ITC 4/54

Information Technology and Control
Vol. 54 / No. 4/ 2025
pp. 1139-1158
DOI 10.5755/j01.itc.54.4.40518

Deepfake Detection with Metaheuristic Algorithms and Deep Features

Received 2025/02/14 | Accepted after revision 2025/06/26

HOW TO CITE: Koçak, A., Alkan, M. (2025). Deepfake Detection with Metaheuristic Algorithms and Deep Features. *Information Technology and Control*, 54(4), 1139-1158. https://doi.org/10.5755/j01.itc.54.4.40518

Deepfake Detection with Metaheuristic Algorithms and Deep Features

Aynur Koçak*

Department of Electrical and Electronics Engineering, Faculty of Technology, Gazi University, 06500 Beşevler, Ankara, Turkey; Institute of Science, Gazi University, 06500 Beşevler, Ankara, Turkey; e-mail: aynurkocak@gazi.edu.tr

Mustafa Alkan

Department of Electrical and Electronics Engineering, Faculty of Technology, Gazi University, 06500 Beşevler, Ankara, Turkey; e-mail: alkan@gazi.edu.tr

Corresponding author: aynurkocak@gazi.edu.tr

The rapid development and spread of deepfake technology have posed serious threats to cybersecurity, information security, privacy, and public safety; consequently, reliable detection of deepfake content has become a critical necessity. We present a metaheuristic-based hybrid approach that combines deep learning architectures with metaheuristic algorithms. In the study conducted on FaceForensics (FF++) and Celeb-DF datasets, four feature vectors extracted from Xception and ResNet50 architectures underwent a metaheuristic-based feature selection process incorporating Cuckoo Search Algorithm (CS), Success-History Based Adaptive Differential Evolution with Linear Population Size Reduction (L-SHADE), Particle Swarm Optimization (PSO) and Weighted Mean of Vectors Optimization Algorithm (INFO) algorithms. Sub-feature vectors were obtained through feature selection for each algorithm. The four feature vectors obtained from the architectures and sixteen sub-feature vectors generated after feature selection were classified using machine learning and deep learning methods. When comparing performance metrics before and after feature selection, the INFO algorithm provided the highest performance across both datasets, achieving AUC values of 99.05% for the FF++ dataset and 99.01% for the Celeb-DF dataset. We believe that our comprehensive experimental study demonstrates more significant and effective results than existing methods.

KEYWORDS: Deepfake, Metaheuristic-Based, Feature Selection, Hybrid Approach, INFO

1. Introduction

Face forensics differs from traditional forensic techniques, focusing on detecting artificial intelligence (AI)-generated or digitally manipulated facial content rather than relying on physical evidence such as fingerprints or handwriting. It employs computer vision and deep learning approaches to analyze visual media, particularly in the context of deepfake detection. In recent years, AI and deep learning advancements have made it increasingly facile to generate fake images and videos through digital facial manipulation. These advancements have led to highly realistic AI-generated content, a growing diversity of manipulation techniques, and a lack of generalizable detection models that can perform consistently across different datasets and manipulation types. While these manipulations were initially employed for entertainment purposes, they have eventually evolved into a threatening dimension targeting political figures and celebrities, leading to serious security and privacy concerns in society [43, 7, 25]. Deep learning-based techniques such as Generative Adversarial Networks (GAN) [22] enable malicious actors to perform nearly imperceptible facial modifications, allowing them to successfully alter an individual's identity with a different face or expression [34].

Deepfake videos encompass many photorealistic manipulations, from altering facial expressions to scene compositions. Moreover, these manipulations have been observed even in medical imaging. For instance, the artificial processing of cancerous and benign tumors in lung disease data raises significant security concerns in this domain [30]. As such, high-quality fake images are difficult for humans to detect, so the development of automated methods for detecting of forgeries has become crucial.

The production of fake images and videos, which pose significant societal threats, has become accessible even to ordinary users. GAN-based algorithms such as StyleGAN [31, 32] have accelerated face-swapping and deceptive content creation processes, proliferating rapidly among internet users through applications like ZAO [10] and FaceApp [51]. These developments have made distinguishing deepfake content from authentic images increasingly complex. Thus, deepfake detection has gained paramount importance regarding cybersecurity and ethics. Previous work in deepfake manipulation detection focused on analyzing visible features such as eye blinking [28], heart rate [20, 46], and head pose inconsistencies [60]. These discontinuities typically arise from imperfectly preserving natural movements and facial or body anatomical variations in deepfake videos [37, 1]. However, such conventional approaches have relied on "hand-crafted" features that are valid only for specific scenarios and can extract more detailed and specific characteristics. These techniques attempted to distinguish fake videos from original ones using database-driven analyses such as facial alignment, symmetry, and color conditions [29, 42]. Nevertheless, the scenario-specific nature of hand-crafted features and their need to adapt to each new processing technique has limited the flexibility of these approaches. With the advancement of deep learning technologies, software based on architectures such as Xception [12] and similar frameworks has automatically learned deep and detailed features through multi-layered structures, marking a significant step forward in accuracy and generalization [57]. These architectures can make more accurate and reliable determinations by learning high-level features in images. In our study, deep learning-based architectures such as Xception and ResNet [24] were preferred for this purpose. Xception is a model that particularly leverages the advantages of deep convolutional network structure and has the ability to effectively learn fine details in an image. On the other hand, ResNet enables us to achieve more efficient results by utilizing residual connections to solve the vanishing gradient problem encountered in training deeper networks. Both models demonstrate robust performance in deepfake video detection.

Feature Selection (FS) is a crucial step that enhances model performance by reducing dimensionality, improving computational efficiency, and mitigating the risk of overfitting [55]. Selecting only the most relevant features makes the model more robust and interpretable, critical for high-dimensional data [9, 52]. Metaheuristic search (MHS) algorithms are employed to optimize the FS process. These algorithms enable the efficient identification of optimal feature subsets that contribute most to classification accuracy by providing flexibility in complex search spaces [2, 18]. Consequently, utilizing MHS approaches for FS improves detection accuracy and offers computational advantages in processing datasets with large frameworks [35].

While deep learning architectures offer the advantage of automatic feature learning, machine learning algorithms can provide effective results when supported by specific FS methods [27, 58]. The comparison of these two approaches aims to determine their advantages and limitations by evaluating each method's classification performance. The main contributions of this study to the literature are given below.

- This paper presents a comprehensive comparative analysis integrating two current deepfake datasets, two feature extraction methods, four metaheuristic algorithms, and both deep learning and machine learning classification approaches.
- This paper presents the implementation of CS, L-SHADE, PSO, and INFO algorithms for feature selection in deepfake detection. Based on our literature review, l-SHADE and INFO are applied for the first time in this domain.
- This paper provides an extensive evaluation of twenty feature vectors, comparing non-FS feature vectors with sub-feature vectors obtained from MHS algorithms, contributing to the literature with comprehensive performance analysis.

2. Related Work

Deepfake detection has emerged as a crucial research topic in cybersecurity and media security in recent years, with solutions in this domain continuously evolving and diversifying. Studies propose various approaches to detect deepfake videos using different methods and techniques, and these solutions are becoming increasingly effective.

2.1. Deepfake Detection Techniques

Deepfake detection methods present a wide spectrum of approaches ranging from traditional feature-based analyses to deep learning architectures, analyzing visual, auditory, and behavioral inconsistencies in manipulated content. In particular, temporal and spatial features are utilized alongside biometric characteristics in detecting fake videos. In the study conducted in [57], the model called Tception is a two-stream network that combines both RGB spatial features and noise-space features. In the proposed model, RGB features and noise features are merged. Features are learned using convolution kernels of different sizes in a parallel structure. In another study that considers both audio and visual elements, the Modality Dissonance Score (MDS) model was employed. It detects deepfake videos based on inconsistencies between audio and visual modalities [13]. The study in [36] proposed a model by analyzing eye movements and gaze orientations in each frame. The method predicts gaze orientations. The study in [14] utilizes a 3D Morphological Model (3DMM) to analyze a person's movements during speech.

In addition to temporal inconsistencies, Convolutional Neural Network (CNN) architectures are employed in the proposed models [37, 11, 54, 39, 5, 8]. The study in [11] performs deepfake video detection by combining optical flow fields with CNN. The Longterm Recurrent Convolutional Neural Network (LRCN) model is utilized to model human blinking behavior due to its temporal dependencies [37]. A CNN-based model is proposed by integrating various techniques, including manual distillation, data augmentation, and dynamic threshold classification [54]. A CNN-Long Short Term Memory (CNN-LSTM) architecture has been developed that incorporates a Facial Geometry Prior Module (FGPM) for extracting facial geometry feature maps and analyzing manipulation regions [39]. Another study using the CNN-LSTM hybrid model combines the CNN model for spatial awareness capabilities with the LSTM model for temporal context analysis [5]. Almestekawy et al., 3D images are created from face and background regions, and spatio-temporal features are extracted for deepfake detection. While face images are processed with Gray Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP), LBP features are fed to 3D CNN for further analysis. In addition, face and background features are extracted using shared weights Siamese network, and all these features are combined to detect fake video segments [8].

Generating deepfake content, particularly during image merging and resynthesis operations, can introduce inconsistencies in frequency components. Studies [59, 47, 40] demonstrate deepfake video detection methods that leverage frequency awareness and analyze frequency component patterns to identify such manipulations.

Vision Transformer (ViT), a deep learning model based on the Transformer architecture, is gaining popularity in deepfake detection [17, 21, 49]. In a study combining ViT with CNN architecture [17], the Convolutional Vision Transformer (CViT2) model determines whether an image is real or fake by extracting meaningful features from facial images using CNN while analyzing these features through its ViT component. Another ViT architecture study proposes the Vision Transformer with Xception Network (ViXNet) model [21]. In [49], a Video ViT (ViViT)-based system has been developed. ViViT processes sequential data in image patches for classification by transforming image data into vectors and utilizing the transformer encoder module.

