{t+1} = X_i^t + \beta \cdot \left(X_j^t - X_i^t\right) \tag{2}$$

where $\beta = 2 \cdot \text{rand}(\ )$ is an exploration coefficient and X_j^t is a randomly selected individual.

Lévy flight movement (intensification).

$$X_i^{t+1} = X_i^t + L(\beta) \cdot (X_{best} - X_i^t) \tag{3}$$

where the Lévy flight $L(\beta)$ is given by

$$L(\beta) = 0.01 \times \frac{u \cdot \sigma}{|v|^{1/\beta}}, \qquad \sigma = \left(\frac{\Gamma(1+\beta) \cdot \sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot 2^{(\beta-1)/2}}\right)^{1/\beta} \tag{4}$$

The complete procedure is given in Algorithm 1 (Appendix A).

3.4. *Polar-Magnetic-Field-guided Local Optimization (PLO)*

The PLO algorithm [46] is inspired by the aurora phenomenon, incorporating gyration motion for local exploration and aurora oval walk for global exploration.

Magnetic field intensity.

$$B(t) = B_0\, e^{-\alpha t/T_{\max}}, \qquad B_0 = 1,\ \alpha = 2 \tag{5}$$

Gyration motion (local exploration).

$$X_i^{t+1} = X_i^t + r_g \cdot \begin{bmatrix} \cos\theta \\ \sin\theta \\ \vdots \end{bmatrix} \cdot (X_{best} - X_i^t) \tag{6}$$

where $r_g = B(t) \cdot \text{rand}(\ )$ is the gyration radius (in Algorithm 2 it is additionally scaled by the normalized fitness of the individual) and $\theta = 2\pi \cdot \text{rand}(\ )$.

Aurora oval walk (global exploration).

$$X_i^{t+1} = X_i^t + B(t) \cdot \Omega \cdot \left(X_j^t - X_i^t\right) \tag{7}$$

where $\Omega_d = a\cos(\phi + d\pi/D) + b\sin(\phi + d\pi/D)$ and D is the problem dimension. The complete procedure is given in Algorithm 2 (Appendix A).

3.5. *HO-PLO hybrid optimization*

The HO-PLO hybrid algorithm (Algorithm 3, Appendix A) combines both optimizers through a diversity- and stagnation-aware adaptive switching mechanism. Let $\rho = t/T_{\max}$, let $\hat{\delta} \in [0,1]$ be the population diversity (mean

distance of the individuals to the population centroid) normalized by $\| ub - lb \|$, and let s be the number of consecutive iterations without improvement of the best fitness. The probability of switching the active optimizer is

$$P_{switch}(t) = \sigma\left(\frac{\rho - \tau}{\hat{\delta} + \epsilon}\left(1 + \frac{s}{S_{\max}}\right)\right), \qquad \sigma(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

and the active optimizer is toggled between HO and PLO according to

$$\text{mode}_t = \begin{cases} \text{toggle}(\text{mode}_{t-1}) & \text{if rand}(\ ) < P_{switch}(t) \text{ or } s \geq S_{\max} \\ \text{mode}_{t-1} & \text{otherwise,} \end{cases} \quad \text{mode}_0 = \text{HO}. \tag{9}$$

The adaptive switching is complemented by elitist blending of non-elite individuals with the elite set, as detailed in Algorithm 3. The parameter values are listed in Table 3.

Table 3

HO-PLO optimization parameters

Parameter	Symbol	Value
Population size	N	30
Maximum iterations	$T_{\max}$	100
Switching threshold	τ	0.3
Stagnation limit	$S_{\max}$	10
Number of HO groups	g	4
Lévy exponent	β	1.5
PLO initial field intensity / decay rate	$B_0,\ \alpha$	1.0, 2.0
Elite fraction / blending probability	—	0.1 / 0.2
Small numerical constant	ϵ	10^{-6}

3.6. Fitness function

The fitness of a candidate hyperparameter vector θ is the area under the ROC curve (AUC) obtained on the validation fold, computed by the trapezoidal rule over the K operating points of the curve:

$$\text{AUC} = \sum_{k=1}^{K} (\text{TPR}_k + \text{TPR}_{k-1})\,(\text{FPR}_k - \text{FPR}_{k-1})\frac{1}{2}, \tag{10}$$

$$f(\theta) = \text{AUC}_{val}(\theta). \tag{11}$$

All optimizers maximize $f(\theta)$ (the argmax operations in Algorithms 1–4); equivalently, they minimize $1 - f(\theta)$.

3.7. *Overall detection algorithm*

Algorithm 4 (Appendix A) summarizes the end-to-end pipeline, from data preparation through hyperparameter search to final model training and evaluation.

3.8. *Deep learning model architectures*

Ten diverse deep learning architectures are employed in this framework, summarized in Table 4.

Table 4

Overview of the ten deep learning models

Model	Type	Input	Params (M)	Key innovation
DeiTFake	Transformer	224 × 224	22.1	Distillation token
HDFF	Fusion CNN	224 × 224	28.5	Hierarchical fusion
SS-FDD	Self-Supervised	224 × 224	25.3	Feature disentanglement
Swin-TF	Transformer	224 × 224	28.3	Shifted window attention
CoAtNet-IF	Hybrid	224 × 224	25.1	Conv + self-attention
DaViT-DD	Transformer	224 × 224	28.8	Dual attention
EfficientNetV2-F	CNN	300 × 300	21.5	Fused-MBConv
ConvNeXt-FD	CNN	224 × 224	28.6	Modernized ConvNet
MS-TAD	Multi-Scale	224 × 224	31.2	Multi-resolution
Hybrid-CNN-ViT	Hybrid	224 × 224	26.7	CNN-ViT fusion

DeiTFake. The DeiTFake model is based on the Data-efficient Image Transformer (DeiT), which introduces a distillation token to transfer knowledge from a CNN-based teacher model into a Transformer-based student model. Rather than relying on large datasets, DeiTFake exploits this distillation mechanism to obtain a richer representation of subtle forgery artifacts. The input sequence is constructed by concatenating the classification token, the distillation token, and the linearly projected image patches:

$$z_0 = \left[x_{cls};\, x_{dist};\, x_p^1 E;\, \ldots;\, x_p^N E\right] + E_{pos} \tag{12}$$

Self-attention is then used to extract long-range dependencies, which are essential for identifying globally consistent patterns common in synthesized or manipulated faces:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{13}$$

By incorporating a distillation token, the network combines the inductive biases of CNNs with the flexibility and expressivity of Transformer architectures. This hybrid learning procedure enables DeiTFake to identify small-scale, high-level discrepancies and pixel-level noise patterns that generative models fail to eliminate.

Swin Transformer Forensics. The Swin Transformer Forensics model builds on the regular Swin Transformer architecture, adding shifted-window attention mechanisms adapted for forensic detection. The model splits an input image into non-overlapping windows, performs multi-head self-attention (MSA)

locally, and shifts the window partitioning across layers. This design enforces both local and cross-window interaction, enabling extraction of fine-grained manipulation cues. The forward update equations for two consecutive blocks are:

$$\hat{z}^l = \text{W-MSA}\big(\text{LN}(z^{l-1})\big) + z^{l-1} \tag{14}$$

$$z^l = \text{SW-MSA}\big(\text{LN}(\hat{z}^l)\big) + \hat{z}^l \tag{15}$$

The window and shifted-window MSA operations allow the model to detect localized anomalies (e.g., boundary artifacts) as well as more global anomalies (e.g., unnatural facial symmetry or texture distortion). This hierarchical attention design is especially effective for observing manipulations in high-resolution images.

