



Research article

Deepfake image detection and classification model using Bayesian deep learning with coronavirus herd immunity optimizer

Wahida Mansouri¹, Amal Alshardan², Nazir Ahmad^{3,*} and Nuha Alruwais⁴

¹ Department of Computer Science and Information Technology, Faculty of Sciences and Arts, Turaif, Northern Border University, Arar 91431, Saudi Arabia

² Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia

³ Department of Computer Science, Applied College at Mahayil, King Khalid University, Saudi Arabia

⁴ Department of Computer Science and Engineering, College of Applied Studies and Community Services, King Saud University, P.O. Box 22459, Riyadh 11495, Saudi Arabia

* **Correspondence:** Email: naahmad@kku.edu.sa.

Abstract: Deepfake images are combined media constructed from deep learning (DL) methods, usually Generative Adversarial Networks (GANs), to manipulate visual content, often giving rise to convincing and fabricating descriptions of scenes or people. The Bayesian machine learning (ML) model has made crucial strides over the past two decades, illustrating promise in diverse applications. In deepfake images, detection utilizes computer vision (CV) and ML to spot manipulated content by analyzing unique artefacts and patterns. Recent techniques utilize DL to train neural networks to discriminate between real and fake images, improving the fight against digital manipulation and preserving media integrity. These systems can efficiently detect subtle inconsistencies or anomalies specific to deepfake creations by learning from large datasets of both real and deepfake images. This enables the mitigation of fraudulent content and reliable detection in digital media. We introduce a new Coronavirus Herd Immunity Optimizer with a Deep Learning-based Deepfake Image Detection and Classification (CHIODL-DIDC) technique. The CHIODL-DIDC technique aimed to detect and classify the existence of fake images. To accomplish this, the CHIODL-DIDC technique initially used a median filtering (MF) based image filtering approach. Besides, the CHIODL-DIDC technique utilized the MobileNetv2 model for extracting feature vectors. Moreover, the hyperparameter tuning of the MobileNetv2 model was accomplished using the CHIO method. For deepfake image detection,

the CHIODL-DIDC technique implements the deep belief network (DBN) model. Finally, the Bayesian optimization algorithm (BOA) was utilized to select the effectual hyperparameter of the DBN model. The CHIODL-DIDC method's empirical analysis was examined using a benchmark fake image dataset. The performance validation of the CHIODL-DIDC technique illustrated a superior accuracy value of 98.16% over other models under $Accu_y$, $Prec_n$, $Reca_l$, F_{Score} , and MCC metrics.

Keywords: deepfake; deep learning; image detection; coronavirus herd immunity optimizer; machine learning

Mathematics Subject Classification: 11Y40

1. Introduction

The common accessibility of low-cost digital devices, including laptops, smartphones, digital cameras, and desktop computers, has generated the growth of multimedia material (like movies and photographs) on the Internet and wireless communication methods [1]. Also, in the past few years, social media has permitted people to rapidly communicate recorded multimedia content, resulting in a major evolution in multimedia content output and availability. Improved the rapidity with which fake and wrong data can be produced and spread, knowing the reality and believing the information has become gradually challenging, maybe resulting in terrible consequences [2]. Therefore, a deepfake is material produced by the DL model, which seems real in a human's eyes. The word deepfake is a combination of the phrases DL and false, and it normally refers to material made by a deep neural network (DNN) that is a subdivision of an ML model. Everybody can use image and video items [3]. This has been accessible for numerous years owing to numerous accessible software packages that permit video editing, pictures, and audio [4]. The adoption of smartphone applications to perform automatic processes, such as audio instrumentation, lip-syncing, and face swaps, has simplified media manipulation. Also, DL-driven technological growths have resulted in a slide of AI-driven technologies that create influences that are very satisfying and realistic [5]. Each of these methods is a beneficial addition to the digital artist toolbox. However, when utilized maliciously to generate false media, they may have important negative personal or social suggestions. Deepfakes are a famous artificially employed media that has produced severe concern [6]. This frequently spreads false data by imitating politicians and dispensing revenge porn. The spread of deepfake tools provides growth to many anxieties and possible hazards through many businesses. One major region obstructed is cybersecurity, where the capability to operate facial images strongly increases fears about individuality theft, tricks, and illegal access to sensitive information [7].

Besides, the general usage of deepfakes poses an extensive danger to community faith, as malicious persons can utilize this skill to construct dishonest graphic cues, smudge the reputation of others, and spread misinformation. Owing to these problems, academics and researchers have been concentrating on developing models to identify and diminish the opposing effects of deepfakes. With emerging innovative techniques, the goal is to protect organizations and individuals from the latent damages modeled by this developing technology [8]. It contains the development prepared in CV, ML, and forensic study to identify vital signs of image processing and efficiently discriminate amongst manipulated and authentic facial imageries. Numerous techniques have been proposed to discover deepfakes and a major part of trust in DL models. Presently, many prominent methods have been proposed for recognizing fake images. However, these approaches frequently display restricted

generalized skills, leading to reduced performance when handled with modern manipulation or deepfake techniques [9]. The proliferation of affordable digital devices and widespread social media use has paved the way for a surge in multimedia content creation and sharing. This rapid increase has unfortunately been accompanied by the rise of manipulated media, which can be deceptively realistic. As the spread of such altered content becomes more prevalent, distinguishing genuine data from falsified material has become increasingly complex, potentially resulting in severe consequences. This challenge emphasizes the requirement for advanced detection and classification methods to resist the proliferation of misleading or fabricated media [10].

We introduce a new coronavirus herd immunity optimizer with a deep learning-based deepfake image detection and classification (CHIODL-DIDC) technique. The CHIODL-DIDC technique aims to detect and classify the existence of fake images and uses a median filtering (MF) based image filtering approach. Besides, the CHIODL-DIDC technique utilizes the MobileNetv2 model for extracting feature vectors. Moreover, the hyperparameter tuning of the MobileNetv2 model is achieved using the CHIO method. For deepfake image detection, the CHIODL-DIDC technique implements the deep belief network (DBN) model. Finally, the Bayesian optimization algorithm (BOA) is utilized to select the effectual hyperparameter of the DBN model. The CHIODL-DIDC method's empirical analysis is examined using a benchmark fake image dataset. The major contribution of the CHIODL-DIDC method is listed below.

- The CHIODL-DIDC approach utilizes the MF technique to mitigate noise in images, which substantially improves the quality of the input data. This preprocessing step confirms that subsequent stages receive cleaner, more accurate data for evaluation. As a result, the overall performance of the visual recognition process is enhanced.
- The CHIODL-DIDC methodology employs the MobileNetv2 technique for feature extraction, implementing its lightweight and effective architecture to capture robust and high-quality image features. This confirms that the extracted features are elaborate and computationally effective, improving the accuracy of the visual recognition process.
- The CHIODL-DIDC approach integrates the CHIO technique for tuning hyperparameters, which refines the model's performance by optimizing parameter settings. This methodology enhances the model's accuracy and effectiveness, confirming improved outcomes in visual recognition tasks.
- The CHIODL-DIDC model utilizes the DBN approach for image recognition, employing its DL capabilities to classify visual data precisely. This incorporation improves visual recognition accuracy by effectively learning and interpreting complex patterns in the images.
- The CHIODL-DIDC technique implements the BOA method for additional parameter tuning, which refines the accuracy and effectiveness of the model. This additional optimization step improves the model's performance by fine-tuning parameters to attain enhanced outcomes in visual recognition tasks.
- The CHIODL-DIDC method innovatively incorporates diverse advanced techniques into a cohesive framework: MF-based preprocessing for noise reduction, MobileNetv2 for feature extraction, CHIO and BOA for parameter tuning, and DBN for image recognition. This multi-stage technique improves overall performance and effectiveness, setting a new standard by seamlessly integrating these techniques to optimize every stage of the visual recognition process.

The remaining sections of the article are arranged as follows: In Section 2, we provide the literature review. In Section 3, we show the proposed method. Then, we provide the results in Section 4. In Section 5, we complete the work.

2. Related works

Sushir et al. [11] presented an optimum blind forgery recognition utilizing DL techniques. The VGGNet and Hybrid dual-tree complex wavelet trigonometric transform (Hybrid DTT) is employed for extraction. The feature dimensional is decreased over improved horse herd optimizer (IHH). At last, the hybrid deep convolutional capsule autoencoder (Hybrid DCCA) structure is employed for recognition. In [12], the fisherface linear binary pattern histogram utilizing the DBN (FF-LBPH DBN) model has been executed. Deepfake face image operations were analyzed by employing the proposed technique, which also formed a higher stage of performance. The function of preprocessing is completed by utilizing Kalman filtering to recognize fake imageries in a modified detection. A fusion of the FF-LBPH model used the decrease in the dimensionality of features. Ghosh et al. [13] proposed a method for removing the imagery content, categorizing it, and confirming digital imageries' reality (genuine or forged). The technique utilizes CNN to identify genuine and fake images. The error rate bid information and the DL technique have been employed for additional solutions. Hashmi et al. [14] introduced a strong technique for fake news recognition. This method combined FastText word embedding with numerous ML and DL models. Particularly, a hybrid method uniting CNN and LSTM, supplemented with FastText embedding, exceeded other methods. Moreover, advanced transformer-based methods like XLNet, BERT, and RoBERTa are used to improve them over hyperparameter alterations. Boyd et al. [15] proposed a Convey Brain Oversight training approach to Raise Generalization (CYBORG). These novel methods include human-annotated saliency mapping into the loss function, which directs the model's learning to concentrate on imagery areas. The Class Activation Mapping (CAM) device was employed to inquire about the model's saliency in every training batch and correct huge dissimilarities.