The model proposed in [23] employs an Adaptive Manipulation Trace Extraction Network (AMTEN) to emphasize manipulation traces while suppressing image content. AMTEN utilizes convolution layers to predict manipulation traces while adapting weights during backpropagation to reuse these traces in subsequent layers. DF-UDetector, a recovery method in the feature domain rather than the image domain, models corrupted images and enhances extracted features to high quality [33].

2.2. Metaheuristic Algorithms in Deepfake Video Detection

MHS algorithms have emerged as powerful optimization techniques for solving complex and high-dimensional problems. These algorithms, inspired by natural processes (such as evolution, swarm intelligence, or physical phenomena), are practical in feature selection, parameter tuning, and model optimization [41]. In deepfake detection, metaheuristic approaches are increasingly being utilized to enhance the robustness and accuracy of detection frameworks. When integrated with machine learning and deep learning methods, these algorithms have significantly improved the detection of manipulated content [26].

In [44], the Competitive Swarm Sunflower Optimization Algorithm (CSSFOA-RMDL) method is proposed, leveraging these advantages of metaheuristic algorithms. The proposed model is developed by combining Competitive Swarm Optimization (CSO) and Sunflower Optimization (SFO), aiming to detect video forgery more effectively. Face detection is performed using the Viola-Jones algorithm, and local best-oriented pattern features are calculated considering light coefficients and facial image coefficients.

PSO is the most preferred metaheuristic algorithm in deepfake video detection and is predominantly used for hyperparameter optimization [15, 62, 4]. For instance, PSO is utilized for hyperparameter optimization in a study where EfficientNet Gate Recurrent Unit (EfficientNet-GRU) and EfficientNet-B0 architecture are used for visual feature extraction [15]. In [62], PSO is employed in a hybrid manner with 3D CNN and RNN models to perform video authenticity classification. Another hybrid implementation is realized with CNN-RNN models [4]. The PSO algorithm initializes the weights of CNN and RNN, and the model is trained by updating the loss function gradients. Video classification as 'fake' or 'real' is achieved by applying a threshold value to the RNN output.

In the hybrid approach of Ant Colony Optimization (ACO) and PSO [6], spatial and temporal patterns from manipulations in video frames are removed. These extracted features are integrated into deep learning models to process deepfake and authentic content.

In updated models within the field of deepfake video detection, the integration of metaheuristic algorithms with hybrid models has become a preferred approach to enhance detection performance. These algorithms significantly contribute, particularly in optimizing complex feature spaces and improving model efficiency by selecting crucial features. Inspired by these advancements in the literature, we incorporated metaheuristic algorithms into the feature selection process in our study. Thus, we not only achieved a superior feature set but also succeeded in developing a model that demonstrates high performance in deepfake video detection. We believe this approach makes a significant contribution to existing studies in deepfake detection.

3. Material and Method

In this study, a model integrated with MHS algorithms is proposed for the detection of fake videos. Initially, frames were extracted from video data, and feature extraction was performed on these frames using two different deep learning models, Xception and ResNet50. These extracted features were subsequently optimized using MHS algorithms, and the most significant features were selected. Finally, the selected features were classified using both deep learning-based models and machine learning methods. This process presents a systematic framework for comparing the performance of different approaches. The proposed methodology is illustrated in the flow diagram shown in Figure 1.

Figure 1

Workflow of the proposed method.

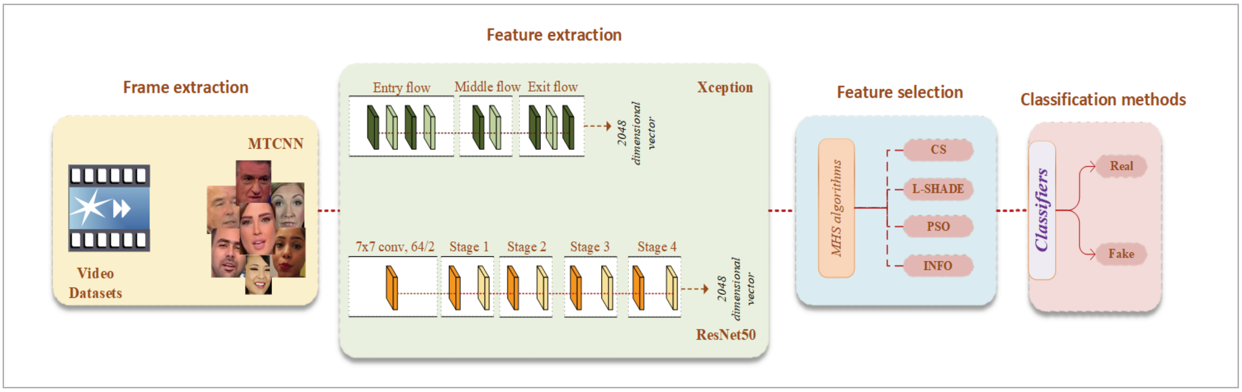
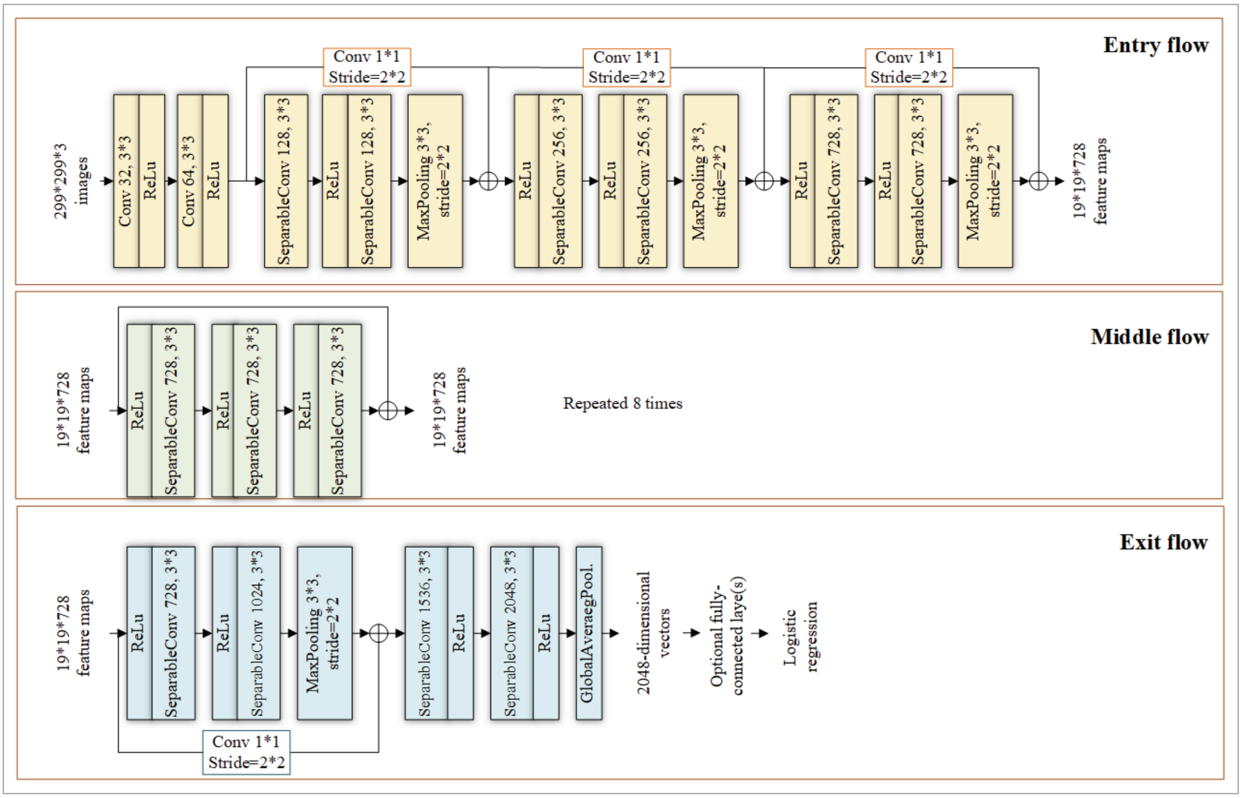


Figure 2

Xception architecture.



3.1. Frame Extraction

Preprocessing steps were applied to video data to create the image dataset used in this study. Initially, the process of extracting consecutive frames from videos was performed. In this process, the Multi-task Cascaded Convolutional Networks (MTCNN) [61] method, which demonstrates effective performance in face detection and tracking tasks, was utilized. MTCNN enabled the extraction of each frame by accurately detecting the facial region across consecutive frames in videos. This stage was implemented as a preprocessing step to prepare the video data for modeling.

3.2. Feature Extraction

For feature extraction from images, two powerful deep learning-based models, Xception and ResNet50, were employed [56]. Xception possesses an architecture based on depthwise separable convolutions and offers high performance, particularly in tasks such as image classification and feature extraction. The Xception architecture is illustrated in Figure 2.

ResNet50 is an architecture based on residual learning to solve the vanishing gradient problem encountered in training deep networks, and with its 50-layer structure enables the learning of complex features. A 2048-dimensional feature vector was obtained for each image using the proposed architectures. These high-dimensional feature vectors were used as subsets for the FS processes to be applied in subsequent stages. The architectural flow of ResNet50, used as the base architecture, is presented in Figure 3.