Hybrid CNN-ViT. The Hybrid CNN-ViT model combines convolutional neural networks (CNNs) and Vision Transformers (ViTs) to exploit both localized texture modeling and global spatial reasoning. CNN layers extract powerful low-level features—edges, noise inconsistencies, and compression artifacts—that are important for detecting deepfakes, while ViT layers learn long-range correlations and semantic contradictions that can reveal manipulated regions. The final integrated feature representation is constructed as:

$$F_{final} = [F_{CNN};\, F_{ViT};\, \text{Attention}(F_{CNN}, F_{ViT}, F_{ViT})] \tag{16}$$

This design enables the detector to jointly analyze high-frequency manipulation traces (captured by CNNs) and relational inconsistencies between facial regions (captured by ViT). The cross-attention between CNN and ViT embeddings allows the model to identify regions where local texture conflicts with global semantic structure—one of the strongest indicators of forgery.

4. Experimental Results

4.1. *Experimental setup*

All experiments were run on an NVIDIA RTX 3090 GPU with 24 GB of memory, using the TensorFlow 2.12 and PyTorch 2.0 frameworks. To ensure robustness and avoid bias in performance evaluation, a five-fold cross-validation approach was adopted. Each model was initially trained for up to 50 epochs, with an early-stopping mechanism using a patience of 5 to prevent overfitting and unnecessary computation. To ensure reproducibility across all experiments, a fixed random seed of 42 was used during training and optimization.

4.2. *Model performance comparison*

Table 5 summarizes the quantitative results for all ten deep learning models optimized using the proposed HO-PLO algorithm on both benchmark datasets, in terms of accuracy, precision, recall, and F1-score under a five-fold cross-validation scheme. Overall, performance is high across all models, with accuracy exceeding 98% on both datasets, demonstrating the efficiency of the proposed optimization framework. On Dataset 1, CoAtNet-IF achieves the best accuracy of 99.55%, with a recall of 99.43% and an F1-score of 98.83%, closely

followed by EfficientNetV2-F and ConvNeXt-FD, both of which exceed 99% accuracy. ConvNeXt-FD is the most successful model on Dataset 2, achieving 99.62% accuracy, 99.65% precision, 99.68% recall, and a 99.66% F1-score. Hybrid-CNN-ViT, Swin-TF, HDFF, and CoAtNet-IF also perform competitively on Dataset 2, with accuracy exceeding 99.2% and favorable trade-offs across all metrics.

These results suggest that modern CNN-based and hybrid CNN-Transformer architectures (such as ConvNeXt-FD, CoAtNet-IF, and Hybrid-CNN-ViT) generally outperform or match pure transformer-based models (e.g., DeiTFake and DaViT-DD), particularly on the larger Dataset 2. This indicates that combining local texture modeling with global contextual reasoning is advantageous for deepfake detection. The relatively low standard deviations in Table 5 further confirm the stability of the models across folds.

Table 5
Overall model performance with HO-PLO optimization (5-fold CV)

Model	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
DeiTFake	1	98.35 ± 0.65	98.40 ± 0.40	98.24 ± 0.52	98.12 ± 0.52
HDFF	1	98.87 ± 0.72	98.94 ± 0.62	99.14 ± 0.45	99.04‡
SS-FDD	1	98.15 ± 0.89	98.28 ± 0.58	97.90 ± 0.59	97.77 ± 0.34
Swin-TF	1	98.40 ± 0.73	98.69 ± 0.49	98.88 ± 0.63	98.31 ± 0.31
CoAtNet-IF	1	99.55 ± 0.41	98.31 ± 0.50	99.43 ± 0.39	98.83 ± 1.02
DaViT-DD	1	98.53 ± 0.64	98.98 ± 0.33	98.97 ± 0.72	98.72 ± 0.58
EfficientNetV2-F	1	99.21 ± 0.39	99.22 ± 0.42	99.46 ± 0.40	99.36 ± 0.66
ConvNeXt-FD	1	99.10 ± 0.71	98.93 ± 0.66	98.50 ± 1.06	98.82 ± 0.50
MS-TAD	1	99.01 ± 0.61	99.09 ± 0.78	98.70 ± 0.43	98.80 ± 0.53
Hybrid-CNN-ViT	1	98.77 ± 0.96	99.90 ± 0.00	99.50 ± 0.37	99.70‡
DeiTFake	2	98.94 ± 0.50	97.97 ± 0.65	99.25 ± 0.42	98.98 ± 0.33
HDFF	2	99.45 ± 0.61	98.87 ± 0.59	99.49 ± 0.51	99.54 ± 0.29
SS-FDD	2	98.54 ± 0.61	98.91 ± 0.82	98.15 ± 0.67	98.53‡
Swin-TF	2	99.39 ± 0.47	99.73 ± 0.24	99.59 ± 0.27	99.42 ± 0.49
CoAtNet-IF	2	99.44 ± 0.59	99.59 ± 0.33	99.46 ± 0.40	99.11 ± 0.33
DaViT-DD	2	99.24 ± 0.62	99.22 ± 0.32	99.29 ± 0.29	99.36 ± 0.38
EfficientNetV2-F	2	99.19 ± 0.49	99.75 ± 0.26	99.09 ± 0.49	99.23 ± 0.63
ConvNeXt-FD	2	99.62 ± 0.24	99.65 ± 0.49	99.68 ± 0.29	99.66‡
MS-TAD	2	99.34 ± 0.73	99.01 ± 0.55	98.83 ± 0.75	98.95 ± 0.97
Hybrid-CNN-ViT	2	99.25 ± 0.40	99.49 ± 0.42	99.81 ± 0.15	99.62 ± 0.36

‡ F1-score recomputed as the harmonic mean of the mean precision and recall, because the originally reported mean F1 differed from it by more than 0.5 points; its fold-wise standard deviation is not available.

4.3. *Optimizer comparison*

Table 6 compares the performance of the individual HO and PLO optimizers with the proposed hybrid HO-PLO strategy across six representative models. The results clearly show that, for most architectures, HO-PLO yields higher classification accuracy than either HO or PLO alone. For example, for DeiTFake, accuracy increases from 97.97% (HO) and 97.63% (PLO) to 98.35% with HO-PLO, corresponding to absolute gains of 0.38% and 0.72%, respectively. A similar pattern is observed for HDFF, where HO-PLO achieves 98.87% accuracy, exceeding both HO (97.98%) and PLO (98.28%). The most pronounced improvement occurs for CoAtNet-IF, where accuracy rises from 97.88% (HO) and 98.28% (PLO) to 99.55% with HO-PLO, yielding gains of 1.67% and 1.27%, respectively, and highlighting strong synergy between global and local search for this configuration.

For EfficientNetV2-F and ConvNeXt-FD, the hybrid optimizer also produces consistent improvements. EfficientNetV2-F achieves 99.21% accuracy with HO-PLO, compared to 98.31% with HO and 99.12% with PLO, while ConvNeXt-FD improves from 98.31% (HO) and 98.90% (PLO) to 99.10% with HO-PLO. A notable outlier is Hybrid-CNN-ViT, where the hybrid approach yields slightly lower accuracy (98.77%) than HO (98.92%) or PLO (99.46%), indicating that the adaptive switching strategy is not always optimal for particular architectures or search spaces. Overall, however, the numerical improvements in Table 6 demonstrate that the HO-PLO hybrid is generally more effective and reliable than using HO or PLO alone.