Li et al. [16] proposed a novel generation technique utilizing a one-class classification method. The technique projected in this paper has subsequent features such as numerous filter enhancement models; an enhanced Multi-Channel CNN (MCCNN) has been implemented as the foremost system; the method increased the data utilizing weakly supervised learning techniques and trained it in dual stages. The 1st and 2nd steps used a corresponding dual and one-class classification loss function. Dwivedi and Wankhade [17] developed a semantically improved multi-modal fake news recognition technique that uses pre-trained language methods to seize contained factual knowledge and openly removes visual objects to recognize the deep semantics of multi-modal news better. This method removes noticeable features at dissimilar semantic levels, utilizes a text-guided attention mechanism to model semantic relations among text and imageries, and combines multi-modal features. The authors [18] presented a cooperative DL-based false news recognition method. The proposed method utilizes consumer responses to evaluate news faith stages, and news position was defined as dependent upon these values. High-ranked content is known as real news, whereas low-ranked news is conserved for language processing to safeguard its strength. The CNN technique was employed to turn consumer feedback into positions in the layer of DL. Zhang et al. [19] propose a methodology using weighted and evolving ensemble models with 3D CNNs and CNN-RNNs. This technique utilizes a Particle Swarm Optimization (PSO) approach, which improves network topologies and learning parameters through advanced techniques such as Muller's method and reinforcement learning. Chen et al. [20] introduce a DeepFake detection technique that incorporates a Variational Autoencoder (VAE) and a GAN model (D-VAEGAN). It also extracts an encoder and decoder to reconstruct clean images from low-dimensional features. It also uses an additional discriminative network and feature similarity loss to enhance image quality and adversarial robustness.

Omar et al. [21] present a bagging ensemble classifier to detect manipulated faces in videos by utilizing CoAtNet. This methodology integrates depthwise convolution and self-attention layers to enhance feature extraction and capture local and global data. The model classifies videos as real or fake by training on random data subsets and aggregating predictions. CutMix data augmentation is employed to improve generalization and localization. Yang et al. [22] introduce a multi-modal forgery detection technique that integrates face recognition, video frame extraction, and rPPG signal analysis utilizing 3D and 2D CNNs, incorporated through a stacking approach for enhanced accuracy. Hasanaath et al. [23] introduce a Frequency Enhanced Self-Blended Images (FSBI) methodology for deepfake detection. It utilizes Discrete Wavelet Transforms (DWT) to extract features from self-blended images, which mix an image with its own copy to introduce forgery artefacts. Rangarajan et al. [24] developed an AI model by utilizing the DenseNet121 model for detecting fake images. Ilyaset al. [25] present the AVFakeNet framework for deepfake detection, utilizing both audio and visual modalities. The model features the Dense Swin Transformer Net (DST-Net) method, which incorporates dense layers in the input and output blocks and a custom Swin Transformer module for feature extraction. Ige et al. [26] propose a versatile framework incorporating DL and Random Forest methodologies to improve phishing attack detection through image analysis, speech synthesis from deep fake videos, and natural language processing models. Liu et al. [27] introduce a deepfake detector that utilizes a Noise Residual Unit (NRU) to emphasize features by comparing Gaussian-noise-degraded images with their high-pass-filtered versions, generating a Noise Residual Image (NRI).

The limitations of the existing studies encompass difficulties in handling high-dimensional feature spaces and intrinsic backgrounds, which can affect the robustness of forgery recognition. Others might encounter high computational costs or require extensive data for efficient training and generalization across varied fake image manipulations. Some techniques may face difficulties scaling to large datasets or incorporating diverse data types effectually. There are also threats related to the detection of growing forgery methods and the accuracy of consumer feedback in cooperative approaches. Furthermore, complex frameworks integrating diverse modalities or DL methods may be less practical for real-time applications due to their computational demands and integration issues. Existing techniques for deepfake detection encounter threats with high-dimensional feature spaces, scalability, and the growing behavior of forgery techniques. There is a requirement for more effectual, scalable solutions that integrate various data types and adapt to new manipulation methods while maintaining real-time applicability.

3. The proposed model

This article introduces a new CHIODL-DIDC method. The method aims to detect and classify fake images. To obtain this, it comprises several stages: Image preprocessing, MobileNetv2-based feature extractor, CHIO-based parameter tuning, DBN-based image recognition, and BOA-based parameter tuning. Figure 1 represents the flow from the CHIODL-DIDC technique.

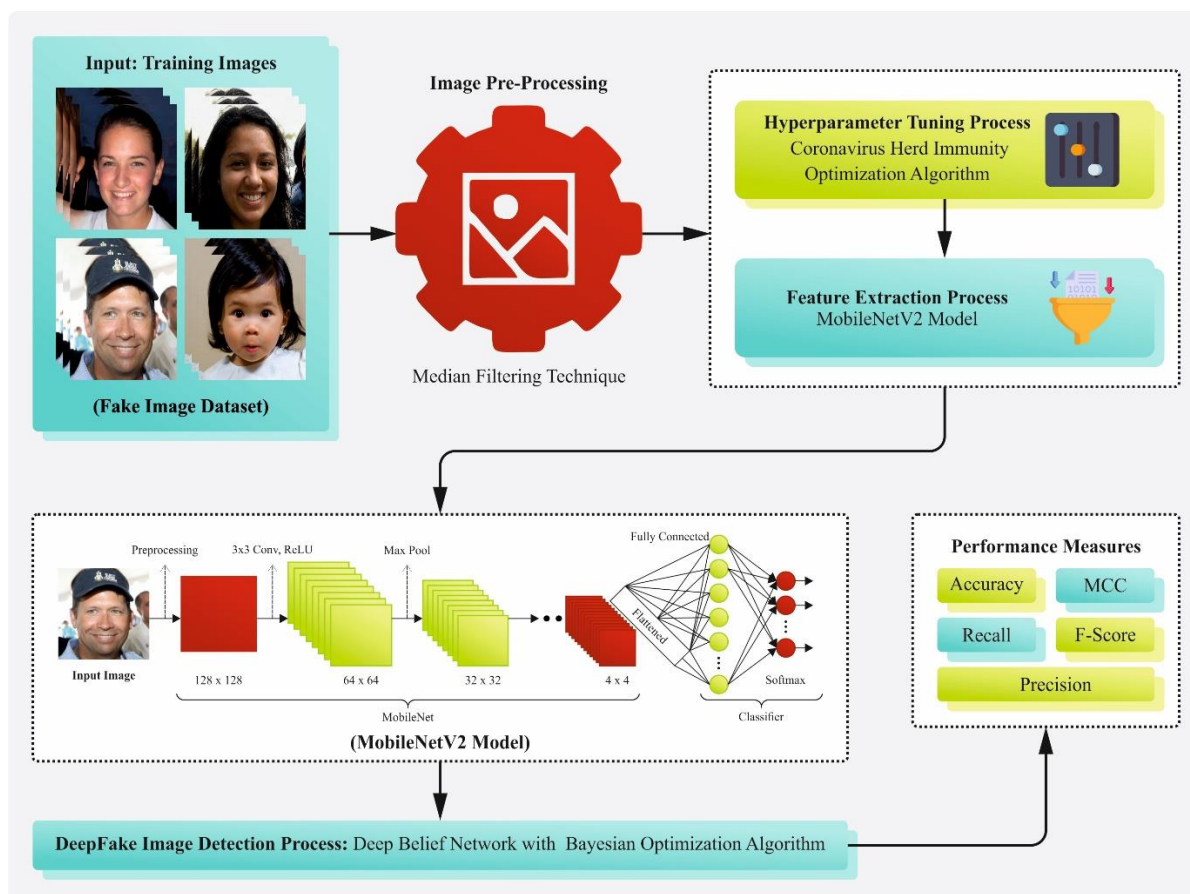


Figure 1. Overall workflow of the CHIODL-DIDC technique.

3.1. Image pre-processing

Initially, the CHIODL-DIDC technique uses an MF-based image-filtering approach [28]. Choosing the MF model for image filtering is justified due to its efficient performance in noise reduction while preserving significant image details. Unlike linear filters, the MF method operates non-linearly by replacing each pixel with the median value of its neighboring pixels, which is specifically effective at removing salt-and-pepper noise without blurring edges. This methodology outperforms in situations where other filters, namely Gaussian, might blur the image and lose crucial structural details. MF approach is computationally efficient and simple to implement, making it appropriate for real-time applications. Furthermore, it maintains the integrity of edge data, which is crucial for subsequent processing phases such as feature extraction and classification. Thus, MF-based filtering gives a robust balance between noise reduction and detail preservation, making it a superior choice for improving image quality in various CV tasks. Figure 2 illustrates the structure of the MF model.