3.3. Feature Selection

In our experimental study, an innovation was presented by incorporating the FS method in addition to commonly used approaches in deepfake detection in the literature. FS aims to enable the model to operate more effectively by removing redundant or insignificant features from the feature vector that contribute the least to the accuracy rate. In our study, MHS algorithms were utilized for FS. MHS algorithms are powerful methods used to obtain effective results in large and complex search spaces, and in this study, a suitable solution for FS is presented.

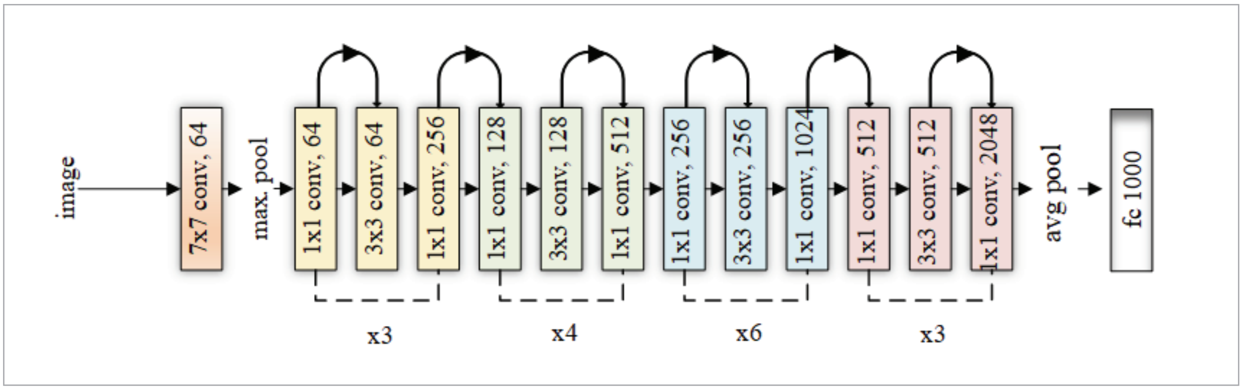
In this study, the Random Forest (RF) model was employed as a fitness function to evaluate the performance of MHS algorithms and perform the optimal feature selection process. RF served as a validation tool to measure classification performance and was solely used to assess the impact of selected features. The fitness value was defined by the Accuracy (ACC) score of the RF model trained with selected features on both training and test datasets. In each iteration, datasets were filtered using the selected features, the RF model was trained, and test accuracy was calculated as the fitness value. Thus, continuous feedback was provided to the algorithm by evaluating the performance of feature subsets. Mathematically, the fitness function is given in Equation (1).

$$F = \frac{1}{N}\sum_{i=1}^{N}\delta(\hat{y}_i, y_i) . \tag{1}$$

Here, N represents the total number of samples in the test dataset, $\hat{y}_i$ represents the predicted label by

Figure 3

ResNet50 architecture.



the model, and y_i represents the actual label. In the model, the RF classifier was preferred due to its effectiveness in working with small-subset data and its high classification performance. Additionally, the fitness value was assigned as zero in cases where no features were selected, ensuring the algorithm eliminates low-performance solutions.

In our study, four different MHS algorithms CS [48], L-SHADE [53], PSO [19] and INFO [3], were comparatively analyzed to optimize the high-dimensional feature space obtained from Xception and ResNet50 architectures. This approach aimed to achieve high accuracy rates in deepfake detection.

3.3.1. Cuckoo Search Algorithm

The CS algorithm, inspired by the egg-laying behavior of cuckoo birds and a search mechanism called Levy flight, was proposed by Rajabioun in 2011 [48]. This proposed algorithm enhances its global search capability by utilizing Levy flights and follows an evolutionary process by replacing poor solutions with better ones. The algorithm stages are sequenced as generating new solutions, evaluating fitness, selecting a nest, replacing solutions, and finding the current best. The main control parameters affecting the CS algorithm's performance are the probability (ρ_a) and the number of host nests (H).

3.3.2. Success-History Based Adaptive Differential Evolution with Linear Population Size Reduction

L-SHADE algorithm is an advanced version of the Differential Evolution (DE) [45] algorithm, which was developed to solve high-dimensional optimization problems. In this algorithm, the population size is linearly reduced as iteration progresses, thus ensuring diversity in the search space initially while performing a more focused search in the final stages. Additionally, hyperparameters such as scaling factor (F) and crossover rate (CR) are adapted based on the history of previous successful solutions. This mechanism enhances optimization success by enabling the algorithm to work adaptively in different problem domains [53]. The algorithm consists of four stages: mutation, crossover, selection step, and linear population reduction.

3.3.3. Particle Swarm Optimization

This mechanism enhances optimization success by enabling the algorithm to work adaptively in different problem domains [19]. The algorithm consists of four stages: mutation, crossover, selection step, and linear population reduction. In the algorithm, each particle represents a candidate solution in the solution space and has a velocity vector. The velocity vector determines the particle's next movement and is updated in each iteration by considering the personal best position and global best position. In velocity updating, an adaptation of particles to themselves and the swarm is achieved through control parameters called c1 (personal learning coefficient) and c2 (global learning coefficient). Additionally, the w (inertia weight) parameter controls the particle's tendency to maintain its current velocity. After the velocity update operation, the particle's position is updated according to this velocity. PSO is frequently preferred due to its simplicity, requirement for few hyperparameters, and ability to provide effective solutions across a wide range of problems through these mechanisms.

3.3.4. Weighted Mean of Vectors Optimization Algorithm

The INFO algorithm was proposed by Ahmadianfar et al. in 2022 [3]. It is a population-based optimization method where the population consists of a set of vectors [16]. These vectors represent possible solutions, and each has a specific weight. The algorithm calculates an average according to the weights of these vectors. This process consists of four stages: initialization, updating rule, vector combining, and local search. The weighted average function that the algorithm is based on is given in Equation (2).

$$WM = \frac{\sum_{i=1}^{N} x_i \times w_i}{\sum_{i=1}^{N} w_i} . \tag{2}$$

According to the given equation, N represents the total number of vectors. (x_i) denotes each position value, and (w_i) represents the weight of each position. The weighted mean (WM) is the ratio of the weighted sum ($x_i \times w_i$) to the sum of all weights (w_i). The weight calculation for each vector is given in Equation (3).

$$w = \cos(x)\times \exp\left(-\frac{x^{2}}{w}\right). \tag{3}$$

Here, w is the dilation parameter and is a constant number.

Initialization phase

At this stage, two control parameters - weighted mean factor (δ) and scaling factor (σ), and the main inputs, a set of possible (X), dimension (D) and number of vectors of INFO (N_{INFO}), are defined. Then, vectors are initialized with random initialization as given in Equation (4).

$$\begin{aligned} &X_{l,j}^{t} = \left\{X_{l,1}^{t}, X_{l,2}^{t}, \ldots, X_{l,D}^{t}\right\} \\ &l = 1,2,3,\ldots,N_{INFO} \end{aligned}. \tag{4}$$

Updating rule

After initializing the algorithm with a random start, a rule is established to proceed to the solution. For this rule, vector weights are evaluated using the worst, good, and best vectors. The goal at this stage is not to reach the best solution in the current vector but to take the average of vectors. The *MeanRule* is given in Equation (5).

$$\begin{aligned} &MeanRule = r\times WM1_{l}^{t} + (1-r)\times WM2_{l}^{t}; \\ &l = 1,2,\ldots N_{p} \end{aligned}. \tag{5}$$

In the given equation, r is a number in the range [0-0.5]. $WM1_l^t$ represents the first weighted mean value, and $WM2_l^t$ represents the second weighted mean value. These values are given in Equations (6)-(7), respectively.

$$\begin{aligned} WM1_{l}^{t} = \delta \times &\left[\begin{array}{c} \begin{pmatrix} w_1\left(x_{a1}-x_{a2}\right)+ \\ w_2\left(x_{a1}-x_{a3}\right)+ \\ w_3\left(x_{a1}-x_{a3}\right) \end{pmatrix} / \\ \left(w_1+w_2+w_3+\varepsilon\right) \end{array}\right]. \\ +\left(\varepsilon\times rand\right) & \end{aligned} \tag{6}$$

$$\begin{aligned} WM2_{l}^{t} = \delta \times &\left[\begin{array}{c} \begin{pmatrix} w_1\left(x_{bs}-x_{bt}\right)+ \\ w_2\left(x_{bs}-x_{ws}\right)+ \\ w_3\left(x_{bt}-x_{ws}\right) \end{pmatrix} / \\ \left(w_1+w_2+w_3+\varepsilon\right) \end{array}\right] \\ +\left(\varepsilon\times rand\right) & \end{aligned} \tag{7}$$

The value ε in Equations (6)-(7) is a constant number. The terms (w_1), (w_2), and (w_3) represent wavelet functions. The first, second, and third wavelet functions are used to calculate the weighted means of vectors, respectively. The wavelet functions of the first and second weighted means are given in Equations (6.1)-(6.3) and Equations (7.1)-(7.3), respectively. In the equations, the variable ω represents the dilation parameter, and $f(x)$ is referred to as the objective function. The values a_1, a_2, and a_3 are distinct integers randomly selected from the interval $[1, N_{INFO}]$.