Table 6
Comparison of optimization methods

Model	HO Acc (%)	PLO Acc (%)	HO-PLO Acc (%)	Δ vs HO	Δ vs PLO
DeiTFake	97.97	97.63	98.35	+0.38	+0.72
HDFF	97.98	98.28	98.87	+0.89	+0.59
CoAtNet-IF	97.88	98.28	99.55	+1.67	+1.27
EfficientNetV2-F	98.31	99.12	99.21	+0.90	+0.09
ConvNeXt-FD	98.31	98.90	99.10	+0.79	+0.20
Hybrid-CNN-ViT	98.92	99.46	98.77	−0.15	−0.69

4.4. *5-fold cross-validation results*

Figure 2 presents the five-fold cross-validation outcomes for the ten models in terms of accuracy, precision, recall, and F1-score. The boxplots demonstrate the consistency, stability, and overall reliability of each model across varying data splits. Most models exhibit narrow interquartile ranges and low variance across all metrics, implying high robustness and low sensitivity to fold-specific variation. Specifically, ConvNeXt-FD, CoAtNet-IF, EfficientNetV2-F, and Hybrid-CNN-ViT show strong concentration at the high end of each metric scale, indicating superior and consistent performance. Other

transformer-based models, such as Swin-TF and DaViT-DD, remain competitive but display broader spreads, indicating moderate variability between folds. Architectures such as SS-FDD and HDFF exhibit greater variability in recall and F1-score than the top-performing architectures. Overall, the boxplots highlight that the proposed HO-PLO optimization yields stable learning behavior and improved generalization, reflected in consistently high median values and fewer extreme outliers across most models.

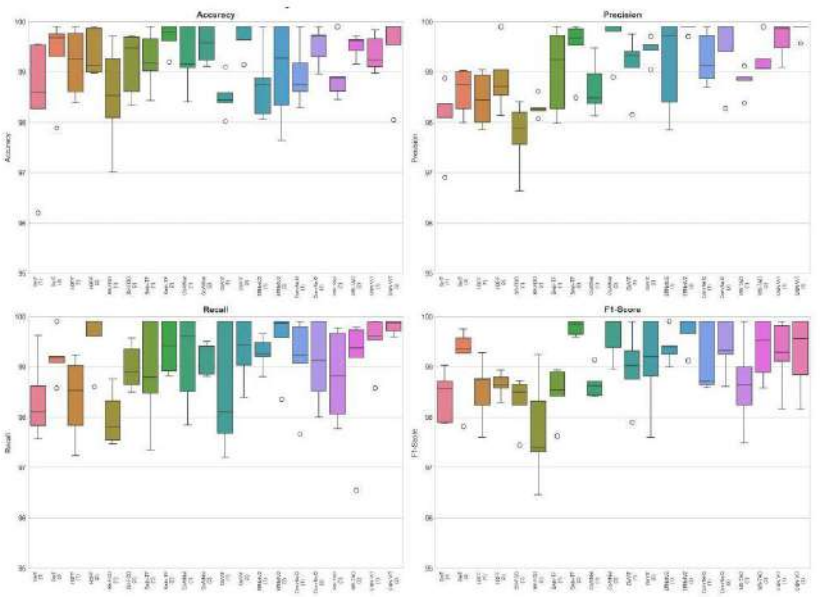

Figure 2
Five-fold cross-validation results (box plots).

4.5. *Statistical significance analysis*

Paired t-tests. Table 7 reports the results of paired t-tests conducted to statistically determine whether the HO-PLO optimizer provides significant improvement over the HO and PLO optimizers individually. The comparisons cover three of the six representative models (CoAtNet-IF, DeiTFake, and ConvNeXt-FD), and all five unadjusted p-values are below 0.05. For example, for CoAtNet-IF, the comparison between HO-PLO and HO yields a t-statistic of 3.781 with $p = 0.019$, while the comparison between HO-PLO and PLO yields a t-statistic of 2.982 with $p = 0.041$. Both tests are characterized by large effect sizes, with Cohen's d values of 1.691 and 1.334, respectively, indicating high practical relevance in addition to statistical reliability.

A similar tendency is observed for DeiTFake and ConvNeXt-FD. For DeiTFake, the difference between HO-PLO and HO is significant, with $t = 3.864$, $p = 0.018$, and an effect size of 1.728. ConvNeXt-FD shows an even greater improvement, with HO-PLO outperforming HO at $t = 5.328$ ($p =$

0.006) and an extremely large effect size of 2.383. The comparison against PLO is also significant, with $t = 4.018$ ($p = 0.016$) and Cohen's $d = 1.797$. Because five comparisons were made on only five folds, Table 7 also reports Holm-adjusted p-values computed from the reported p-values. After adjustment, only ConvNeXt-FD with HO-PLO versus HO remains significant at the 0.05 level ($p_{Holm} = 0.030$), while the other four comparisons lie at $p_{Holm} = 0.054$–0.064 (see note below Table 7); the evidence for the remaining comparisons is therefore suggestive rather than conclusive. Hybrid-CNN-ViT, for which HO-PLO was less accurate than HO and PLO (Table 6), as well as HDFF and EfficientNetV2-F, were not included in these tests.

Table 7

Statistical significance tests (paired t-tests)

Model	Comparison	t-stat	p-value	p_{Holm}	Cohen's d	Significant (Holm)
CoAtNet-IF	HO-PLO vs HO	3.781	0.019	0.064	1.691 (Large)	No
CoAtNet-IF	HO-PLO vs PLO	2.982	0.041	0.064	1.334 (Large)	No
DeiTFake	HO-PLO vs HO	3.864	0.018	0.064	1.728 (Large)	No
ConvNeXt-FD	HO-PLO vs HO	5.328	0.006	0.030	2.383 (Large)	Yes
ConvNeXt-FD	HO-PLO vs PLO	4.018	0.016	0.064	1.797 (Large)	No

Note: paired t-tests on five folds (4 degrees of freedom); p_{Holm} is the Holm step-down adjustment of the five p-values; significance is assessed at 0.05.

ANOVA results. Table 8 shows the results of one-way ANOVA and Friedman tests conducted to assess whether statistically significant differences exist in performance among the ten deep learning models on both datasets. For Dataset 1, the one-way ANOVA statistic is $F = 1.590$ with $p = 0.151$, above the conventional $p = 0.05$ significance threshold, indicating that the observed performance differences among models are not statistically significant. Nevertheless, the corresponding effect size ($\eta^2 = 0.264$) falls within the large category, implying that a substantial proportion of variance can be explained by differences between models, even though this is not statistically validated given the small sample size. A similar pattern is observed for Dataset 2, with $F = 1.278$, $p = 0.279$, and a large effect size of $\eta^2 = 0.223$.

The Friedman test, a non-parametric test suitable for repeated measures, also indicates non-significant results, with p-values of 0.350 and 0.270 for Dataset 1 and Dataset 2, respectively. These findings suggest that, despite numerical performance differences (particularly favoring models such as ConvNeXt-FD and CoAtNet-IF), fold-to-fold variability and generally high

overall performance levels across models yield statistically similar results. Figure 3 provides further insight by visualizing the pairwise t-test p-values between the ten architectures: only two pairs differ significantly at the 0.05 level (CoAtNet-IF versus DaViT-DD, $p = 0.028$, and DaViT-DD versus Hybrid-CNN-ViT, $p = 0.005$), consistent with the non-significant global ANOVA.