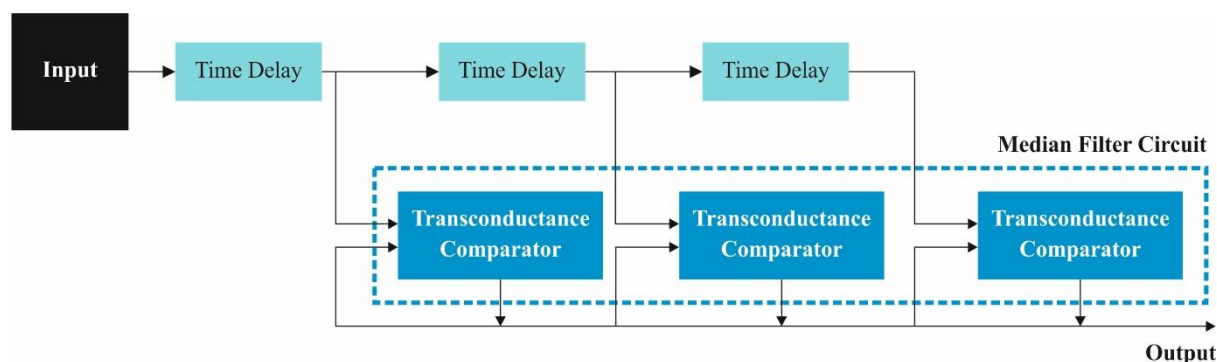


Figure 2. Structure of the MF model.

MF is a nonlinear image processing methodology that reduces noise while preserving edges in images. It replaces the pixel value with the median value of adjacent pixels within the definite kernel window. Unlike linear filters, like mean or Gaussian filtering, MF is beneficial at eliminating salt-and-pepper noise, where isolated pixels are very dark or brighter than their surroundings. This makes it a standard option for optimizing image quality in different applications, such as digital photography, medical imaging, and satellite images.

3.2. MobileNet2 model

Next, the CHIODL-DIDC technique applies the MobileNet2 model to extract feature vectors [29]. Applying the MobileNet2 method for feature vector extraction is advantageous due to its balance between effectualness and accuracy. MobileNet2 approach is designed with depthwise separable convolutions, which substantially reduce computational complexity while maintaining high performance, making it a precise choice for resource-constrained environments. This lightweight architecture enables faster processing and lower latency related to more complex models without compromising the quality of feature representation. The model's efficient utilization of the parameters and layers confirms that it can extract detailed and discriminative features from images, which is significant for robust visual recognition tasks. Furthermore, MobileNet2's pre-trained weights provide a strong starting point for transfer learning, improving its feature extraction efficiency across diverse applications. Its suitability for deployment on mobile and edge devices underscores its practical merit in real-world scenarios. Figure 3 depicts the architecture of the MobileNet2 method.

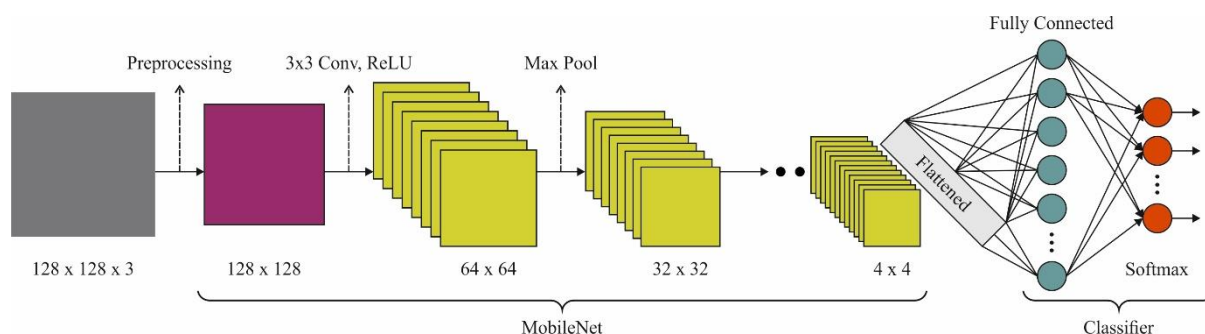


Figure 3. Architecture of MobileNet2 approach.

MobileNetV2 model helps as the feature extraction module in the structure, enhanced mainly for efficacy in embedded vision, mobile uses, and edge devices. This efficacy was vital for real skin disorder tasks of classification where fast analysis is vital, particularly for spreadable disorders such as chickenpox, measles, and monkeypox. If a few of these diseases are spreadable, there is a vital need for rapid, exact analysis to originate on-time treatment and separation processes. The structure of the MobileNetV2 method contains dual vital methods: Inverted residuals and depthwise separable convolutions. These methods contribute to their efficacy, making them suitable for real resource-constrained device studies. Then, the dual essential modules of the MobileNetV2 method are defined.

The MobileNetV2 method improves efficacy by utilizing depth-wise separable convolution, which distinguishes the convolutional procedure into dual layers, decreasing parameters and computations. The first layer, depth-wise convolution, uses a filter for every input channel. The second layer, pointwise convolution, unites these distinct channel outputs into a novel feature map utilizing 1×1 convolutional filters. This procedure permits the system to mix data from the dissimilar networks, enabling it to learn more compound features.

This procedure can be signified as below:

1) Depth-wise Convolution: Uses a filter f_d individually to every network of the input X . For every channel c , the output D is calculated as:

$$(D_c)_{i,j} = \sum_m \sum_n (f_{d,c})_{m,n} \cdot X_{c,i+m,j+n}, \forall c \in \{1, \dots, C\}, \quad (1)$$

where $(D_c)_{i,j}$ signifies the output for network c at location (i,j) , the filter f_d is used separately for every channel c of the input.

2) Pointwise Convolution: Unites the depthwise output utilizing a 1×1 convolutional layer f_p , resulting in the final output Y as:

$$Y_{i,j,k} = \sum_{m=1}^c (f_p)_{k,m} \cdot D_{i,j,m}, \forall k \in \{1, \dots, C'\}. \quad (2)$$

Here, $Y_{i,j,k}$ denotes the final output, uniting the depth-wise output. Correspondingly, C and C' represent the channel counts in the input and output from the pointwise convolution.

The depth-wise separable convolution efficiently reduces computation weight while holding the capability to procedure compound features. The complete process is stated as follows:

$$Y_{i,j,k} = \sum_{m=1}^c (f_p)_{k,m} \cdot \left(\sum_a \sum_b (f_d)_{a,b} \cdot X_{m,i+a,j+b} \right). \quad (3)$$

MobileNetV2 presents inverted residuals, conflicting with classical residual networks' transformation, expansion, and contraction bottleneck plans. This technique tracks a series of expansion, depth-wise convolution, and projection, improving the efficacy and feature of model protection. It is conveyed below:

$$\text{Expansion} \rightarrow \text{Depthwise Convolution} \rightarrow \text{Projection}. \quad (4)$$

This series in MobileNetV2 inverts the convention residual block technique, concentrating on 1st increasing and improving the feature networks for improved efficacy and feature representation. The facts of these procedures are defined below:

1) Expansion: An inverted residual block that is responsible for growing the number of networks in the feature maps. When C is the channel count in the feature of input maps X , the layer of expansion, the channel quantity turns into $\alpha \times C$. Furthermore, α denotes the expansion factor (generally larger than one). This process is signified below:

$$X \rightarrow X_{\text{expanded}} \text{ (Shape: } \alpha \times C \times H \times W \text{)}. \quad (5)$$

2) Depth-wise Convolution: Executed on the extended feature maps. This outcome in a novel feature maps D with a similar quantity of channels $\alpha \times C$ but classically with decreased spatial sizes $H' \times W'$. The process is mathematically signified as:

$$D_{i,j,m} = \sum_a \sum_b (f_d)_{a,b} \cdot X_{expanded,i+a,j+b,m}, \forall m \in \{1, \dots, \alpha \times C\}. \quad (6)$$

3) Projection: A 1x1 (pointwise convolution) plans the feature maps into a low-dimension space with C' networks. This process may be denoted as:

$$F_{i,j,k} = \sum_{m=1}^{\alpha \times C} D_{i,j,m} \cdot (f_p)_{k,m}, \forall k \in \{1, \dots, C'\}. \quad (7)$$

Here, F denotes the last feature maps with sizes $C' \times H' \times W'$.

3.3. CHIO-based hyperparameter tuning

Here, the hyperparameter tuning of the MobileNetv2 model is performed using the CHIO method [30]. In this segment, the mathematical method of the CHIO method was demonstrated at the levels mentioned below. Utilizing the CHIO method for hyperparameter tuning is highly advantageous due to its innovative optimization approach inspired by herd immunity dynamics. CHIO leverages a unique mechanism that simulates the spread of immunity in a population, enabling it to explore and exploit the hyperparameter space efficiently. This method balances exploration and exploitation effectively, leading to better convergence on optimal parameter settings. The ability of CHIO to escape local minima and adaptively fine-tune parameters enhances the model's performance and generalization. Its robust performance in optimizing complex models makes it particularly valuable for DL tasks, where precise hyperparameter tuning is crucial for achieving high accuracy and reliability. The CHIO model's efficiency in handling large search spaces and its novel approach provide significant advantages over traditional optimization techniques. Figure 4 demonstrates the structure of the CHIO model.

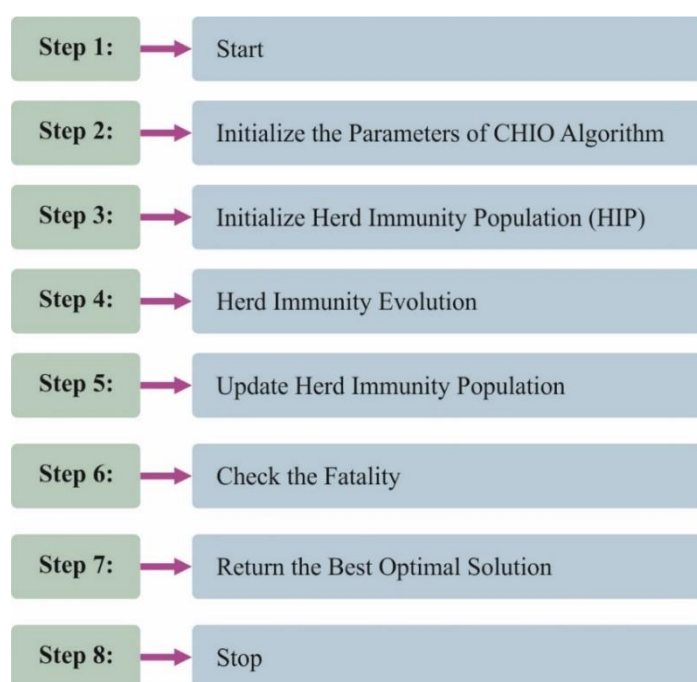


Figure 4. Architecture of the CHIO technique.