$$\begin{aligned} &w_1 = \cos((f(x_{a1}) - f(x_{a2})) + \pi)\times \\ &\exp(-|(f(x_{a1}) - f(x_{a2}))/\omega|) \end{aligned} \tag{6.1}$$

$$\begin{aligned} &w_2 = \cos((f(x_{a1}) - f(x_{a3})) + \pi)\times \\ &\exp(-|(f(x_{a1}) - f(x_{a3}))/\omega|) \end{aligned} \tag{6.2}$$

$$\begin{aligned} &w_3 = \cos((f(x_{a2}) - f(x_{a3})) + \pi)\times \\ &\exp(-|(f(x_{a2}) - f(x_{a3}))/\omega|) \end{aligned} \tag{6.3}$$

$$\omega = \max(f(x_{a1}), f(x_{a2}), f(x_{a3})) \tag{6.4}$$

$$\begin{aligned} &w_1 = \cos((f(x_{bs}) - f(x_{bt})) + \pi)\times \\ &\exp(-|(f(x_{bs}) - f(x_{bt}))/\omega|) \end{aligned} \tag{7.1}$$

$$\begin{aligned} &w_2 = \cos((f(x_{bs}) - f(x_{ws})) + \pi)\times \\ &\exp(-|(f(x_{bs}) - f(x_{ws}))/\omega|) \end{aligned} \tag{7.2}$$

$$\begin{aligned} &w_3 = \cos((f(x_{bt}) - f(x_{ws})) + \pi)\times \\ &\exp(-|(f(x_{bt}) - f(x_{ws}))/\omega|) \end{aligned} \tag{7.3}$$

$$\omega = f(x_{ws}) \tag{7.4}$$

The weighted mean factor value (δ) is given in Equation (8). In Equation (9), (T_{INFO}) represents the maximum number of iterations/generations of INFO, and (t) is the current iteration.

$$\delta = 2 \times \beta \times rand - \beta \tag{8}$$

$$\beta = \alpha = 2 \times \exp\left(-4 \times \frac{t}{T_{INFO}}\right) \tag{9}$$

While x_{a1}, x_{a2}, and x_{a3} represent vector positions, x_{bs}, x_{bt} and x_{ws} represent the best, good, and worst solutions, respectively. The first stage of updating the rule will be completed by calculating the weighted mean. In the next step, the convergence acceleration phase will be implemented. The objective at this stage is to reach the best solution. As given in Equation (10), vectors are directed towards a better direction by differentiating the step size in each generation to reach the target. The new vector (z_l^t) is given in Equation (11).

$$CA = \begin{cases} randn \times \left(\dfrac{x_{bs} - x_{a1}^t}{f(x_{bs}) - f(x_{a1}^t) + 1}\right) \\ randn \times \left(\dfrac{x_{a2}^t - x_{a3}^t}{f(x_{a2}^t) - f(x_{a3}^t) + 1}\right) \\ randn \times \left(\dfrac{x_{a1}^t - x_{a2}^t}{f(x_{a1}^t) - f(x_{a2}^t) + 1}\right) \end{cases} \tag{10}$$

$$z_l^t = x_l^t + \sigma \times MeanRule + CA \tag{11}$$

The updating rule ($Rule_{ur}$), which supports the exploration phase of the INFO algorithm and enables global search, is defined by Equation (12). This rule is applied to create new vectors for the first and second vectors depending on the parameter r. The scaling factor σ of vectors is expressed by Equation (13) and provides a balance between exploration and exploitation processes. If the value of σ is low, exploitation is performed to improve the current position based on the weighted mean of vectors. However, when the value of σ is high, the tendency to deviate from the weighted mean increases, which promotes the exploration phase. The values of σ are calculated using Equation (14), where the constants c and d used in this calculation are set to 2 and 4, respectively.

$$Rule_{ur} = \begin{cases} z1_l^t = x_l^t + \sigma \times MeanRule + \\ \left[\begin{pmatrix} x_{bs} \\ -x_{a1}^{\ t} \end{pmatrix} / \begin{pmatrix} f(x_{bs} \\ -f(x_{a1}^t) + 1 \end{pmatrix}\right] \\ z2_l^t = x_{bs} + \sigma \times MeanRule + \\ \left[\begin{pmatrix} x_{a1}^t \\ -x_b^t \end{pmatrix} / \begin{pmatrix} f(x_{a1}^t) \\ -f(x_{a2}^t) + 1 \end{pmatrix}\right] \\ z1_l^t = x_a^t + \sigma \times MeanRule + \\ \left[\begin{pmatrix} x_{a2}^t \\ -x_{a3}^t \end{pmatrix} / \begin{pmatrix} f(x_{a2}^t) \\ -f(x_{a3}^t) + 1 \end{pmatrix}\right] \\ z2_l^t = x_{bt} + \sigma \times MeanRule + \\ \left[\begin{pmatrix} x_{a1}^t \\ -x_{a2}^t \end{pmatrix} / \begin{pmatrix} f(x_{a1}^t) \\ -f(x_{a2}^t) + 1 \end{pmatrix}\right] \end{cases} \quad r < 0.5 \tag{12}$$

$$\sigma = 2 \times \alpha \times rand - \alpha \tag{13}$$

$$\alpha = c \times \exp\left(-d \times \frac{t}{T_{INFO}}\right) \tag{14}$$

Vector combining

The third stage aims to increase population diversity and expand the scope of local search. The vectors $z1_l^t$ and $z2_l^t$, created by the updating rule operator, are combined according to the vector combining principle defined in Equation (15) to form a new vector. The parameter μ used in this process is expressed by Equation (16).

$$Rule_{vc} = \begin{cases} u_l^t = z1_l^t + \mu.\left|z1_l^t - z2_l^t\right| \\ rand_2 < 0.5 \\ rand_1 < 0.5 \\ u_l^t = z2_l^t + \mu.\left|z1_l^t - z2_l^t\right| \\ rand_2 \geq 0.5 \\ rand_1 \geq 0.5 \\ u_l^t = x_l^t \end{cases} \tag{15}$$

$$\mu = 0{,}05 \times randn \tag{16}$$

Local search

The fourth stage of the INFO algorithm focuses on the local search process and promotes exploitation activities by targeting the global best point. For this purpose, a new vector is generated using the global position (x^t_{best}) and the averaging rule. The updating rule ($Rule_{lc}$), new solution (x_{rnd}), and mean solution (x_{avg}) used in this process are defined by Equations (17)-(21).

$$Rule_{lc} = \begin{cases} u^t_l = x_{bs} + randn \times \\ (MeanRule + \\ randn \times (x^t_{bs} - x^t_{a1})), \\ rand_2 < 0.5 \\ u^t_l = x_{rnd} + randn \times \\ (MeanRule + \\ randn \times (\nu_1 \times x_{bs} - \nu_2 \times x_{rnd})), \\ rand_2 \geq 0.5 \end{cases} \tag{17}$$

$$\begin{aligned} x_{rnd} &= \phi \times x_{avg} + (1-\phi) \times \\ &(\phi \times x_{bt} + (1-\phi) \times x_{bs}) \end{aligned} \tag{18}$$

$$x_{avg} = (x_a + x_b + x_3)/3 \tag{19}$$

$$\nu_1 = \begin{cases} 2 \times rand & p > 0.5 \\ 1 & p \leq 0.5 \end{cases} \tag{20}$$

$$\nu_2 = \begin{cases} rand & p < 0.5 \\ 1 & p \geq 0.5 \end{cases} \tag{21}$$

Table 1 shows the inspiration, stages, control parameters, and parameter values used in our experimental study for the MHS algorithms.

3.4. Proposed Metaheuristic-Based Model

In this study, a metaheuristic-based model is proposed for deepfake video detection. In the first stage of the proposed model, Xception and ResNet50 deep learning architectures were used for feature extraction. The high-dimensional feature

Table 1

Inspiration, stages, control parameters and values of algorithms.

Algorithm	Inspiration	Stages	Control Parameter	Value
CUCKOO	Cuckoo behavior and Le´vy flight	Generating new solutions Evaluate fitness Choose a nest Replace the new solution Find the current best	ρ_a H	0.25 30
L-SHADE	Mutation strategy	Mutation Crossover Selection step Linear population reduction	F CR	0.8 0.9
PSO	Flock behavior	Update velocity Update value of particle	C_1 C_2 w	1.5 1.5 0.9
INFO	Weighted mean of vector	Updating rule Vector combining Local search	c d	2 4

vectors obtained from these architectures were optimized using a series of MHS algorithms to enhance classification performance. CS, L-SHADE, PSO, and INFO algorithms were applied in the FS process. These algorithms enabled the elimination of features with low contribution to classification performance and the creation of sub-feature vectors with more informative features for the classifier. This process reduces the risk of overfitting and enables the production of more meaningful, interpretable results. Videos were classified as real or fake using MultiLayer Perceptron (MLP) and Fully Connected Neural Network (FCNN) classification methods with both the feature vectors obtained from Xception and ResNet50 architectures and the sub-feature vectors obtained from MHS algorithms. This study conducted a comprehensive comparative analysis for deepfake video detection. The proposed methodology was tested on two datasets, and two feature extraction techniques were applied. Four different metaheuristic optimization algorithms were evaluated in the feature selection phase, while two different approaches based on deep learning and machine learning were used for classification. This multi-faceted experimental design enabled a detailed comparison of different approaches' performances, thereby enhancing the model's accuracy and generalizability.

3.5. Evaluation Criteria

In the deepfake video detection study, accuracy (ACC) and area under the ROC curve (AUC) values were used to evaluate performance. Accuracy provides an important metric in understanding overall model success by showing the ratio of correctly classified samples, as given in Equation (22). However, since accuracy alone would not be sufficient, particularly in imbalanced datasets, the model's discriminative ability between True Positive Rate (TPR) and False Positive Rate (FPR) was also evaluated using the AUC metric. The combined use of these two metrics comprehensively demonstrates the model's overall performance success and its sensitivity and specificity in classification. TP represents the correct classification of a fake image as fake, TN represents the correct classification of a real image as real, FN represents the misclassification of a fake image as real, and FP represents the misclassification of a real image as fake.

4. Results and Discussion

In this study, we proposed a new model leveraging the optimization capabilities and advantages of MHS algorithms, such as avoiding local minima, for detecting fake videos or images. This model was evaluated using FF++ [50] and Celeb-DF (CDF) [38] datasets, which are widely used in the field of deepfake detection.