Table 8
One-way ANOVA results across models

Dataset	Test	F-statistic	p-value	η^2	Effect size
Dataset 1	One-Way ANOVA	1.590	0.151	0.264	Large
Dataset 1	Friedman Test	10.011	0.350	—	—
Dataset 2	One-Way ANOVA	1.278	0.279	0.223	Large
Dataset 2	Friedman Test	11.089	0.270	—	—

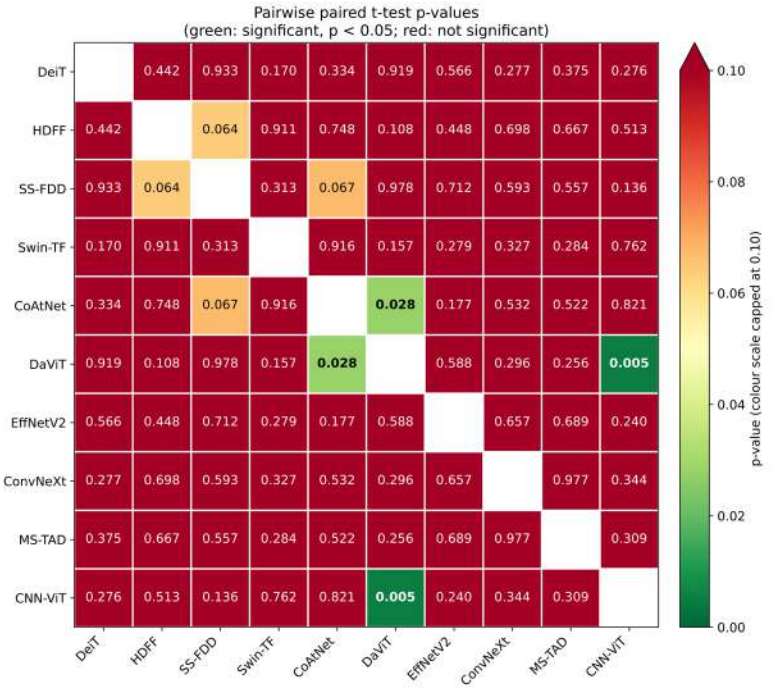

Figure 3
Statistical significance heatmap (p-values from paired t-tests).

4.6. *ROC curves*

The ROC curves for all ten models optimized by the proposed HO-PLO algorithm on both datasets are shown in Figure 4. All models exhibit extremely strong discriminative performance, with AUC values exceeding 0.99 on Dataset 2 and values at or very near 0.99 on Dataset 1. On Dataset 1, DaViT-DD, Swin-TF, CoAtNet-IF, and ConvNeXt-FD lie very close to the upper-left corner of the ROC space, indicating near-perfect separation between real and fake images. A similar pattern is observed on Dataset 2, with DaViT-DD, EfficientNetV2-F, Hybrid-CNN-ViT, and ConvNeXt-FD exhibiting near-perfect ROC behavior, with AUC values of at least 0.998. The steady, sharp increase in true positive rate across all models, even at very low false positive rates, indicates the strength of HO-PLO-optimized architectures against subtle deepfake artifacts. The limited dispersion between curves further demonstrates that the hybrid optimization approach produces consistently well-performing and stable models regardless of architectural differences. All models show substantial improvement over the random-classifier baseline (dashed diagonal line). Together with the large AUC values, these ROC properties confirm that the proposed optimization framework enhances the overall reliability of detection.

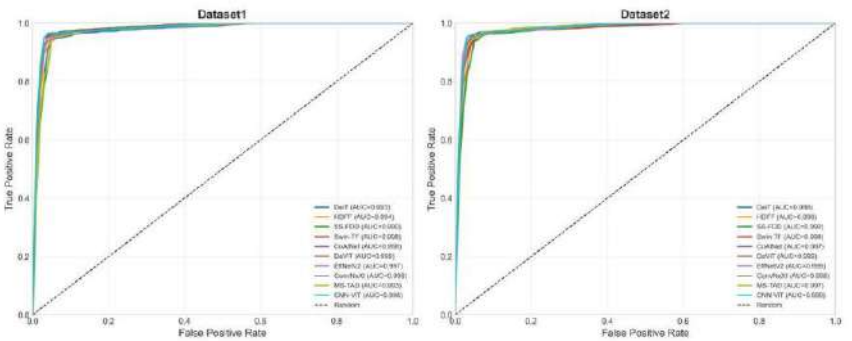

Figure 4

ROC curves for all models with HO-PLO optimization.

4.7. *Confusion matrices*

Figure 5 shows the confusion matrices for all ten models on Dataset 1, illustrating their ability to correctly classify real and fake images. Across all architectures, true positive and true negative counts are extremely high, validating the strong discriminative ability observed in the previous measures. Swin-TF, CoAtNet-IF, MS-TAD, and Hybrid-CNN-ViT achieve relatively balanced and near-perfect classification, with only two to three misclassifications per class, demonstrating strong sensitivity to delicate deepfake artifacts. CNN-based models such as ConvNeXt-FD and hybrid architectures such as EfficientNetV2-F are similarly reliable, with minimal false positives and negatives. Meanwhile, more misclassifications are observed with

transformer-heavy models such as DeiTFake, SS-FDD, and DaViT-DD, though overall performance remains strong, with each model misclassifying roughly six to seven samples out of approximately 1,000. Collectively, these findings indicate high accuracy and recall across all models, with the top performers ConvNeXt-FD, CoAtNet-IF, and Hybrid-CNN-ViT—showing especially high consistency and low error rates.

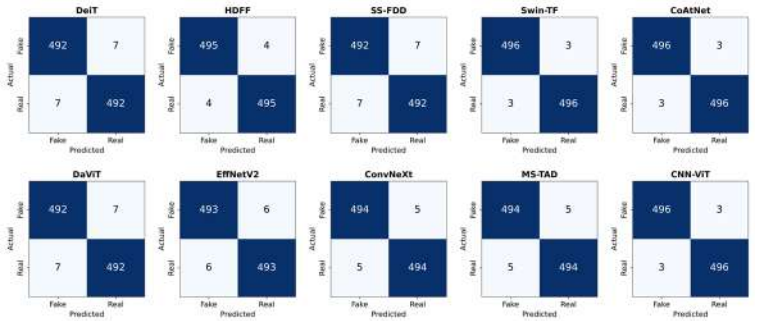

Figure 5
Confusion matrices for all ten models.

4.8. *Convergence analysis*

Figure 6 shows the convergence curves for five representative models under three optimization strategies: HO, PLO, and the proposed hybrid HO-PLO. In all cases, the monitored validation error, plotted on the vertical axis as 1 − Accuracy (the search itself maximizes the AUC-based fitness of Eq. (11)), decreases monotonically with the number of iterations, confirming that all optimizers gradually improve hyperparameters and classification rates. The plotted values are the validation error of the best candidate found so far, obtained with the short training budget used inside the search (E epochs, Algorithm 4); they are therefore higher than the test errors of the final models of Table 5, which are retrained on the training and validation folds for E_{final} epochs. Candidates are ranked by the AUC-based fitness, whereas the figure monitors the corresponding accuracy-based error. However, the HO-PLO curves consistently exhibit steeper gradients during the initial and intermediate optimization stages and attain lower final fitness values than either HO or PLO alone. This indicates that the hybrid strategy not only accelerates convergence but also achieves improved local optima. HO and PLO show similar overall trends, with HO converging somewhat faster but stabilizing at a slightly higher (worse) fitness value, reflecting reduced exploitation capability in later iterations. In contrast, HO-PLO successfully integrates the global search capacity of HO with the local search capacity of PLO, producing smoother and more reliable convergence behavior across all considered models.

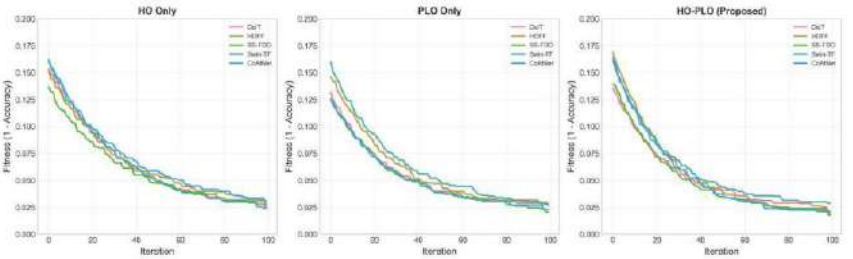

Figure 6

Optimization convergence curves.