3.3.1. Initialize parameters

$$\min f(x) \quad x \in [lb, ub]. \quad (8)$$

Moreover, $f(x)$ denotes the objective function (rate of immunity) intended for every case.

$$x = (x_1, x_2, x_3, \dots, x_n), \quad (9)$$

wherein, x_i indicates the gene graded by i , and n signifies the entire quantity of the decision variable.

$$x_i \in [lb_i, ub_i]. \quad (10)$$

Now, lb_i and ub_i represent the lower and upper limit of the gene x_i , respectively.

The algorithm of CHIO contains dual control parameters:

- Basic reproduction rate (BR_r): It manages the operators of CHIO by distributing the virus between persons.
- Maximum diseased case age (Max_{Age}): Decide the diseased case position. If a case attains Max_{Age} , then it is retrieved or deceased.

The CHIO procedure contains an additional four parameters:

- Max_Iter : Signifies the highest iteration count.
- C_0 : This value embodies the quantity of main diseased cases that generally are one.
- HIS : Denotes the dimension of population.
- n : Signifies the size of the problem.

3.3.2. Producing a herd immunity population (HIP)

CHIO produces a group of individual cases, such as HIS . The group of produced individuals was kept as a dual-dimension matrix of $n \times HIS$ in the HIP as below:

$$HIP = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ x_1^{HIS} & x_2^{HIS} & \dots & x_n^{HIS} \end{bmatrix}, \quad (11)$$

whereas every row signifies x^j , which is computed as below:

$$\begin{aligned} x_i^j &= lb_i + (ub_i - lb_i) \times U(0,1) \quad \forall i = 1,2,3, \dots, n. \\ \forall j &= 1,2,3, \dots, HIS. \end{aligned} \quad (12)$$

The objective function was computed for every case utilizing Eq (8).

The fitness for every searching agent has been fixed as below:

$$\begin{aligned} S_j &= 0 \quad \forall j = 1,2,3, \dots, HIS, \\ A_j &= 0 \quad \forall j = 1,2,3, \dots, HIS, \end{aligned}$$

whereas S denotes the vector of status.

3.3.3. Corona-virus herd immunity progress

In this segment, the foremost development loop of CHIO was introduced. The gene (x_i^j) of the case (x^j) also endures the similar or turns into categorized as a diseased, immune, or vulnerable case as per the ratio of BR_r as below:

$$x_i^j(t+1) \leftarrow \left\{ \begin{array}{ll} x_i^j(t) & r \geq BR_r \\ C(x_i^j(t)) & r < \frac{1}{3} \times BR_r // \text{Infected case} \\ N(x_i^j(t)) & r < \frac{2}{3} \times BR_r // \text{susceptible case} \\ R(x_i^j(t)) & r < BR_r // \text{immuned case} \end{array} \right\}. \quad (13)$$

Here, r refers to a randomly generated value between 0 and 1.

For diseased cases:

$$\begin{aligned} x_i^j(t+1) &= C(x_i^j(t)), \\ C(x_i^j(t)) &= x_i^j(t) + r \times (x_i^j(t) - x_i^c(t)). \end{aligned} \quad (14)$$

Now, $x_i^j(t+1)$ signifies the novel gene, and $x_i(t)$ is nominated at random reliant on the vector status (S) from the case of unhealthy x^c as $c = \{i | S_i = 1\}$, and r denotes the arbitrarily produced value between 0 and 1.

For vulnerable cases:

$$\begin{aligned} x_i^j(t+1) &= N(x_i^j(t)), \\ N(x_i^j(t)) &= x_i^j(t) + r \times (x_i^j(t) - x_i^m(t)), \end{aligned} \quad (15)$$

whereas, $x_i^j(t+1)$ represents the novel gene, and $x_i^m(t)$ is selected at random dependent upon the vector of status (S) from every case of vulnerable x^m as $m = \{i | S_i = 0\}$, and r refers to the randomly formed among *zero* and *one*.

For immune cases:

$$\begin{aligned} x_i^j(t+1) &= R(x_i^j(t)), \\ R(x_i^j(t)) &= x_i^j(t) + r \times (x_i^j(t) - x_i^y(t)), \end{aligned} \quad (16)$$

wherein $x_i^j(t+1)$ refers to the novel gene and the value $x_i^y(t)$ is selected arbitrarily dependent upon the vector of status (S) from any case of immune x^y as $f(x^y) = \operatorname{argmin}_{j\{k|S_k=2\}} f(x^j)$.

3.3.4. Upgrade the HIP

For every produced case $x^j(t+1)$, the rate of immunity $f(x_i^j(t+1))$ is computed, and when the produced case $x^j(t+1)$ is superior to the existing one $x^j(t)$ like $f(x^j(t+1)) < f(x^j(t))$, the existing one is substituted by the produced one.

When the status vector ($S_j = 1$), the age vector (A_j) was enlarged by 1.

Depending upon the threshold of herd immunity, every case $x^j(t)$ of (S^j) was upgraded as follows:

$$S_j \leftarrow \left\{ \begin{array}{l} 1 \quad f(x^j(t+1)) < \frac{f(x^j(t+1))}{\Delta f(x)} \wedge S_j = 0 \wedge is_Corona(x^j(t+1)) \\ 2 \quad f(x^j(t+1)) > \frac{f(x^j(t+1))}{f(x)} \vee S_j = 1 \end{array} \right\}. \quad (17)$$

Following calculation: If, $\Delta f(x) = \frac{\sum_{i=1}^{HIS} f(x_i)}{HIS}$.

Whereas, $is_Corona(x^j(t+1))$ signifies a dual value equivalent to one in the novel $x^j(t+1)$, which obtained values from the diseased case, and $\Delta f(x)$ embodies the mean value.

3.3.5. Fatality cases

Suppose the rate of immunity $(f(x^j(t+1)))$ of the existing diseased cases does not increase for a precise iteration period. In that case, it is definite by the parameters Max_Age like $A_j \geq Max_Age$, and then it was measured deceased. Where it is restored from scratch utilizing the below-given expression:

$$x_i^j(t+1) = lb_i + (ub_i - lb_i) \times U(0,1). \quad (18)$$

If $\forall i = 1, 2, \dots, n$, S_j and A_j are fixed to 0.

3.3.6. Stop criterion

CHIO repeats the major loop till the highest iteration count is attained. The complete immune cases count furthermore to the vulnerable cases expertise the population and the diseased persons vanish.

3.4. Deepfake image detection using DBN and BOA models

Finally, the CHIODL-DIDC technique uses the DBN model for deepfake image detection. DBN is a multilayer probability-based generalization model widely used in complicated datasets' classification and feature learning [31]. Selecting DBNs and BOA models for deepfake image detection utilizes their unique strengths. DBNs, with their layered structure, outperform in learning complex features from large datasets, making them greatly efficient for detecting subtle patterns indicative of deepfakes. Their capability to capture hierarchical features and model high-dimensional data improves the accuracy of the detection. On the contrary, BOA is employed for optimizing hyperparameters, which substantially enhances the model's performance by effectually navigating the parameter space to find the best configuration. This methodology confirms that the DBN model operates at its highest potential. Compared to other techniques, the DBN model presents robust feature extraction capabilities, while the BOA model provides a systematic method for parameter tuning, delivering greater detection accuracy and effectualness.

DBN effectively combines the features of NNs with probabilistic graphical models, enabling the processing of high-dimensional and nonlinear information. DBNs are generalization modules that differentiate themselves by learning the joint distribution of information—an ability that exceeds

classification in the new data sample generation. The DBN architecture includes numerous hidden layers (HLs) that learn distinct representations and features of the data. This renders DBN especially useful in extracting features from complicated structures of data. The training method of DBN is a multi-stage system, which maximizes the characterization and capture of the input dataset. The layer-wise pretraining and global finetuning are the two processes of DBN. Figure 5 demonstrates the structure of DBN.