4.1. Dataset Description

The FF++ dataset is a widely used resource for detecting deepfakes and facial manipulations, allowing the examination of effects from techniques such as face swapping and manipulation. The dataset is divided into two folders: original and manipulated. The original videos consist of two subfolders labeled as youtube and actor. The manipulated folder contains six subfolders with fake videos: Deepfakes, DeepFakeDetection, Face2Face, FaceShifter, FaceSwap, and NeuralTextures. For deepfake video detection, 1000 fake videos from the Deepfakes subfolder and 1000 real videos from the youtube subfolder are used. The videos in the dataset were divided into frames using the MTCNN method, with 4 or 5 frames extracted from each video. The total number of images is 9527, consisting of 4585 images obtained from real videos and 4942 images from fake videos.

The CDF dataset contains 590 real videos and 5639 fake videos. The high-quality deepfake videos stand out with their diverse facial expressions, varying lighting conditions, and different poses. The deepfake videos were produced more realistically with natural facial expressions, accurate lip synchronization, and minimal visual artifacts. From real videos, 2704 frames were obtained. Since there is a disproportionate number of real-fake videos in the dataset, there would be an imbalance in classification. Therefore, 5639 frames were obtained by extracting one frame from each fake video.

The dataset was divided into 80% training and 20% testing sets. To prevent class imbalance, 80% of samples from both real and fake data were selected for the training set. The remaining 20% was used as test data to evaluate the model's performance. With this data separation, the proportional distribution of real-fake classes was maintained in training and test sets. The distribution of the dataset is shown in Table 2.

Table 2

Distribution of extracted frames-cropped faces in training and test sets for both datasets.

Dataset	# Real frames in/#Fake frames in		
	Training	Test	Total
FF++	3828/3953	957/989	4785/4942
CDF	2163/3970	541/1669	2704/5639

4.2 Experimental Setup

The experiments of the proposed metaheuristic-based model were conducted using Python on the Google Colab platform. During this process, various Python libraries such as numpy, opencv, and tensorflow were utilized. For each optimization algorithm used in feature selection, the initial population size was set to 30, and the maximum generation was adjusted to 50. The number of features selected after FS is given in Table 3 according to the architectures used. These selected features were classified using MLP and FCNN methods.

Table 3

Number of features in both datasets after FS.

Dataset	Algorithm	Xception Features	ResNet50 Features
FF++	CS	999	1039
	L-SHADE	736	821
	PSO	1000	1087
	INFO	1529	1927
CDF	CS	1024	1052
	L-SHADE	1048	1901
	PSO	1007	1062
	INFO	1684	1968

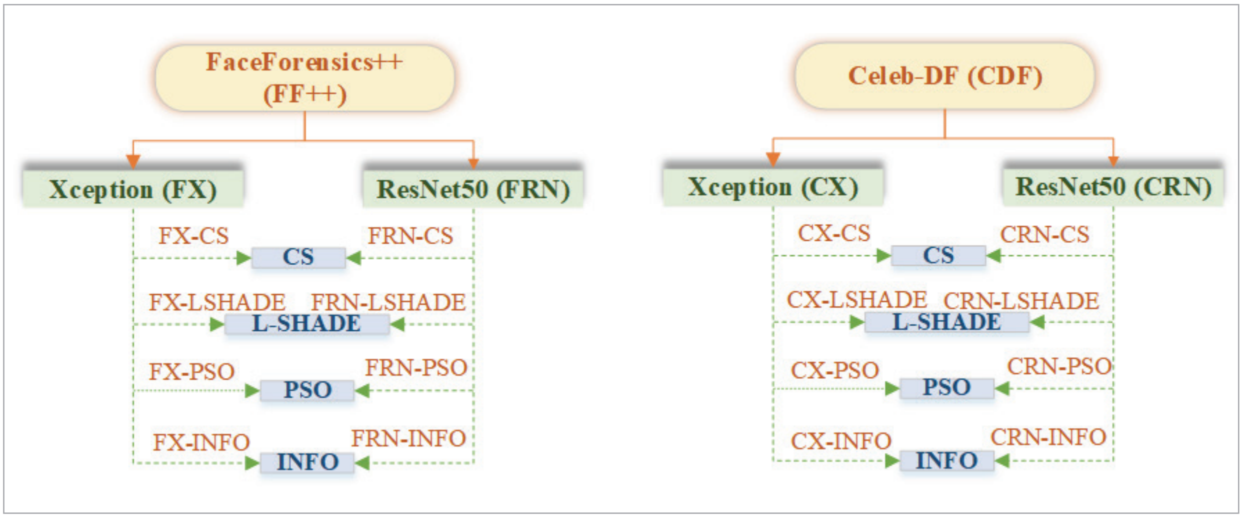
The FCNN architecture is a Dense Neural Network (DNN) structure consisting of fully connected layers and is designed for classification purposes. The input layer receives an input according to the number of selected features. The hidden layers have 512, 256, and 128 neurons, respectively. In each layer, Batch Normalization was used to ensure stability during the learning process, ReLU activation function to add non-linearity, and Dropout (with rates of 0.5 in the first layer and 0.3 in others) to prevent overfitting. The output layer consists of only one neuron with a sigmoid activation function. The model was trained with the Adam algorithm in the optimization process, with a learning rate of 0.001, batch size of 32, and epoch value of 100. This structure was designed to ensure rapid convergence during the learning process and optimize overall performance.

4.3. Experimental Results

In this section, the results of our experimental study are considered. Figure 4 presents a visual summary

Figure 4

Feature sets.



of the sub-feature sets obtained from two datasets used in our experimental study, two architectures preferred for feature extraction, and four MHS algorithms applied for FS. In this visualization, the sub-feature vectors obtained from each algorithm are referred to using abbreviations.

As shown in Figure 4, the 2048-dimensional feature vector obtained from the FF++ dataset using Xception architecture is labeled as FX, while the 2048-dimensional feature vector obtained using ResNet50 architecture is labeled as FRN. Similarly, the feature vector obtained from the CDF dataset using Xception architecture is labeled as CX, while the feature vector obtained using ResNet50 architecture is labeled as CRN. The sub-feature vectors obtained from applying CS, L-SHADE, PSO, and INFO algorithms to the FX feature vector are labeled as FX-CS, FX-LSHADE, FX-PSO, FX-INFO, respectively. In contrast, the sub-feature vectors of the FRN feature vector are labeled as FRN-CS, FRN-LSHADE, FRN-PSO, and FRN-INFO. The sub-feature vectors obtained from algorithms applied to the CX feature vector are labeled as CX-CS, CX-LSHADE, CX-PSO, and CX-INFO, respectively, while the sub-feature vectors of the CRN feature vector are labeled as CRN-CS, CRN-LSHADE, CRN-PSO, and CRN-INFO.

Table 4

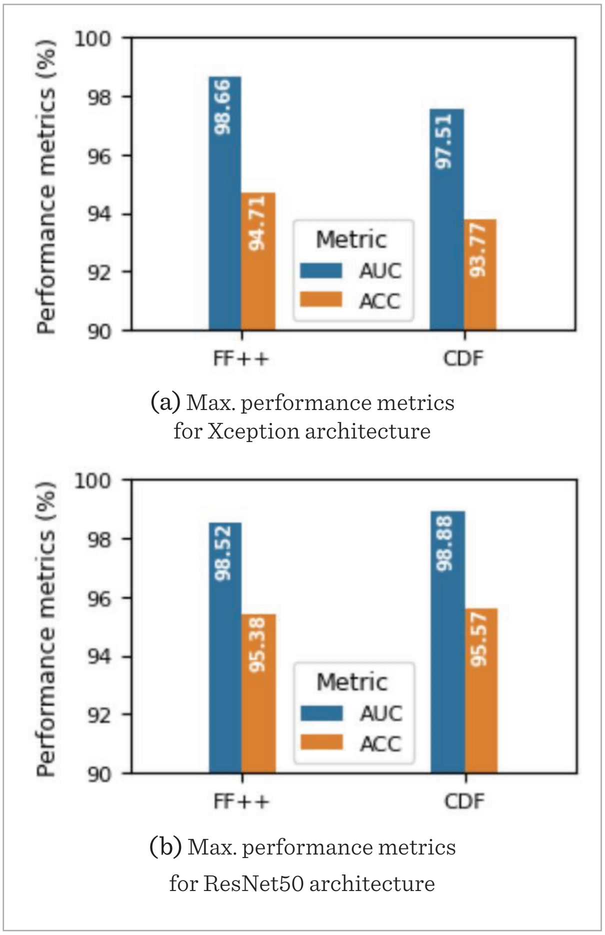
Performance metrics in Non-FS.

Dataset	Architecture	Classification methods	AUC (%)	ACC (%)
FF++	Xception	FCNN	98.34	94.71
		MLP	98.66	94.35
	ResNet50	FCNN	97.75	93.47
		MLP	98.52	95.38
CDF	Xception	FCNN	96.78	93.77
		MLP	97.51	93.55
	ResNet50	FCNN	98.88	95.57
		MLP	98.82	95.57

Table 4 shows the performance metric values before applying FS (Non-FS) to the feature vectors. For the FF++ dataset, the highest AUC value is 98.66%, and the highest ACC value is 95.58%. For the CDF dataset, the highest AUC value is 98.88%, and the highest ACC value is 95.57%. The maximum performance metric values achieved in Non-FS are shown in Figure 5.

Figure 5

Performance metrics Non-FS.



(a) Max. performance metrics for Xception architecture

(b) Max. performance metrics for ResNet50 architecture

The performance metric values of sub-feature vectors obtained from MHS algorithms are presented in Tables 5-6.