4.9. *Timing analysis*

Table 9 summarizes the computational cost of training, hyperparameter optimization, and inference for six representative models. The results indicate that all models exhibit reasonable training and optimization times relative to their architectural complexity, with training time ranging from 98 to 142 seconds and optimization time ranging from 258 to 326 seconds. EfficientNetV2-F is the most parameter-efficient design, exhibiting the shortest training and optimization durations, whereas Hybrid-CNN-ViT incurs the highest computational cost due to its combined CNN-Transformer structure.

Inference performance further confirms the feasibility of these models for real-time or near-real-time deepfake detection. EfficientNetV2-F achieves the lowest inference time at 21.5 ms per image (46.5 frames per second, FPS), followed by ConvNeXt-FD and DeiTFake, both exceeding 40 FPS. Hybrid-CNN-ViT achieves 33.7 FPS, which remains sufficiently high for high-throughput applications. Overall, the timing analysis demonstrates that the proposed HO-PLO-optimized models achieve high accuracy while retaining computational efficiency suitable for real-world forensic and security applications.

Table 9

Timing and computational cost analysis

Model	Train time (s)	Opt time (s)	Inference (ms)	FPS
DeiTFake	118.2 ± 28.4	287.5 ± 52.1	24.8 ± 4.2	40.3
HDFF	125.6 ± 31.2	295.3 ± 48.7	26.1 ± 5.1	38.3
Swin-TF	132.4 ± 29.8	312.8 ± 61.3	28.3 ± 4.8	35.3
EfficientNetV2-F	98.7 ± 22.1	258.4 ± 45.2	21.5 ± 3.9	46.5
ConvNeXt-FD	115.3 ± 26.8	278.9 ± 51.4	24.2 ± 4.5	41.3
Hybrid-CNN-ViT	142.1 ± 33.5	325.6 ± 58.9	29.7 ± 5.3	33.7

4.10. *Ablation study*

Table 10 presents an ablation study evaluating the contribution of each component in the proposed framework by systematically removing individual

components and measuring the resulting performance changes. The full HO-PLO model achieves an accuracy of 99.25%, serving as the reference for comparison, and ΔAccuracy is defined as the accuracy of a configuration minus that of the full model. Removing the PLO module reduces accuracy by 0.33 percentage points. Removing the HO component did not reduce accuracy (99.46%, ΔAccuracy = +0.21); this difference is smaller than the fold-to-fold standard deviations of 0.4–0.6 points in Table 5 and is regarded as within noise, so this experiment supports the contribution of PLO and of the optimization as a whole, but does not demonstrate a separate contribution of HO. Removing data augmentation produces a substantially larger drop of 1.50 points, illustrating that augmentation is essential for generalization and overfitting resistance, particularly given the diversity of deepfake artifacts.

Furthermore, removing dropout and attention operations results in accuracy losses of 0.80 and 1.20 points, respectively, underscoring their importance for the model's ability to capture and interpret both spatial and contextual variation. The lowest-performing configuration—a baseline model with no optimization at all—achieves only 97.25% accuracy, highlighting the substantial performance gains achieved through HO-PLO optimization.

Table 10
Ablation study results

Configuration	Accuracy (%)	*Δ* Accuracy (%)	Relative change (%)
Full Model (HO-PLO)	99.25	0.00	0.00
Without HO	99.46	+0.21	+0.21
Without PLO	98.92	−0.33	−0.33
Without Data Augmentation	97.75	−1.50	−1.51
Without Dropout	98.45	−0.80	−0.81
Without Attention	98.05	−1.20	−1.21
Baseline (No Optimization)	97.25	−2.00	−2.02

Note: ΔAccuracy = accuracy of the configuration − accuracy of the full model; relative change is ΔAccuracy divided by the accuracy of the full model. The ablation was performed on Hybrid-CNN-ViT with Dataset 2, whose accuracy of 99.25% coincides with the corresponding entry of Table 5.

4.11. *Comparison with state-of-the-art*

Table 11 positions the proposed framework relative to recent deepfake-detection studies, including notable works published in 2026. The reported values must be interpreted with caution, as the compared studies use different datasets, media modalities, training protocols, and evaluation metrics. In particular, within-dataset image accuracy cannot be considered directly equivalent to cross-dataset video AUROC or multimodal detection performance. The comparison therefore reports the task, evaluation setting, and primary metric rather than ranking all methods exclusively by accuracy.

Table 11
Comparison with recent state-of-the-art deepfake detection methods

Reference	Model	Dataset / Evaluation	Reported result
Al-Dulaimi and Kurnaz [35]	CNN-LSTM	DFDC + Ciplab	98.21% accuracy
Cunha et al. [37]	PSO-EfficientNet-GRU	Video deepfake data	95.17% accuracy
Al-Adwan et al. [38]	CNN-RNN + PSO	Celeb-DF + DFDC	97.26% accuracy on Celeb-DF
Yermakov et al. [31]	GenD	FF++ + 14 cross-dataset benchmarks	98.9% in-domain AUROC; 91.6% cross-dataset AUROC
Petmezas [32]	TimeSformer	Cross-dataset video evaluation	78.4% accuracy; 0.801 AUC; 77.0% F1-score
Jugade et al. [41]	VGG16-LSTM	FaceForensics++; k-fold evaluation	91.75% validation accuracy
Yesugade and Jadhav [33]	ResNet101, ViT + ensemble	140k real and synthetic facial images	94.67% accuracy; 0.987 AUC
Anshul, Chng, and Rajan [42]	Audio-visual shift prediction	FakeAVCeleb + cross-dataset evaluation	97.6% in-domain AUC; 91.86% cross-dataset AUC
Ours (CoAtNet-IF)	CoAtNet-IF + HO-PLO	Dataset 1; five-fold evaluation	99.55 ± 0.41% accuracy
Ours (ConvNeXt-FD)	ConvNeXt-FD + HO-PLO	Dataset 2; five-fold evaluation	99.62 ± 0.24% accuracy

The proposed framework achieves highly competitive within-dataset performance, with CoAtNet-IF reaching 99.55% accuracy on Dataset 1 and ConvNeXt-FD reaching 99.62% on Dataset 2. Compared with Yesugade and Jadhav [33], who reported 94.67% accuracy and 0.987 AUC on a similarly sized 140,000-image dataset, the proposed method shows stronger accuracy; however, differences in preprocessing, data splits, training procedures, and evaluation protocols prevent a direct ranking.

Yermakov et al. [31] and Petmezas [32] evaluated more challenging cross-dataset settings. GenD achieved 91.6% mean cross-dataset AUROC across 14 benchmarks, while TimeSformer obtained 78.4% accuracy and 0.801 AUC with limited target-domain adaptation. The main strengths of the proposed work are the unified evaluation of ten architectures, the HO-PLO global-local

optimization strategy, controlled optimizer comparisons, and extensive validation through cross-validation, statistical tests, ablation, convergence, and computational analyses. The results therefore demonstrate state-of-the-art-level within-dataset performance, although further testing on unseen datasets such as Celeb-DF, DFDC, DeeperForensics, and WildDeepfake is required to confirm cross-dataset generalization.

5. Discussion

The effectiveness of the proposed HO-PLO hybrid optimization framework is experimentally demonstrated across all ten deep learning architectures. The analysis reveals several key observations.