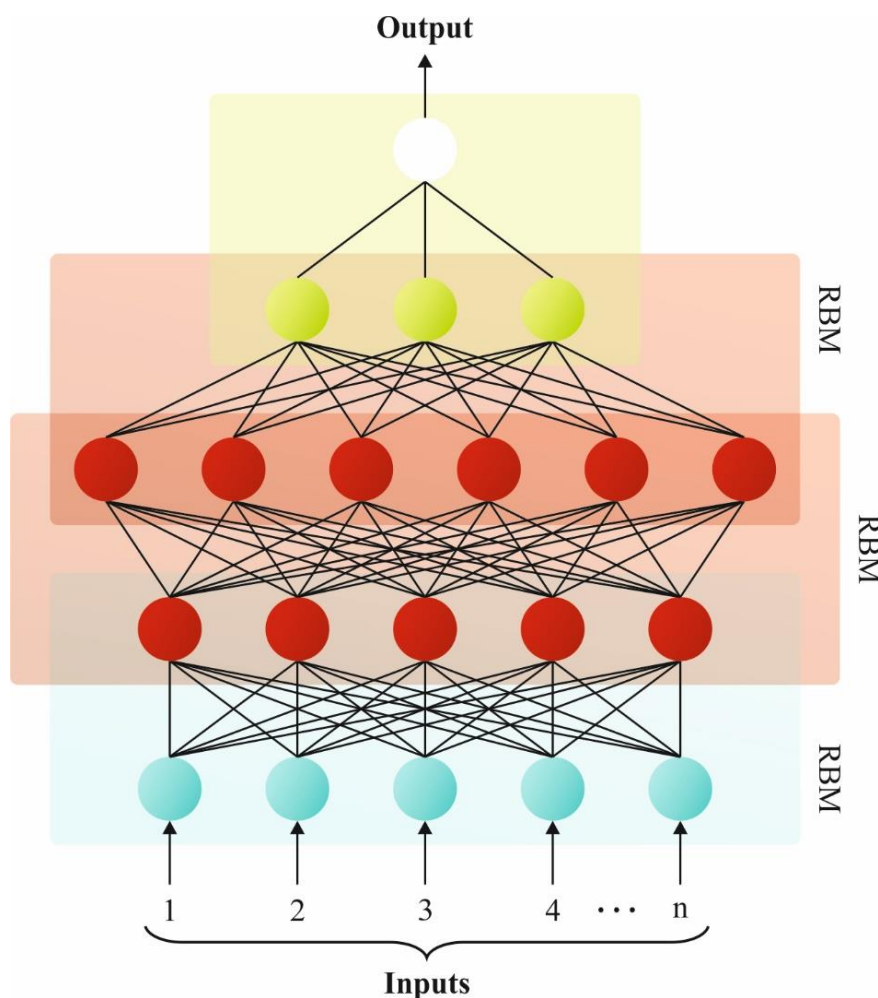


Figure 5. Architecture of the DBN model.

The Restricted Boltzmann Machines (RBMs) have multiple layers that capture features from different input datasets and are independently trained during layer-wise pretraining. This enables us to concentrate solely on learning feature representation from the input received. RBM's succeeding layer utilizes the previous layer's output as an input. As a result, the RBM in the bottom layer of the network is proficient at learning essential features. The top layer gradually extracts complex and abstract feature representations.

$$\begin{aligned}
 h_j &= w_{ij}v_i + b_j, \\
 v_i &= w_{ij}h_j + a_i,
 \end{aligned}
 \tag{19}$$

where h_j and b_j are the j^{th} neurons output and bias of HL; v_i and a_i are the i^{th} neurons input and the visible layer (VL) bias; and w_{ij} refers to the weight connecting among the neurons of the VL and HL.

$$E(v, h) = -\sum_{i=1}^n a_i v_i - \sum_{j=1}^m b_j h_j - \sum_{i=1}^n \sum_{j=1}^m v_j w_{ij} h_j, \quad (20)$$

$$P(v) = \frac{\sum_h e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}}, \quad (21)$$

where $P(v)$ and $E(v, h)$ are the probability distribution functions of VL of each layer in the RBM, h and m are the number of neurons in the VL and HLs, correspondingly.

$$P(v_i = 1|h) = \text{Sigmoid}(a_i + \sum_{j=1}^m h_j w_{ij}). \quad (22)$$

Assume the neuron's independence within the similar layer; it can evaluate the probability related to the weight $v_i = 1$

$$P(h_j = 1|v) = \text{Sigmoid}(b_j + \sum_{i=1}^n v_j w_{ij}). \quad (23)$$

Likewise, the probability related to the activation function $h_j = 1$ is evaluated:

$$\text{Sigmoid}(x) = \frac{1}{1+\exp(-x)}. \quad (24)$$

$$\theta^* = \operatorname{argmax} \sum_{t=1}^T \text{Inp}(v^t). \quad (25)$$

The RBM pretraining aims to attain optimum parameters, represented as parameters θ , to accomplish the DBN architecture with the fitting effect. Here, the model-generated probability and the sample statistical probability within the DBN are maximized. In Eq (25), T indicates the overall training instances.

Global finetuning involves the complete transformation of a DBN based on the Backpropagation (BP) model, following the pretraining of each layer. This stage aims to correct the biases and weights through the network, which enables better data prediction and representation. The BP model updates the parameter from the topmost layer Using the error gradient, thus improving the predictive capabilities of the network. The importance of the finetuning stage lies in its incorporation of single-layer learning, which ensures the effective representations of the DBN's data. The DBN obtains initial parameters after pre-training. The finetuning stage focuses on enhancing the model's fitting performance. Finetuning is used to adjust the initial parameter layer-wise downwards via a less labeled dataset attained during pretraining, beginning at the top layer of DBN. The SoftMax function is typically applied as the last classification by the BP model, which performs the finetuning process. Once the DBN model includes l RBMs, the output of pretraining is given below:

$$u^l(x) = \text{Sigmoid}(a^l + w^l u^{(l-1)}(x)). \quad (26)$$

The SoftMax function defines the DBN output by detecting the class with the maximum likelihood as the prediction classes. This efficiently transforms the output into the distribution probability through the predicted class, which facilitates the classifier task by highlighting the possible classes.

Initially, the DBN is determined. Next, the optimal selection of weight and threshold in the DBN model is randomized, which directly connects the iteration time to the network parameters and negatively affects the global search ability. This technique consistently navigates the optimization of DBN parameters in the training module, improving the predictive accuracy and the model's learning efficiency.

Next, the hyperparameter tuning of the DBN model takes place with the use of BOA. BOA is commonly utilized in ML, DL hyperparameter tuning, neural network structure search, engineering design optimizer, automatic ML, etc. [32]. Owing to its capability to discover the optimum

hyperparameter formation in moderately some iteration, hence securing computational sources, it is repeatedly used for issues that comprise discovering the finest hyperparameter setting. In this paper, to improve the hyperparameter of the forecast method, the BO is applied to optimize a parameter. During the state of a new function, the BO values the subsequent objective function distribution depending upon recognized data and preceding distribution and picks the subsequent sample point depending upon the distribution.

The exact steps are given below:

- 1) Make the initial sample point at random in the optimizer limit of the hyperparameter model. Input this step into the Gaussian procedure and train the subsequent method. Estimate and alter the Gaussian method built on the value of loss output by the objective function of models, permitting the method for estimating the accurate function distribution;
- 2) once the Gaussian method is assessed and adapted, utilize the sample function to select the subsequent set of sample points to input into the technique for training. Acquire novel values of output for the model's loss of objective function by upgrading the Gaussian method and the sample sets;
- 3) when the loss value of nominated sample points of new encounters the needs, end the procedure and yield the presently nominated finest parameter sequence beside by the equivalent loss value of the objective function;
- 4) when the loss value of the nominated sample points of new doesn't meet the needs, upgrade the sample point in the sample sets and go back to Stage 2. Then, go again to assess and alter the Gaussian method until the conditions are met.

The BOA method derives an FF to obtain better classification results. It defines a positive integer to characterize the candidate solution's best performances. Now, the decline of the classifier error rate can be implicit by FF.

$$\begin{aligned} \text{fitness}(x_i) &= \text{ClassifierErrorRate}(x_i) \\ &= \frac{\text{No. of misclassified samples}}{\text{Total No. of samples}} \times 100. \end{aligned} \quad (27)$$

4. Experimental validation

The performance analysis of the CHIIDL-DIDC model is studied using 140k real and fake faces datasets from the Kaggle repository [33]. The dataset holds 2500 real and 2400 fake images, as defined in Table 1. Figure 6 portrays the sample of real and fake images.

Table 1. Details on database.

Classes	No. of samples
Real	2500
Fake	2400
Total samples	4900



Figure 6. Sample images (a) real and (b) fake.

Figure 7 depicts the confusion matrices created by the CHIODL-DIDC method on different epoch counts. The outcome shows that the CHIODL-DIDC method effectively detects the real and fake samples under each class label. The outcomes portray the performance of the technique across diverse epochs. At Epoch 500, the “Real” class has a TP of 99.06% and an FN of 0.47%, while the “Fake” class has a TP of 97.22% and an FN of 1.39%. By Epoch 1000, the “Real” class TP improves to 99.66% with an FN of 0.16%, and the “Fake” class TP is 94.96% with an FN of 2.59%. Epoch 1500 shows similar trends with TP for the “Real” class at 99.57%. At Epoch 2500, TP for “Real” is 99.70% with an FN of 0.20%, and TP for “Fake” is 94.17% with an FN of 3.02%. By Epoch 3000, TP for “Real” is 99.65% with an FN of 0.14%, while TP for “Fake” ranges from 91.68% to 93.26%, with FNs between 3.53% and 4.43%.