With the use of MHS algorithms, for the FF++ dataset, the ACC value increased from 95.38% (Non-FS) to 95.63%, and the AUC value increased from 98.66% (Non-FS) to 99.05% (Tables 4-5). For the CDF dataset, the ACC value increased from 95.57% (Non-FS) to 95.87%, and the AUC value increased from 98.88% (Non-FS) to 99.01% (Tables 4 and 6). Figures 6-7 show the top results of each MHS algorithm based on the classifier that achieved the highest performance. The corresponding classifier is labeled on each bar.

Table 5

MHS algorithms and performance metrics with FS on FF++ dataset.

Architecture	Classification Methods	Algorithm	AUC (%)	ACC (%)
Xception	FCNN	CS	98.48	94.76
		L-SHADE	98.72	94.66
		PSO	98.87	95.02
		INFO	99.05	95.27
	MLP	CS	98.48	95.58
		L-SHADE	98.52	95.63
		PSO	98.74	95.63
		INFO	98.85	95.38
ResNet50	FCNN	CS	97.92	94.04
		L-SHADE	97.96	93.73
		PSO	97.82	92.81
		INFO	98.16	93.99
	MLP	CS	98.48	94.55
		L-SHADE	97.63	93.58
		PSO	98.65	95.53
		INFO	98.22	94.14

Table 6

MHS algorithms and performance metrics with FS on CDF dataset.

Architecture	Classification Methods	Algorithm	AUC (%)	ACC (%)
Xception	FCNN	CS	96.40	92.27
		L-SHADE	96.33	97.49
		PSO	96.16	92.57
		INFO	96.84	92.87
	MLP	CS	96.69	92.51
		L-SHADE	92.87	93.11
		PSO	96.52	92.93
		INFO	96.93	92.87
ResNet50	FCNN	CS	98.57	94.73
		L-SHADE	98.59	95.57
		PSO	98.39	94.55
		INFO	98.64	95.21
	MLP	CS	98.60	95.07
		L-SHADE	98.99	95.87
		PSO	98.64	94.91
		INFO	99.01	95.69

Figure 6

MHS algorithms and max. performance metrics for Xception architecture.

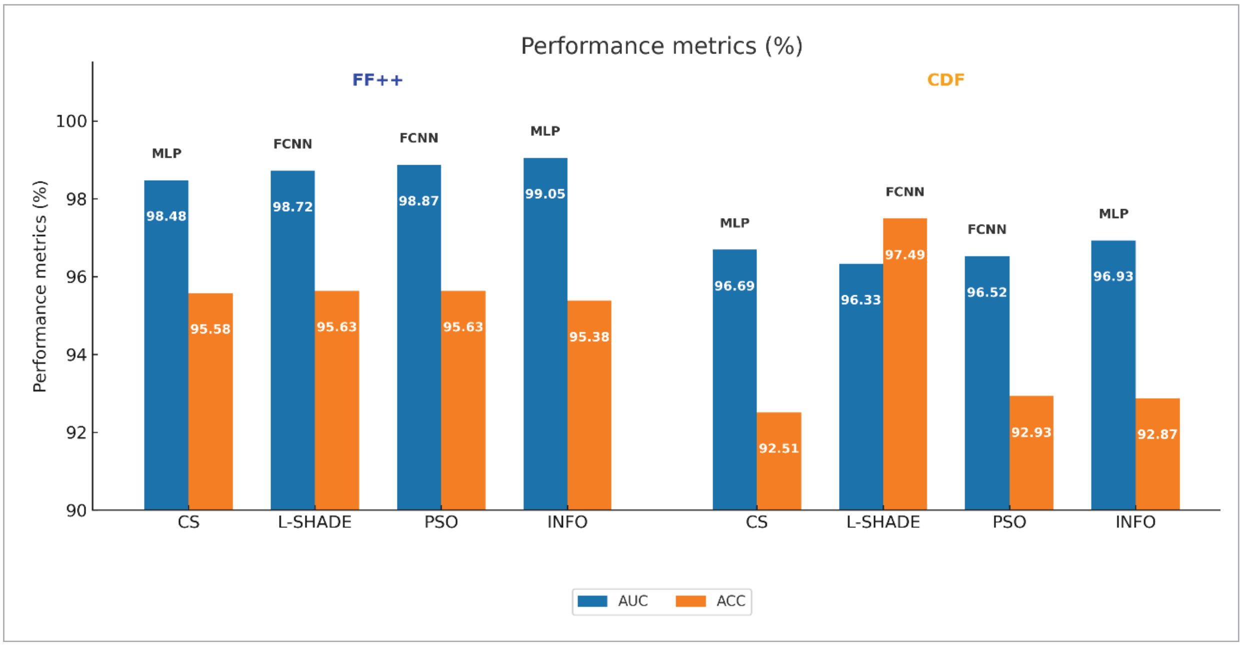
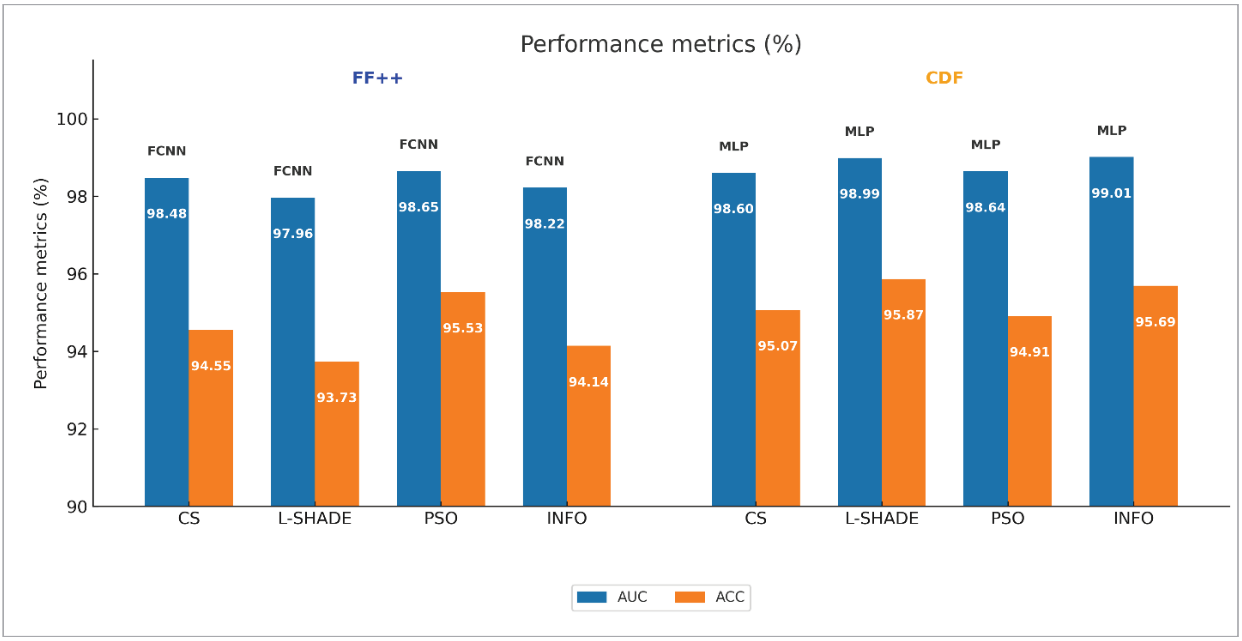


Figure 7

MHS algorithms and max. performance metrics for ResNet50 architecture.



Considering the results, the following comparisons can be made:

- In the Non-FS, for the FF++ dataset, the maximum AUC value was achieved with the FX feature vector, while the maximum ACC value was achieved with the FRN feature vector. For the CDF dataset, maximum AUC and ACC values were achieved with the CRN feature vector (Table 4).
- Considering the performance metrics of all sub-feature vectors belonging to the FF++ dataset, the highest AUC and ACC values were obtained with the FX feature vector. The maximum AUC value was achieved with the FX-INFO sub-feature vector, while the maximum ACC value was achieved with both FX-LSHADE and FX-PSO sub-feature vectors (Table 5).
- Considering the performance metrics of all sub-feature vectors belonging to the CDF dataset, the highest AUC value was obtained with the CRN feature vector, while the highest ACC value was obtained with the CX feature vector. The maximum AUC value was achieved with the CRN-INFO sub-feature vector, and the maximum ACC value was achieved with the CX-LSHADE sub-feature vector (Table 6).
- For the FF++ dataset, maximum performance metric values were achieved with the FX feature vector, while for the CDF dataset, maximum performance metric values were achieved with the CRN feature vector. In this case, features obtained from the Xception architecture for the FF++ dataset and features obtained from the ResNet50 architecture for the CDF dataset are more significant (Tables 4-6).
- For the FF++ dataset, the maximum ACC value was achieved using the MLP method, while for the CDF dataset, the maximum ACC value was achieved using the FCNN method (Tables 4-6).
- For the FF++ dataset, the AUC value increased from 98.66% to 99.05% using the INFO algorithm. For the CDF dataset, the AUC value increased from 98.88% to 99.01% using the INFO algorithm (Tables 4-6).
- For the FF++ dataset, the ACC value increased from 95.38% to 95.63% using the L-SHADE algorithm. For the CDF dataset, the ACC value increased from 95.57% to 95.87% using the L-SHADE algorithm (Tables 4-6).

Considering all experimental results in our study, the sub-feature vectors obtained from the INFO algorithm achieved maximum AUC values for both datasets. The sub-feature vectors obtained from the L-SHADE algorithm achieved maximum ACC values. The INFO algorithm and L-SHADE algorithm demonstrated maximum performance in our experimental study.