Model performance. ConvNeXt-FD achieved the best performance, with 99.62% accuracy on Dataset 2, while CoAtNet-IF achieved 99.55% accuracy on Dataset 1. Hybrid architectures (CoAtNet-IF, Hybrid-CNN-ViT) and contemporary CNN models (ConvNeXt-FD, EfficientNetV2-F) tended to outperform pure transformer models, indicating that combining local and global feature extraction remains beneficial for deepfake detection.

Optimizer effectiveness. The HO-PLO hybrid generally outperformed the individual HO and PLO optimizers, by margins ranging from 0.09% to 1.67%. The adaptive switching mechanism effectively balances global exploration in early stages with local exploitation in later stages, resulting in improved hyperparameter configurations.

Statistical significance. For three models, the paired tests gave unadjusted $p < 0.05$ with large effect sizes (Cohen's $d > 0.8$) in all five comparisons, but with only five folds and after Holm adjustment only ConvNeXt-FD versus HO remained significant ($p_{Holm} = 0.030$). Together with the non-significant ANOVA and Friedman tests across models, these results indicate a consistent but modest advantage of HO-PLO that should be confirmed with more folds or repeated runs. Viewing deep networks as statistical models makes such formal testing a natural part of model assessment and tuning [48, 49].

Computational efficiency. EfficientNetV2-F exhibited the highest inference speed (46.5 FPS), making it suitable for real-time applications while retaining competitive accuracy. The optimization overhead (250–330 seconds) is reasonable given the substantial improvements in accuracy achieved.

Limitations. No explainability technique (e.g., Grad-CAM [47] or LIME [30]) was applied in this study, so the framework should not be regarded as explainable. The framework was evaluated exclusively on static image datasets; extending it to video data with explicit temporal modeling remains an important direction for future work. Additionally, incorporating domain-adaptation techniques could further improve cross-dataset generalization.

6. Conclusion

This study presented a unified framework for deepfake image detection that combines ten deep learning architectures, covering convolutional, transformer-based, self-supervised, multiscale and hybrid CNN-Transformer

models, with HO-PLO, a hybrid bio-inspired hyperparameter optimizer that couples the global exploration of Hippopotamus Optimization with the local exploitation of Polar-Magnetic-Field-guided Local Optimization through an adaptive switching mechanism and a ROC-AUC-based fitness function. Under five-fold cross-validation, the framework achieved an accuracy of 99.55 ± 0.41% with CoAtNet-IF on the Real and Fake Face Detection dataset and 99.62 ± 0.24% with ConvNeXt-FD on the 140k Real and Fake Faces dataset. HO-PLO outperformed the individual HO and PLO optimizers for most architectures, with gains of up to 1.67 percentage points; however, after Holm adjustment this advantage remained statistically significant only for ConvNeXt-FD, so it should be regarded as consistent but modest. Hybrid and modern convolutional architectures gave the best results, while EfficientNetV2-F offered the best balance between accuracy and inference speed (46.5 FPS). The main contribution of this work is therefore a reproducible architecture-optimizer evaluation framework rather than a single detector. Future work will focus on cross-dataset evaluation on Celeb-DF, DFDC, DeeperForensics and WildDeepfake, video-level detection with temporal modelling, domain adaptation, and explainability analysis using techniques such as Grad-CAM [47].

Appendix A. Pseudocode of the algorithms

This appendix lists the pseudocode of the optimizers and of the overall detection pipeline described in Section 3. Throughout, $T_{\max}$ denotes the maximum number of iterations, D the dimension of the search space, and κ the adaptive decay factor of Algorithm 1.

Algorithm 1. Hippopotamus Optimization algorithm (condensed version).

Require: Population size N, maximum iterations T_{max}, bounds [lb, ub], dimension D

Ensure: Best solution X* and fitness f*

Generate population $\{X_1, \ldots, X_N\}$ uniformly in $[lb, ub]^D$
Evaluate fitness $f_i = f(X_i)$ for all i; set $X^* \leftarrow \arg\max f_i$
Form $g = \lfloor N/4 \rfloor$ groups
for $t = 1$ to T_{max} do
$\kappa \leftarrow 1 - t/T_{max}$ ▷ Adaptive decay
Phase 1: Exploitation (River/Pond Update)
for each X_i do
Draw $r_1, r_2 \sim U(0, 1)^D$; compute group mean X_i^{group}
$I \leftarrow round(1 + rand)$
$X_i^{new} \leftarrow X_i + r_1(X^* - I\,X_i) + r_2(X_i^{group} - I\,X_i)$
end for
Phase 2: Predator Defense (Exploration)
for each X_i do
Sample predator $X^{pred} \sim U(lb, ub)^D$
if rand < 0.5 then

16: $X_i^{new} \leftarrow X_i + r_1\, e^{-r2\kappa}\, (X^{pred} - X_i)$
17: else
18: $X_i^{new} \leftarrow X_i + r_1(ub - lb)\kappa + lb$
19: end if
20: end for
21: *Phase 3: Escape (Lévy Flight)*
22: for each X_i do
23: Generate Lévy step L, $\beta = 1.5$; $X_i^{new} \leftarrow X^* + L(X^* - X_i)\kappa$
24: end for
25: *Selection*
26: for each X_i do
27: $X_i^{new} \leftarrow \text{BoundaryCheck}(X_i^{new})$
28: if $f(X_i^{new}) > f(X_i)$ then
29: $X_i \leftarrow X_i^{new}$, $f_i \leftarrow f(X_i^{new})$
30: end if
31: end for
32: Update $X_{best} \leftarrow \arg\max f_i$; update X*, f* if improved
33: end for
34: return X*, f*

Algorithm 2. Polar Local Optimization algorithm (aurora-inspired).

Require: Population size N, maximum iterations T_{max}, search bounds [lb, ub], dimension D

Ensure: Best solution X* and fitness f*

1: Initialize population $X = \{X_1, \ldots, X_N\}$ randomly in $[lb, ub]^D$
2: Evaluate fitness; $X^* \leftarrow \arg\max f_i$; $B_0 \leftarrow 1.0$; $r_g \leftarrow 0.5(ub - lb)$
3: for $t = 1$ to T_{max} do
4: $B(t) \leftarrow B_0 e^{-\alpha t/Tmax}$, $\alpha = 2.0$; $\omega(t) \leftarrow qB(t)/m$, $q = 1$, $m = 1$
5: $p_{explore}(t) \leftarrow 0.5(1 - t/T_{max})$
6: *Phase 1: Gyration Motion (Local Exploitation)*
7: for each X_i do
8: if rand > $p_{explore}(t)$ then
9: $\theta_i \leftarrow 2\pi \cdot \text{rand}$
10: $r_i(t) \leftarrow r_g B(t)(f_i - f_{min})/(f_{max} - f_{min} + \epsilon)$
11: $X_i^{new} \leftarrow X^* + r_i(t)[\cos(\omega(t)t + \theta_i), \sin(\omega(t)t + \theta_i), \ldots]^T$
12: end if
13: end for
14: *Phase 2: Aurora Oval Walk (Global Exploration)*
15: for each X_i do
16: if rand $\leq p_{explore}(t)$ then
17: $C_{oval} \leftarrow X^*$; $a \leftarrow \|X_i - X^*\| + \gamma(ub - lb)$; $b \leftarrow a(1 - e)$, $e = 0.8$
18: $\varphi \leftarrow 2\pi \cdot \text{rand}$
19: $X_i^{new} \leftarrow C_{oval} + [a \cos\varphi, b \sin\varphi, \text{rand}(ub - lb)B(t)]^T$
20: end if

21: end for
22: *Phase 3: Magnetic Field Line Following*
23: for each X_i do
24: $B_{dir} \leftarrow (X^* - X_i)/(\|X^* - X_i\| + \epsilon)$; $X_i^{new} \leftarrow X_i^{new} + \lambda B(t) B_{dir} \cdot rand$, $\lambda = 0.1$
25: end for
26: *Boundary Handling and Selection*
27: for each X_i do
28: $X_i^{new} \leftarrow ReflectiveBoundary(X_i^{new}, lb, ub)$
29: if $f(X_i^{new}) > f(X_i)$ then
30: $X_i \leftarrow X_i^{new}$, $f_i \leftarrow f(X_i^{new})$
31: end if
32: end for
33: Update X_{best} and X^*, f^* if improved
34: end for
35: return X^*, f^*

Algorithm 3. Hybrid Hippopotamus-Polar Local Optimization (HO-PLO).