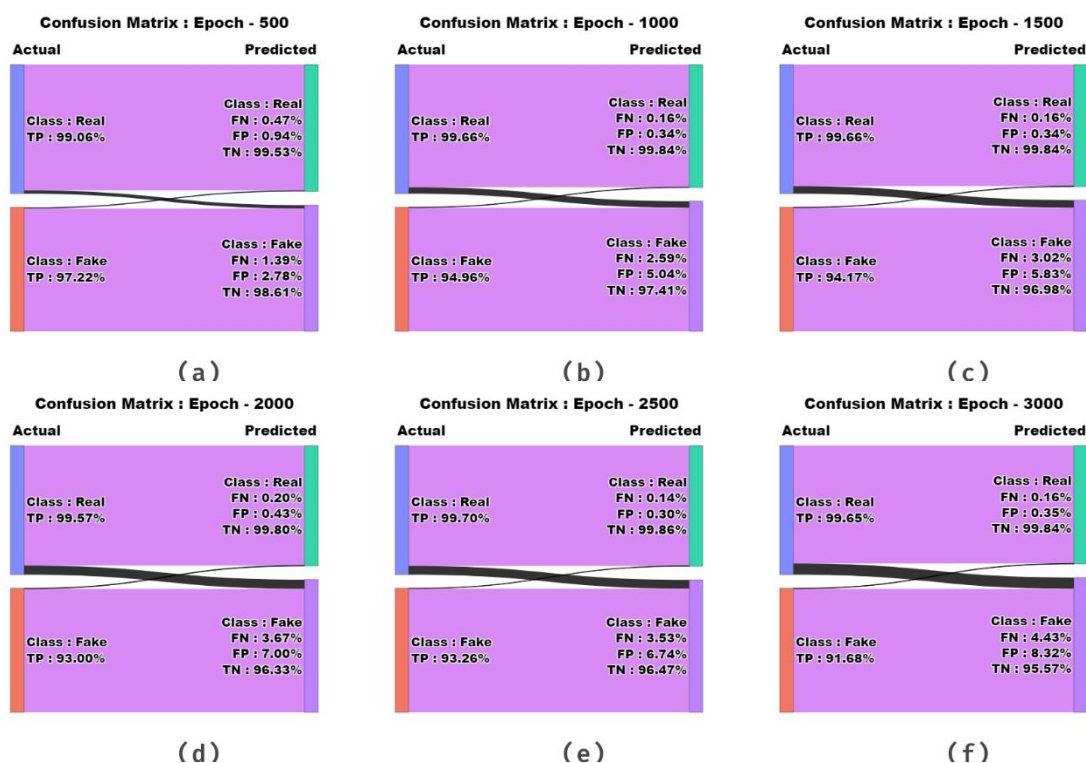


Figure 7. Confusion matrices of CHIODL-DIDC model (a–f) Epochs 500–3000.

In Table 2 and Figure 8, extensive fake image detection results of the CHIODL-DIDC technique are reported. The results stated that the CHIODL-DIDC technique reaches effectual performance under all epochs. With 500 epoch counts, the CHIODL-DIDC methodology obtains an average $accu_y$ of 98.16%, $prec_n$ of 98.14%, $reca_l$ of 98.16%, F_{score} of 98.14%, and MCC of 96.30%. Additionally, with 1000 epochs, the CHIODL-DIDC method obtains an average $accu_y$ of 97.29%, $prec_n$ of 97.31%, $reca_l$ of 97.29%, F_{score} of 97.24%, and MCC of 94.60%. Besides, with 1500 epochs, the CHIODL-DIDC method obtains an average $accu_y$ of 96.87%, $prec_n$ of 96.92%, $reca_l$ of 96.87%, F_{score} of 96.82%, and MCC of 93.79%. Likewise, with 2000 epoch counts, the CHIODL-DIDC approach gains an average $accu_y$ of 96.18%, $prec_n$ of 96.28%, $reca_l$ of 96.19%, F_{score} of 96.12%, and MCC of 92.48%. Last, with 3000 epochs, the CHIODL-DIDC method obtains an average $accu_y$ of 95.49%, $prec_n$ of 95.67%, $reca_l$ of 95.49%, F_{score} of 95.41%, and MCC of 91.16%.

Table 2. Fake image detection outcome of CHIODL-DIDC technique under various epochs.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	MCC
Epoch-500					
Real	97.28	99.06	97.28	98.16	96.30
Fake	99.04	97.22	99.04	98.12	96.30
Average	98.16	98.14	98.16	98.14	96.30
Epoch-1000					
Real	94.92	99.66	94.92	97.23	94.60
Fake	99.67	94.96	99.67	97.26	94.60
Average	97.29	97.31	97.29	97.24	94.60
Epoch-1500					
Real	94.08	99.66	94.08	96.79	93.79
Fake	99.67	94.17	99.67	96.84	93.79
Average	96.87	96.92	96.87	96.82	93.79
Epoch-2000					
Real	92.80	99.57	92.80	96.07	92.48
Fake	99.58	93.00	99.58	96.18	92.48
Average	96.19	96.28	96.19	96.12	92.48
Epoch-2500					
Real	93.08	99.70	93.08	96.28	92.87
Fake	99.71	93.26	99.71	96.38	92.87
Average	96.39	96.48	96.39	96.33	92.87
Epoch-3000					
Real	91.32	99.65	91.32	95.30	91.16
Fake	99.67	91.68	99.67	95.51	91.16
Average	95.49	95.67	95.49	95.41	91.16

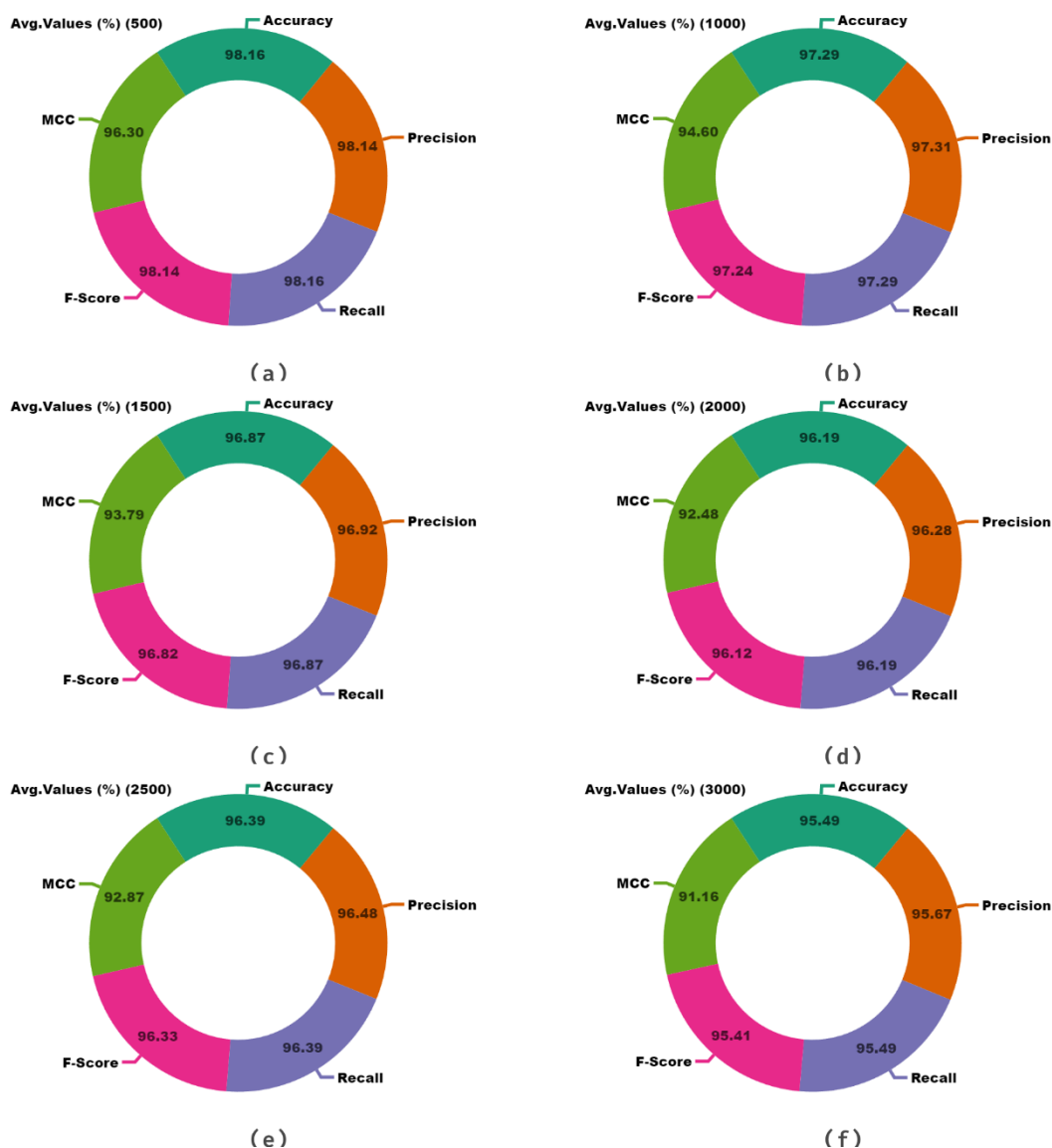


Figure 8. Average outcome of CHIODL-DIDC technique (a–f) Epochs 500–3000.

The classifier outcomes of the CHIODL-DIDC method are graphically shown in Figure 9 for validation accuracy (VALA) and training accuracy (TRAA) curves over dissimilar epoch counts. The figure shows valuable insight into the behavior of the CHIODL-DIDC method over different epochs, demonstrating its generalization abilities and learning process. Notably, the figure indicates the consistent development in the TRAA and VALA with maximum epoch counts. It confirms the wide-ranging nature of the CHIODL-DIDC approach within the pattern recognition technique on both datasets. The increasing tendency in VALA describes the capability of the CHIODL-DIDC model for adapting the TRA dataset. Also, it outshines in providing the correct classifier of the unseen dataset, indicating robust generalization ability.