4.4. Comparison with other Deepfake Detection Techniques

Our proposed deepfake detection method is compared with most state-of-the-art methods from the literature in Table 7 [52, 2, 18, 35, 58, 36, 11, 54, 26, 44, 15]. In the research, methods tested with FF++ and CDF datasets were selected, and unused datasets or performance metrics were marked as '-'.

Results and analyses indicate that the proposed method achieves slightly higher AUC values compared to most methods in the literature, reflecting a strong discriminative capability. However, this improvement comes with a moderate reduction in ACC, suggesting a trade-off that may depend on the priorities of specific application contexts. When comparing the performance metrics of Non-FS and post-FS conditions in our study, maximum AUC values were achieved using the INFO algorithm, and maximum ACC values were achieved using the L-SHADE algorithm for both datasets.

Table 7
AUC (%) and ACC (%) scores of other methods and our proposed method on FF+ and CDF datasets.

Ref.	FF++		CDF	
	AUC (%)	ACC (%)	AUC (%)	ACC (%)
Li et al. [37]	91.8	-	-	-
Caldelli et al. [11]	-	87.20	-	-
Tran et al. [54]	-	-	97.8	-
Liang et al. [39]	-	91.36	-	85.35
Cozzolino et al. [14]	99.0	93.6	84	71.6
Cunha et al. [15]	-	-	89.85	92.47
Zhang et al. [62]	94.25	96.20	92.70	94.98
Adwan et al. [4]	-	-	-	97.26
Guo et al. [23]	-	90.11	-	-
Wang et al. [59]	94.12	90.48	86.62	79.34
Deressa et al. [17]	96	91.26	99.0	**98.25**
Qian et al. [47]	98.1	**97.52**	-	-
Ganguly et al. [21]	98.57	97.0	98.54	94.4
Liu et al. [40]	91.04	86.64	73.09	68.28
Ke et al. [33]	-	94.75	-	93.32
Ramadhani et al. [49]	-	90.27	-	87.17
Proposed Model	**99.05**	95.63	**99.01**	95.87

5. Conclusion

In this study, a metaheuristic-based model is proposed for deepfake video detection. The proposed model employs MHS algorithms for feature selection. Feature selection was implemented by preserving only significant features while eliminating deep learning features that do not contribute to performance. Additionally, MHS algorithms not previously used in the literature were employed for feature selection, and experiments were conducted on two comprehensive datasets. The two classification methods employed in the study were applied to four deep feature vectors obtained using the Xception and ResNet50 architectures on the FF++ and CDF datasets. In the subsequent phase, classification was performed on sixteen sub-feature vectors obtained by applying MHS algorithms to the four feature vectors. The goal of our study, which aimed to improve AUC and ACC values in deepfake video detection through the use of MHS algorithms for feature selection, has been successfully achieved. In line with this objective, ACC values increased from 95.38% to 95.63% for the FF++ dataset and from 95.57% to 95.87% for the CDF dataset. This improvement was achieved through the use of the L-SHADE algorithm in feature selection. The AUC values increased from 98.66% to 99.05% for the FF++ dataset and from 98.88% to 99.01% for the CDF dataset. This improvement was achieved through the use of the INFO algorithm in feature selection. Despite these improvements, the proposed approach still has certain limitations. While the results demonstrate the effectiveness of MHS-based feature selection, the use of fixed feature representations may limit adaptability to different manipulation types or real-world conditions. Additionally, employing standalone MHS algorithms allowed for comparative evaluation, but hybrid or adaptive strategies were not explored in this study. Future research could investigate more dynamic MHS frameworks, incorporate temporal modeling, and evaluate the method on broader and more diverse datasets to enhance its generalizability and robustness.

References

1. Agarwal, S., Farid, H., Fried, O., Agrawala, M., He, K., Sun, J. Protecting World Leaders Against Deep Fakes. CVPR Workshops, 2019.
2. Agrawal, P., Abutarboush, H. F., Ganesh, T., Mohamed, A. W. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019). IEEE Access, 2021, 9, 26766-26791. https://doi.org/10.1109/ACCESS.2021.3056407
3. Ahmadianfar, I., Heidari, A. A., Noshadian, S., Chen, H., Gandomi, A. H. INFO: An Efficient Optimization Algorithm Based on Weighted Mean of Vectors. Expert Systems with Applications, 2022, 195, 116516. https://doi.org/10.1016/j.eswa.2022.116516
4. Al-Adwan, A., Alazzam, H., Al-Anbaki, N., Alduweib, E. Detection of Deepfake Media Using a Hybrid CNN-RNN Model and Particle Swarm Optimization (PSO) Algorithm. Computers, 2024, 13(4), 99. https://doi.org/10.3390/computers13040099
5. Al-Dulaimi, O. A. H. H., Kurnaz, S. A Hybrid CNN-LSTM Approach for Precision Deepfake Image Detection Based on Transfer Learning. Electronics, 2024, 13(9), 1662. https://doi.org/10.3390/electronics13091662
6. Alhaji, H. S., Celik, Y., Goel, S. An Approach to Deepfake Video Detection Based on ACO-PSO Features and Deep Learning. Electronics, 2024, 13(12), 2398. https://doi.org/10.3390/electronics13122398
7. Al-Khazraji, S. H., Saleh, H. H., Khalid, A. I., Mishkhal, I. A. Impact of Deepfake Technology on Social Media: Detection, Misinformation, and Societal Implications. The Eurasia Proceedings of Science, Technology, Engineering and Mathematics, 2023, 23, 429-441. https://doi.org/10.55549/epstem.1371792
8. Almestekawy, A., Zayed, H. H., Taha, A. Deepfake Detection: Enhancing Performance with Spatiotemporal Texture and Deep Learning Feature Fusion. Egyptian Informatics Journal, 2024, 27, 100535. https://doi.org/10.1016/j.eij.2024.100535
9. Asir Antony, D., Appavu Alias Balamurugan, S., Jebamalar Leavline, E. Literature Review on Feature Selection Methods for High-Dimensional Data. International Journal of Computer Applications, 2016, 136(1), 9-17. https://doi.org/10.5120/ijca2016908317
10. Baltrušaitis, T., Robinson, P., Morency, L.-P. OpenFace: An Open Source Facial Behavior Analysis Toolkit. IEEE Winter Conference on Applications of Computer Vision (WACV), 2016. IEEE. https://doi.org/10.1109/WACV.2016.7477553
11. Caldelli, R., Di Pietro, R., Marcelloni, F., Del Bimbo, A., Ballerini, E. Optical Flow Based CNN for Detection of Un-Learnt Deepfake Manipulations. Pattern Recognition Letters, 2021, 146, 31-37. https://doi.org/10.1016/j.patrec.2021.03.005
12. Chollet, F. Xception: Deep Learning with Depthwise

Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. https://doi.org/10.48550/arXiv.1610.02357 https://doi.org/10.1109/CVPR.2017.195

13. Chugh, K., Gupta, P., Dhall, A., Subramanian, R. Not Made for Each Other - Audio-Visual Dissonance-Based Deepfake Detection and Localization. Proceedings of the 28th ACM International Conference on Multimedia, 2020. https://doi.org/10.1145/3394171.3413700

14. Cozzolino, D., Rössler, A., Thies, J., Nießner, M., Verdoliva, L. ID-Reveal: Identity-Aware Deepfake Video Detection. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. https://doi.org/10.1109/ICCV48922.2021.01483

15. Cunha, L., Zhang, L., Sowan, B., Lim, C. P., Kong, Y. Video Deepfake Detection Using Particle Swarm Optimization Improved Deep Neural Networks. Neural Computing and Applications, 2024, 36(15), 8417-8453. https://doi.org/10.1007/s00521-024-09536-x

16. Demirtas, M., Koc, K. Parameter Extraction of Photovoltaic Cells and Modules by INFO Algorithm. IEEE Access, 2022, 10, 87022-87052. https://doi.org/10.1109/ACCESS.2022.3198987

17. Deressa, D. W., Kuo, C.-L., Chen, H.-C., Lee, C.-H., Chang, H.-M., Chen, Y.-C., Chang, C.-H. Improved Deepfake Video Detection Using Convolutional Vision Transformer. Proceedings of the 2024 IEEE Gaming, Entertainment, and Media Conference (GEM), 2024. IEEE. https://doi.org/10.1109/GEM61861.2024.10585593

18. Diao, R., Shen, Q. Nature Inspired Feature Selection Meta-Heuristics. Artificial Intelligence Review, 2015, 44, 311-340. https://doi.org/10.1007/s10462-015-9428-8

19. Eberhart, R., Kennedy, J. Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, 1995. Citeseer. https://doi.org/10.1109/ICNN.1995.488968

20. Fernandes, S., Raj, S., Ortiz, E., Vintilă, I., Salter, M., Urošević, G., Jha, S. K. Predicting Heart Rate Variations of Deepfake Videos Using Neural ODE. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2019, 1721-1729. https://doi.org/10.1109/ICCVW.2019.00213

21. Ganguly, S., Ganguly, A., Mohiuddin, S., Malakar, S., Sarkar, R. ViXNet: Vision Transformer with Xception Network for Deepfakes Based Video and Image Forgery Detection. Expert Systems with Applications, 2022, 210, 118423. https://doi.org/10.1016/j.eswa.2022.118423

22. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. Generative Adversarial Nets. Advances in Neural Information Processing Systems, 2014, 27. https://doi.org/10.48550/arXiv.1406.2661

23. Guo, Z., Yang, G., Chen, J., Sun, X. Fake Face Detection via Adaptive Manipulation Traces Extraction Network. Computer Vision and Image Understanding, 2021, 204, 103170. https://doi.org/10.1016/j.cviu.2021.103170

24. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. cv-foundation.org https://doi.org/10.1109/CVPR.2016.90