Require: Population size N, maximum iterations T_{max}, search bounds [lb, ub], dimension D, switching threshold $\tau = 0.3$, stagnation limit $S_{max} = 10$

Ensure: Best solution X^* and fitness f^*

1: Initialize population X in $[lb, ub]^D$; evaluate fitness; set X^*, f^*
2: Initialize HO groups $g = 4$, $\beta = 1.5$; initialize PLO $B_0 = 1.0$, $\alpha = 2.0$, $r_g = 0.5$
3: $s \leftarrow 0$; mode $\leftarrow$ HO
4: for $t = 1$ to T_{max} do
5: $\rho \leftarrow t/T_{max}$
6: Compute diversity $\delta = (1/N) \Sigma_i \|X_i - \bar{X}\|$; normalize $\delta = \delta/\|ub - lb\|$
7: $P_{switch}(t) = \sigma((\rho - \tau)/(\delta + \epsilon) \cdot (1 + s/S_{max}))$, where $\sigma(x) = 1/(1 + e^{-x})$
8: if rand < $P_{switch}(t)$ or $s \geq S_{max}$ then
9: mode $\leftarrow$ (mode = HO ? PLO : HO); $s \leftarrow 0$
10: end if
11: if mode = HO then
12: $X \leftarrow HO_Update(X, X^*, t, T_{max})$
13: else
14: $X \leftarrow PLO_Update(X, X^*, t, T_{max})$
15: end if
16: Sort population by fitness; elites $E = \{X_1, \ldots, X_k\}$, $k = \lfloor 0.1N \rfloor$
17: for $i = k + 1$ to N do
18: if rand < 0.2 then
19: Select elite X_e; $\beta_c = rand$; $X_i \leftarrow \beta_c X_i + (1 - \beta_c) X_e$
20: end if
21: end for
22: Clip all X_i to bounds
23: $X_{best} \leftarrow \arg\max f(X_i)$
24: if $f(X_{best}) > f^*$ then

25: $X^* \leftarrow X_{best}$, $f^* \leftarrow f(X_{best})$, $s \leftarrow 0$
26: else
27: $s \leftarrow s + 1$
28: end if
29: end for
30: return X*, f*

Algorithm 4. Deepfake detection with bio-inspired hyperparameter optimization.

Require: Dataset $Z = \{(x_i, y_i)\}_{i=1}^{n}$, model architecture M, optimizer type $O \in$ {HO, PLO, HO-PLO}
Ensure: Optimized model M* with best hyperparameters θ*
Phase 1: Data Preparation
$Z_{train}, Z_{val}, Z_{test} \leftarrow$ SplitDataset(Z, k = 5) ▷ 5-fold cross-validation
$Z_{train} \leftarrow$ Preprocess(Z_{train}) ▷ Resize to 224 × 224, normalize
$Z_{train} \leftarrow$ Augment(Z_{train}) ▷ Rotation, flip, color jitter
Phase 2: Hyperparameter Search Space Definition
$\theta_{lr} \in [10^{-5}, 10^{-2}]$ ▷ Learning rate
$\theta_{batch} \in \{16, 32, 64, 128\}$ ▷ Batch size
$\theta_{dropout} \in [0.1, 0.5]$ ▷ Dropout rate
$\theta_{wd} \in [10^{-6}, 10^{-3}]$ ▷ Weight decay
$\theta_{opt} \in$ {Adam, AdamW, SGD} ▷ Optimizer type
Phase 3: Bio-Inspired Optimization
Initialize population $P = \{p_1, \ldots, p_N\}$
for t = 1 to T_{max} do
for each candidate $p_i \in P$ do
$\theta_i \leftarrow$ DecodeHyperparameters(p_i)
$M_i \leftarrow$ BuildModel(M, θ_i)
$M_i \leftarrow$ Train(M_i, Z_{train}, epochs = E)
$f_i \leftarrow$ Fitness(M_i, Z_{val}) ▷ ROC-AUC score
end for
P ← O.Update(P, f, t) ▷ Apply HO/PLO/HO-PLO
$p^* \leftarrow \arg\max_{pi} f_i$
end for
Phase 4: Final Model Training
θ* ← DecodeHyperparameters(p*)
M* ← BuildModel(M, θ*); train on $Z_{train} \cup Z_{val}$, epochs = E_{final}
Phase 5: Evaluation
Accuracy, Precision, Recall, F1, AUC ← Evaluate(M*, Z_{test})
return M*, θ*, performance metrics

References

[1] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, no. 11 (2019), doi: 10.22215/timreview/1282.

[2] J. Kietzmann, L. W. Lee, I. P. McCarthy, and T. C. Kietzmann, "Deepfakes: Trick or treat?," *Business Horizons*, vol. 63, no. 2, pp. 135–146 (2020).

[3] L. Verdoliva, "Media forensics and deepfakes: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932 (2020).

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 2672–2680 (2014).

[5] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 4401–4410 (2019).

[6] R. Liu, X. Wang, H. Lu, Z. Wu, Q. Fan, S. Li, and X. Jin, "SCCGAN: Style and characters inpainting based on CGAN," *Mobile Networks and Applications*, vol. 26, no. 1, pp. 3–12 (2021), doi: 10.1007/s11036-020-01717-x.

[7] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey," *Computer Vision and Image Understanding*, vol. 223, Art. no. 103525 (Oct. 2022), doi: 10.1016/j.cviu.2022.103525.

[8] R. Chesney and D. Citron, "Deep fakes: A looming challenge for privacy, democracy, and national security," *California Law Review*, vol. 107, no. 6, pp. 1753–1820 (2019).

[9] Y. Mirsky and W. Lee, "The creation and detection of deepfakes: A survey," *ACM Computing Surveys*, vol. 54, no. 1, Art. no. 7, pp. 1–41 (Jan. 2021), doi: 10.1145/3425780.

[10] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-forensics for face swapping videos via adversarial training," *IEEE Transactions on Multimedia*, vol. 24, pp. 3429–3441 (2022), doi: 10.1109/TMM.2021.3098422.

[11] S. Wang, Y. Zou, W. Min, J. Wu, and X. Xiong, "Multi-view face generation via unpaired images," *The Visual Computer*, vol. 38, pp. 2539–2554 (2022), doi: 10.1007/s00371-021-02129-y.

[12] Y. Wang, L. Song, W. Wu, C. Qian, R. He, and C. C. Loy, "Talking faces: Audio-to-video face generation," in *Handbook of Digital Face*

Manipulation and Detection, pp. 163–188 (2022), doi: 10.1007/978-3-030-87664-7_8.

[13] C. Vaccari and A. Chadwick, "Deepfakes and disinformation: Exploring the impact of synthetic political video on deception, uncertainty, and trust in news," *Social Media + Society,* vol. 6, no. 1 (2020), doi: 10.1177/2056305120903408.

[14] Z. Wang, Y. Guo, and W. Zuo, "Deepfake forensics via an adversarial game," *IEEE Transactions on Image Processing*, vol. 31, pp. 3541–3552 (2022).