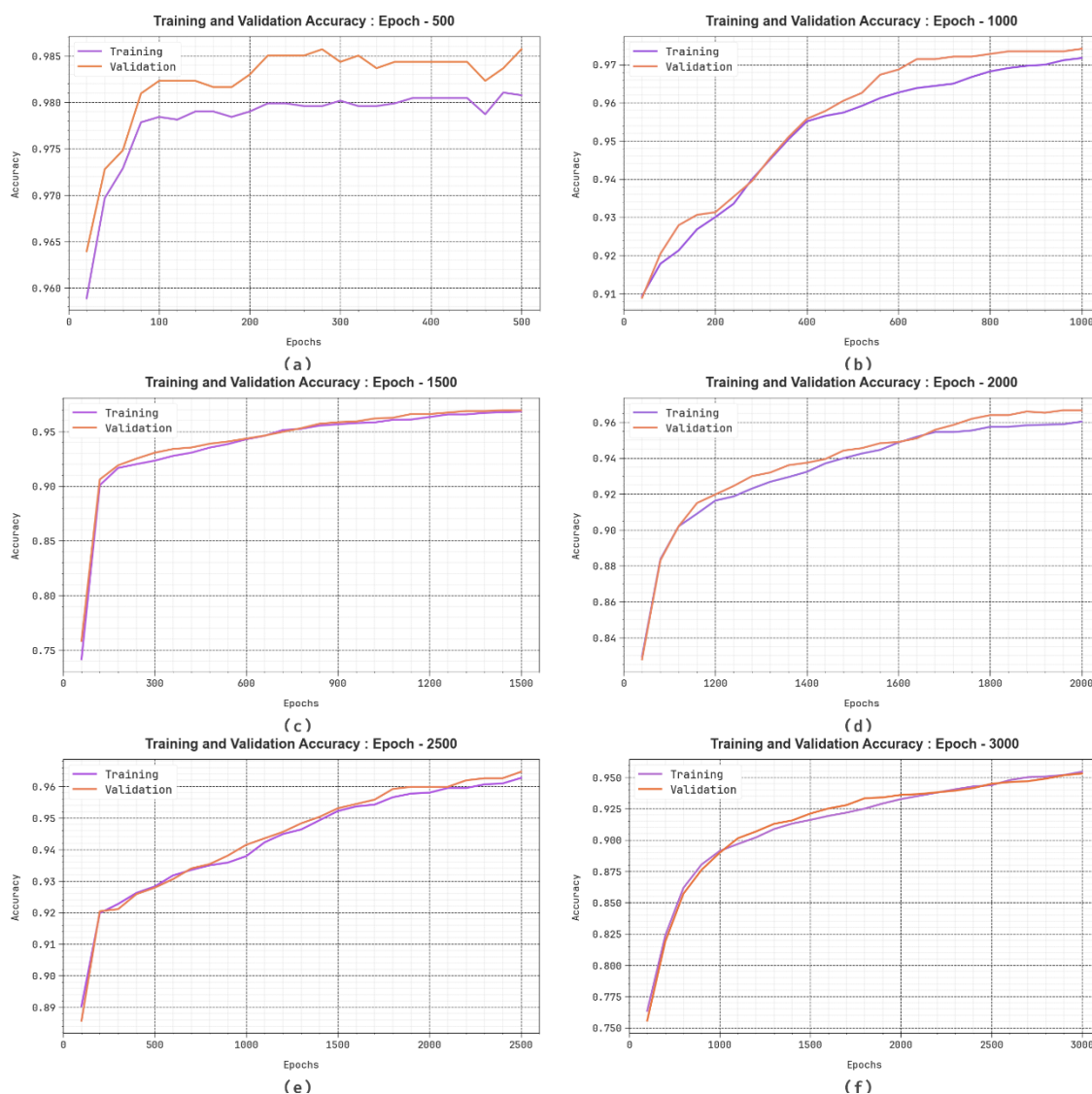


Figure 9. $Accu_y$ curve of CHIODL-DIDC technique (a–f) Epochs 500–3000.

Figure 10 exhibits a detailed review of the validation loss (VALL) and training loss (TRLA) results of the CHIODL-DIDC technique across several epoch counts. The gradual decrease in TRLA underlines the CHIODL-DIDC technique's improvement in the weights and lowering of the classification error on both datasets. The figure clearly shows the CHIODL-DIDC methods' association with the TRA dataset, which emphasizes its capacity for capturing patterns within both data. The CHIODL-DIDC method frequently improves its parameters in minimizing the discrepancies between the real TRA and prediction classes.

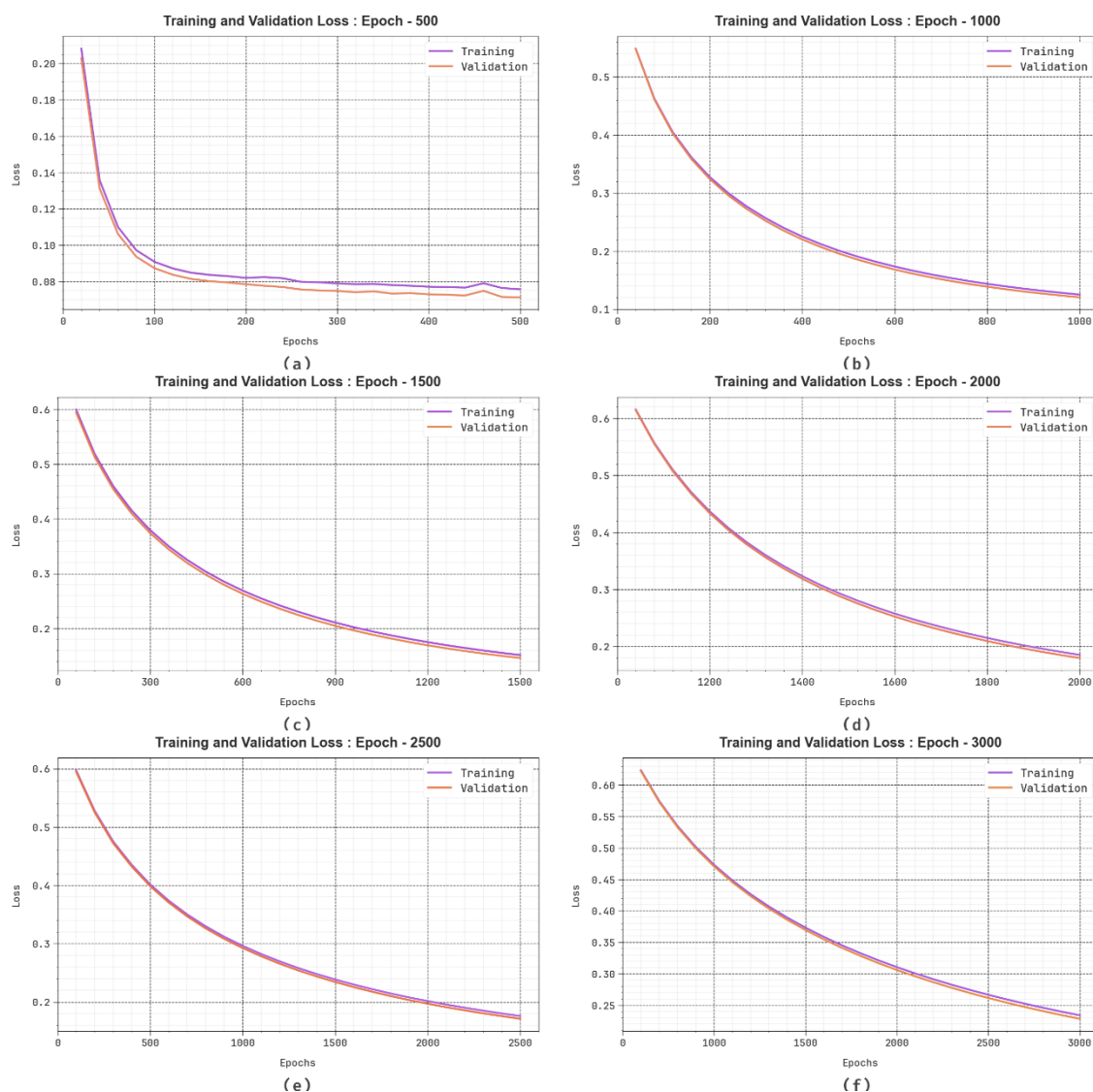


Figure 10. Loss curve of CHIODL-DIDC method (a–f) Epochs 500–3000.

Examining the PR analysis, as shown in Figure 11, the outcomes confirmed that the CHIODL-DIDC technique gradually achieves the maximum PR performance through all the classes over various epochs. It confirms the better capabilities of the CHIODL-DIDC method in detecting numerous class labels, showing the ability to recognize classes.

Moreover, in Figure 12, the ROC investigation produced by the CHIODL-DIDC method outperforms the classifier of different labels over different epochs. It offers a complete review of the trade-off between FRP and TPR over epoch counts and threshold values. This figure emphasizes the superior classification outcomes of the CHIODL-DIDC technique at each class label, describing the efficiency of responding to various classifier problems.