25. Heidari, A., Jafari Navimipour, N., Dag, H., Unal, M. Deepfake Detection Using Deep Learning Methods: A Systematic and Comprehensive Review. WIREs Data Mining and Knowledge Discovery, 2023, 14(2), e1520. https://doi.org/10.1002/widm.1520

26. Houssein, E. H., Saber, E., Ali, A. A., Wazery, Y. M. Centroid Mutation-Based Search and Rescue Optimization Algorithm for Feature Selection and Classification. Expert Systems With Applications, 2022, 191, 116235. https://doi.org/10.1016/j.eswa.2021.116235

27. Janiesch, C., Zschech, P., Heinrich, K. Machine Learning and Deep Learning. Electronic Markets, 2021, 31(3), 685-695. https://doi.org/10.1007/s12525-021-00475-2

28. Jung, T., Kim, S., Kim, K. DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern. IEEE Access, 2020, 8, 83144-83154. https://doi.org/10.1109/ACCESS.2020.2988660

29. Kaddar, B., Bouhlel, M. S., El Alaoui, A., Chergui, H., Lahjouji, A. On the Effectiveness of Handcrafted Features for Deepfake Video Detection. Journal of Electronic Imaging, 2023, 32(5), 053033. https://doi.org/10.1117/1.JEI.32.5.053033

30. Karaköse, M., Yetış, H., Çeçen, M. A New Approach for Effective Medical Deepfake Detection in Medical Images. IEEE Access, 2024, 12, 52205-52214. https://doi.org/10.1109/ACCESS.2024.3386644

31. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T. Analyzing and Improving the Image Quality of StyleGAN. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. https://doi.org/10.1109/CVPR42600.2020.00813

32. Karras, T., Laine, S., Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv Preprint, arXiv:1812.04948, 2019. https://doi.org/10.1109/CVPR.2019.00453

33. Ke, J., Wang, L. DF-UDetector: An Effective Method Towards Robust Deepfake Detection via Feature Restoration. Neural Networks, 2023, 160, 216-226. https://doi.org/10.1016/j.neunet.2023.01.001

34. Khochare, J., Joshi, C., Yenarkar, B., Suratkar, S., Kazi, F. A Deep Learning Framework for Audio Deepfake Detection. Arabian Journal for Science and Engineering, 2021, 1-12. https://doi.org/10.1007/s13369-021-06297-w

35. Kowdiki, M., Khaparde, A. Automatic Hand Gesture Recognition Using Hybrid Meta-Heuristic-Based Feature Selection and Classification with Dynamic Time Warping. Computer Science Review, 2021, 39, 100320. https://doi.org/10.1016/j.cosrev.2020.100320

36. Li, M., Liu, B., Hu, Y., Wang, Y. Exposing Deepfake Videos by Tracking Eye Movements. 2020 25th International Conference on Pattern Recognition (ICPR), 2021. IEEE. https://doi.org/10.1109/ICPR48806.2021.9413139

37. Li, Y., Chang, M.-C., Lyu, S. In Ictu Oculi: Exposing AI-Created Fake Videos by Detecting Eye Blinking. 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018. https://doi.org/10.1109/WIFS.2018.8630787

38. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S. Celeb-DF: A Large-Scale Challenging Dataset for Deepfake Forensics. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, 3207-3216. https://doi.org/10.1109/CVPR42600.2020.00327

39. Liang, P., Liu, G., Xiong, Z., Fan, H., Zhu, H., Zhang, X. A Facial Geometry Based Detection Model for Face Manipulation Using CNN-LSTM Architecture. Information Sciences, 2023, 633, 370-383. https://doi.org/10.1016/j.ins.2023.03.079

40. Liu, H., Li, X., Zhou, W., Chen, Y., He, Y., Xue, H., Zhang, W., Yu, N. Spatial-Phase Shallow Learning: Rethinking Face Forgery Detection in Frequency Domain. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. https://doi.org/10.1109/CVPR46437.2021.00083

41. Ma, J., Xia, D., Wang, Y., Niu, X., Jiang, S., Liu, Z., Guo, H. A Comprehensive Comparison Among Metaheuristics (MHs) for Geohazard Modeling Using Machine Learning: Insights from a Case Study of Landslide Displacement Prediction. Engineering Applications of Artificial Intelligence, 2022, 114, 105150. https://doi.org/10.1016/j.engappai.2022.105150

42. Mohiuddin, S., Chakraborty, S., Sengupta, K., Chattopadhyay, S., Paul, S., Ghosh, A., Bandyopadhyay, S., Chowdhury, A. A Hierarchical Feature Selection Strategy for Deepfake Video Detection. Neural Computing and Applications, 2023, 35(13), 9363-9380. https://doi.org/10.1007/s00521-023-08201-z

43. Naitali, A., Ridouani, M., Salahdine, F., Kaabouch, N. Deepfake Attacks: Generation, Detection, Datasets, Challenges, and Research Directions. Computers, 2023, 12(10), 216. https://doi.org/10.3390/computers12100216

44. Nirmala Priya, G., Kishore, B. Video Forgery Detection Using Competitive Swarm Sunflower Optimization Algorithm Based Deep Learning. Wireless Networks, 2024, 1-19. https://doi.org/10.1007/s11276-024-03743-z

45. Pant, M., Zaheer, H., Garcia-Hernandez, L., Abraham, A. Differential Evolution: A Review of More Than Two Decades of Research. Engineering Applications of Artificial Intelligence, 2020, 90, 103479. https://doi.org/10.1016/j.engappai.2020.103479

46. Qi, H., Guo, Q., Juefei-Xu, F., Xie, X., Ma, L., Feng, W., Liu, Y., Zhao, J. DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms. Proceedings of the ACM International Conference on Multimedia (ACM MM), 2020. https://doi.org/10.1145/3394171.3413707

47. Qian, Y., Yin, G., Sheng, L., Chen, Z., Shao, J. Thinking in Frequency: Face Forgery Detection by Mining Frequency-Aware Clues. European Conference on Computer Vision, 2020, 86-103. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58610-2_6

48. Rajabioun, R. Cuckoo Optimization Algorithm. Applied Soft Computing, 2011, 11(8), 5508-5518. https://doi.org/10.1016/j.asoc.2011.05.008

49. Ramadhani, K. N., Munir, R., Utama, N. P. Improving Video Vision Transformer for Deepfake Video Detection Using Facial Landmark, Depthwise Separable Convolution, and Self Attention. IEEE Access, 2024, 12, 8932-8939. https://doi.org/10.1109/ACCESS.2024.3352890

50. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M. FaceForensics++: Learning to Detect Manipulated Facial Images. Proceedings of the IEEE/CVF International Conference on Comput-

er Vision (ICCV), 2019, 1-11.https://doi.org/10.1109/ICCV.2019.00009

51. Schroff, F., Kalenichenko, D., Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015. https://doi.org/10.1109/CVPR.2015.7298682
52. Sharafaddini, A. M., Mansouri, N. A Binary Chaotic Transient Search Optimization Algorithm for Enhancing Feature Selection. Arabian Journal for Science and Engineering, 2024, 1-24. https://doi.org/10.1007/s13369-024-08861-6
53. Tanabe, R., Fukunaga, A. S. Improving the Search Performance of SHADE Using Linear Population Size Reduction. 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, 1658-1665. IEEE. https://doi.org/10.1109/CEC.2014.6900380
54. Tran, V.-N., Lee, S.-H., Le, H.-S., Kwon, K.-R. High Performance DeepFake Video Detection on CNN-Based with Attention Target-Specific Regions and Manual Distillation Extraction. Applied Sciences, 2021, 11(16), 7678. https://doi.org/10.3390/app11167678
55. Venkatesh, B., Anuradha, J. A Review of Feature Selection and Its Methods. Cybernetics and Information Technologies, 2019, 19(1), 3-26. https://doi.org/10.2478/cait-2019-0001
56. Vo, H.-T., Mui, K. C., Thien, N. N. Fusing Xception and ResNet50 Features for Robust Grape Leaf Disease Classification. Bulletin of Electrical Engineering and Informatics, 2025, 14(1), 268-275. https://doi.org/10.11591/eei.v14i1.7269
57. Wang, B., Zhang, Y., Zhou, H., Li, J., Chen, W. Two-Stream Xception Structure Based on Feature Fusion for DeepFake Detection. International Journal of Computational Intelligence Systems, 2023, 16(1), 134. https://doi.org/10.1007/s44196-023-00312-8
58. Wang, P., Fan, E., Wang, P. Comparative Analysis of Image Classification Algorithms Based on Traditional Machine Learning and Deep Learning. Pattern Recognition Letters, 2021, 141, 61-67. https://doi.org/10.1016/j.patrec.2020.07.042
59. Wang, Y., Sun, Q., Rong, D., Geng, R. Multi-Domain Awareness for Compressed Deepfake Videos Detection Over Social Networks Guided by Common Mechanisms Between Artifacts. Computer Vision and Image Understanding, 2024, 247, 104072. https://doi.org/10.1016/j.cviu.2024.104072
60. Yang, X., Li, Y., Lyu, S. Exposing Deep Fakes Using Inconsistent Head Poses. ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. https://doi.org/10.1109/ICASSP.2019.8683164
61. Zhang, K., Zhang, Z., Li, Z., Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 2016, 23(10), 1499-1503. https://doi.org/10.1109/LSP.2016.2603342
62. Zhang, L., Zhao, D., Lim, C. P., Asadi, H., Huang, H., Yu, Y., Gao, R. Video Deepfake Classification Using Particle Swarm Optimization-Based Evolving Ensemble Models. Knowledge-Based Systems, 2024, 289, 111461. https://doi.org/10.1016/j.knosys.2024.111461