[15] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148 (2020).

[16] M. T. Jafar, M. Ababneh, M. Al-Zoube, and A. Elhassan, "Forensics and analysis of deepfake videos," in *Proc. Int. Conf. Inf. Commun. Syst. (ICICS)*, pp. 53–58 (2020).

[17] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed, "Two-branch recurrent network for isolating deepfakes in videos," in *Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Lecture Notes in Computer Science, vol. 12352. Cham, Switzerland: Springer, pp. 667–684 (2020), doi: 10.1007/978-3-030-58598-4_39.

[18] C. Gosse and J. Burkell, "Politics and porn: How news media characterizes problems presented by deepfakes," *Critical Studies in Media Communication*, vol. 37, no. 5, pp. 497–511 (2020), doi: 10.1080/15295036.2020.1832697.

[19] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. IEEE Winter Appl. Comput. Vis. Workshops (WACVW),* pp. 83–92 (2019).

[20] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in Proc. *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP),* pp. 2307–2311 (2019).

[21] J. Jiang, B. Li, B. Wei, G. Li, C. Liu, W. Huang, M. Li, and M. Yu, "FakeFilter: A cross-distribution deepfake detection system with domain adaptation," *Journal of Computer Security,* vol. 29, no. 4, pp. 403–421 (2021), doi: 10.3233/JCS-200124.

[22] T. Jung, S. Kim, and K. Kim, "DeepVision: Deepfakes detection using human eye blinking pattern," *IEEE Access,* vol. 8, pp. 83144–83154 (2020).

[23] B. Sanghvi, H. Shelar, M. Pandey, and J. Sisodia, "Detection of machine generated multimedia elements using deep learning," in *Proc. 5th Int. Conf. Comput. Methodol. Commun. (ICCMC)*, pp. 1238–1243 (2021), doi: 10.1109/ICCMC51019.2021.9418008.

[24] S. L. Fernandes and S. K. Jha, "Adversarial attack on deepfake detection using RL based texture patches," in *Proc. ECCV Workshops*, pp. 220–235 (2020).

[25] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pp. 1–11 (2019).

[26] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 3207–3216 (2020).

[27] B. Dolhansky, J. Bitton, B. Pflaum, J. Lu, R. Howes, M. Wang, and C. Canton Ferrer, "The DeepFake Detection Challenge (DFDC) dataset," arXiv preprint arXiv:2006.07397 (2020).

[28] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215 (2019).

[29] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "Deepfake detection based on discrepancies between faces and their context," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 44, no. 10, pp. 6111–6121 (2022), doi: 10.1109/TPAMI.2021.3093446.

[30] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1135–1144 (2016).

[31] A. Yermakov, J. Cech, J. Matas, and M. Fritz, "Deepfake detection that generalizes across benchmarks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV),* pp. 773–783 (2026).

[32] G. Petmezas, "Cross-dataset video deepfake detection using Transformer and CNN architectures," *Machine Vision and Applications*, vol. 37, no. 3, Art. no. 52 (2026).

[33] K. Yesugade and R. Jadhav, "Enhancing deep fake detection: Comparative analysis of advanced deep learning architectures," *Proceedings of the Indian National Science Academy*, pp. 1–14 (2026), doi: 10.1007/s43538-026-00810-1.

[34] N. N. Zanje, A. M. Bongale, and D. Dharrao, "Detecting facial image forgeries with transfer learning techniques," *International Journal of Advanced Applied Sciences*, vol. 13, no. 1, pp. 93–105 (2024), doi: 10.11591/ijaas.v13.i1.pp93-105.

[35] O. A. H. H. Al-Dulaimi and S. Kurnaz, "A hybrid CNN-LSTM approach for precision deepfake image detection based on transfer learning," *Electronics*, vol. 13, no. 9, Art. no. 1662 (2024).

[36] S. Alazwari, M. O. J. Alsamri, M. Alamgeer, R. Alabdan, I. Alzahrani, M. Rizwanullah, and A. E. Osman, "Artificial rabbits optimization with transfer learning based deepfake detection model for biometric applications," *Ain Shams Engineering Journal*, vol. 15, no. 12, Art. no. 103057 (Dec. 2024), doi: 10.1016/j.asej.2024.103057.

[37] L. Cunha, L. Zhang, B. Sowan, C. P. Lim, and Y. Kong, "Video deepfake detection using particle swarm optimization improved deep neural networks," *Neural Computing and Applications*, vol. 36, pp. 8417–8453 (May 2024), doi: 10.1007/s00521-024-09536-x.

[38] A. Al-Adwan, H. Alazzam, N. Al-Anbaki, and E. Alduweib, "Detection of deepfake media using a hybrid CNN-RNN model and particle swarm optimization (PSO) algorithm," *Computers*, vol. 13, no. 4, Art. no. 99 (2024).

[39] A. Karathanasis, J. Violos, and I. Kompatsiaris, "A comparative analysis of compression and transfer learning techniques in deepfake detection models," *Mathematics*, vol. 13, no. 5, Art. no. 887 (2025).

[40] B. Batagelj, A. Kronovšek, V. Štruc, and P. Peer, "Robust cross-dataset deepfake detection with multitask self-supervised learning," *ICT Express*, vol. 11, no. 5, pp. 858–862 (2025), doi: 10.1016/j.icte.2025.02.011.

[41] Y. Jugade, Z. Syed, C. Dcosta, A. Panicker, and D. Vora, "Deepfake detection via spatial-temporal deep networks: Leveraging CNNs and LSTMs for enhanced accuracy," *Discover Applied Sciences*, vol. 8, no. 2, Art. no. 179 (2026), doi: 10.1007/s42452-025-08014-w.

[42] A. Anshul, E. S. Chng, and D. Rajan, "AV representation learning via audio shift prediction for multimodal deepfake detection and tem-

poral localization," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, pp. 2553–2563 (2026).

[43] S. Sah, S. Kumar, R. P. Chaturvedi, G. K. Dixit, R. B. Sulaiman, and S. Kumar, "Deepfake detection using contrastive language–image pre-training (CLIP)," *Journal of Information and Optimization Sciences*, vol. 47, no. 9-A, pp. 3207–3223 (2026), doi: 10.47974/JIOS-2416.

[44] S. Sharma, P. Sharma, S. Shanker, P. Veluvali, S. Tanwar, and P. Vats, "A PKI-integrated cryptographic framework for deepfake detection via facial micro-expression analysis," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 28, no. 8, pp. 3101–3109 (2025), doi: 10.47974/JDMSC-2586.

[45] M. H. Amiri, N. Mehrabi Hashjin, M. Montazeri, S. Mirjalili, and N. Khodadadi, "Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm," *Scientific Reports*, vol. 14, Art. no. 5032 (2024), doi: 10.1038/s41598-024-54910-3.

[46] C. Yuan, D. Zhao, A. A. Heidari, L. Liu, Y. Chen, and H. Chen, "Polar lights optimizer: Algorithm and applications in image segmentation and feature selection," *Neurocomputing*, vol. 607, Art. no. 128427 (Nov. 2024), doi: 10.1016/j.neucom.2024.128427.

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 618–626 (2017), doi: 10.1109/ICCV.2017.74.

[48] M. Y. Al Mohamad, H. Bevrani, and A. A. Haydari, "Statistical tuning of artificial neural network," arXiv preprint arXiv:2409.16426 (2024).

[49] M. Y. Al-Mohamad, A. A. Heydari, H. Jabbari Khamnei, M. Lakestani, and H. Franoudkia, "Deep learning as statistical modelling: Seven families and their probabilistic foundations," *SSRN preprint*, SSRN 7467424 (2026).

Received September, 2026