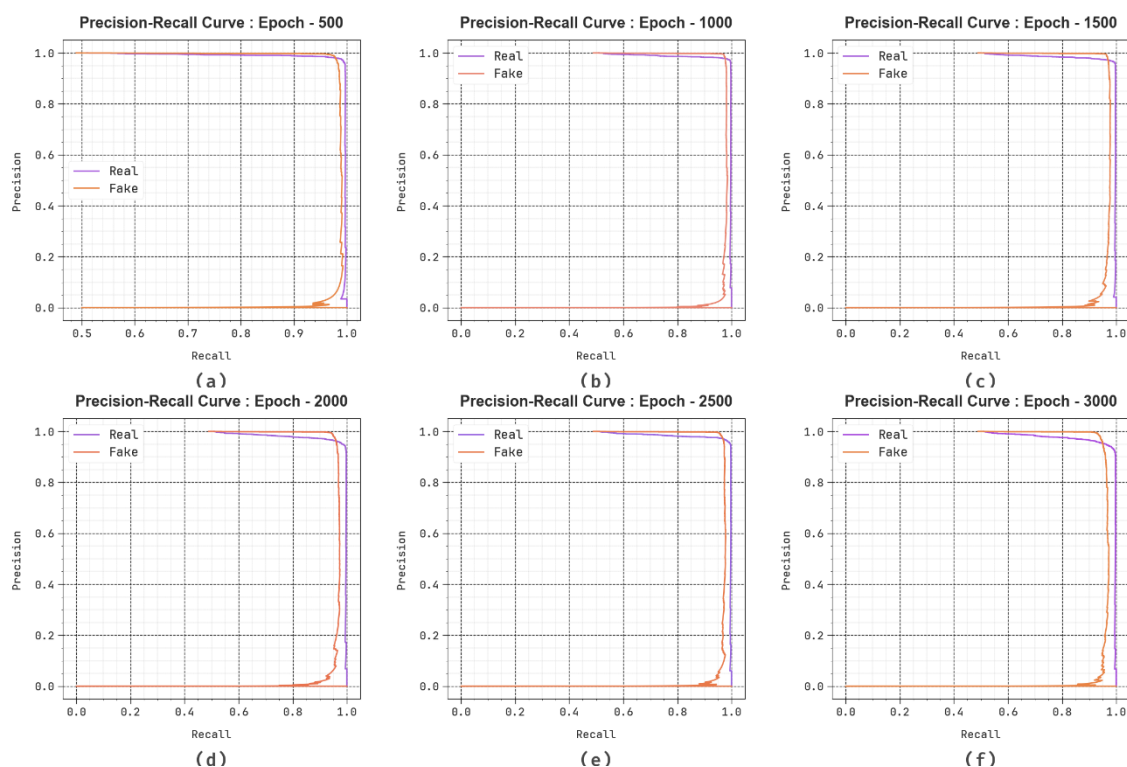


Figure 11. PR curve of CHIODL-DIDC technique (a–f) Epochs 500–3000.

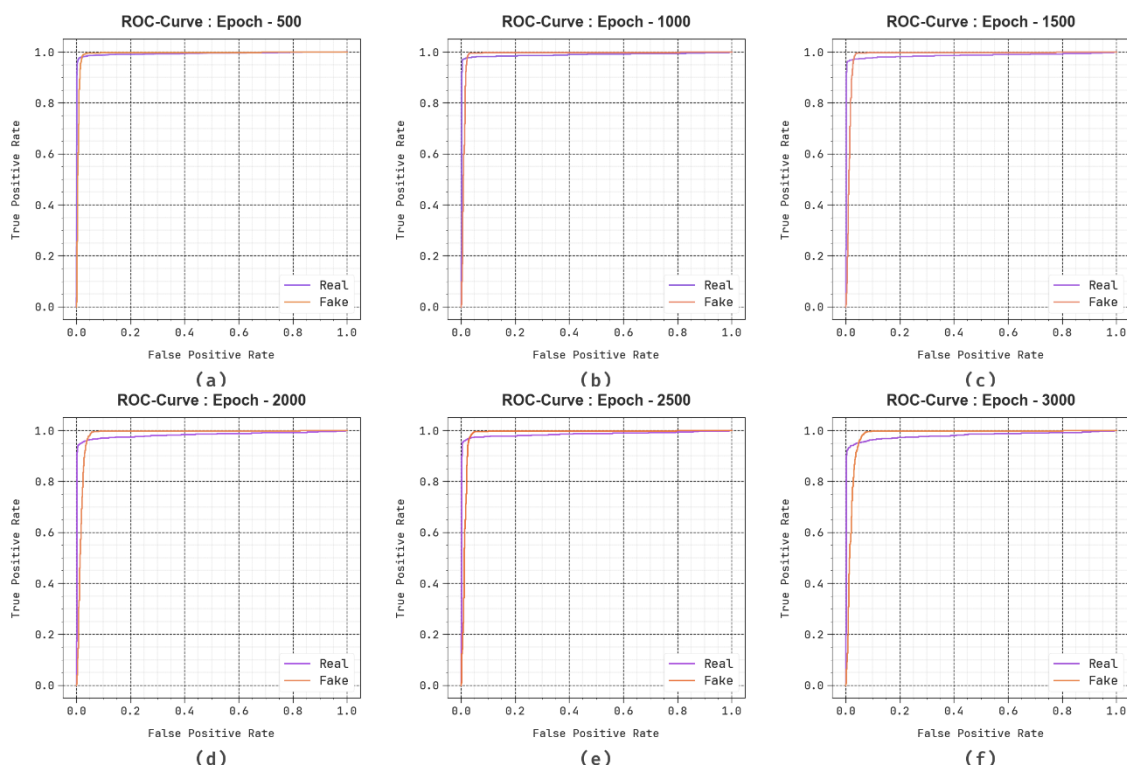


Figure 12. ROC curve of CHIODL-DIDC method (a–f) Epochs 500–3000.

The deepfake image detection results of the CHIODL-DIDC technique are compared with existing studies in Table 3 and Figure 13 [34,35]. The results highlighted that the Cooccurrence model

has reached poor performance, whereas the GRAMNet model has gained slightly boosted results. Moreover, the ResNet50V2, DenseNet201, and MMGANGuard models have obtained closer performance. However, the CHIODL-DIDC technique surpassed existing models with a maximum $accu_y$ of 98.16%, $prec_n$ of 98.14%, $reca_l$ of 98.16%, and F_{score} of 98.14%.

Table 3. Comparative analysis of CHIODL-DIDC technique with existing approaches [34,35].

Model	$Accu_y$	$Prec_n$	$Reca_l$	F_{score}
GRAM Net	94.61	93.02	97.02	95.02
Cooccurrence	92.61	91.01	94.02	93.01
ResNet50V2	96.01	98.02	95.01	96.02
DenseNet201	96.02	95.01	98.02	96.01
MMGANGuard	97.01	96.02	98.02	97.02
Xception	92.29	93.74	97.16	96.70
DSP-FWA	93.62	95.32	93.55	93.58
B4ATT	92.37	93.47	97.14	96.54
E-TAD	93.45	93.51	93.96	97.73
MCX-API	97.80	96.83	97.37	92.75
CHIODL-DIDC	98.16	98.14	98.16	98.14

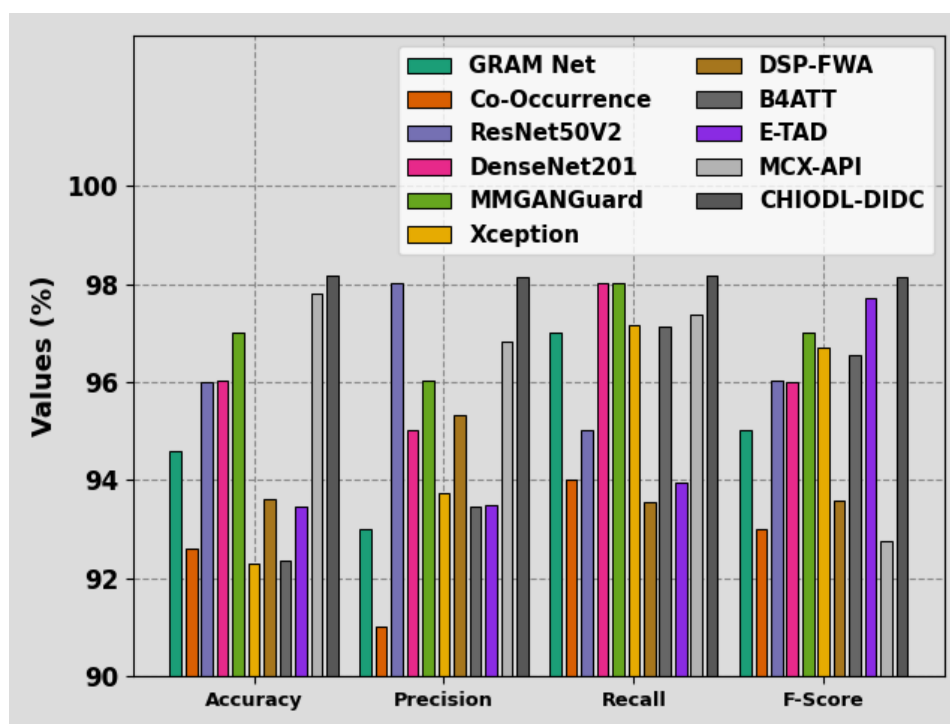


Figure 13. Comparative analysis of CHIODL-DIDC technique with existing approaches.

The comparative computational time (CT) examination of the CHIODL-DIDC method with existing studies is shown in Table 4 and Figure 14. The outcomes highlighted that the co-occurrence method has obtained poor performance. In contrast, the GRAMNet model has obtained slightly lesser outcomes. Moreover, the ResNet50V2, DenseNet201, and MMGANGuard techniques have attained closer performance. However, the CHIODL-DIDC method surpassed existing approaches with a

minimum CT of 6.71s. Additionally, techniques such as Xception, DSP-FWA, B4ATT, E-TAD, and MCX-API portrayed moderate outcome of 8.64s, 11.42s, 11.56s, 9.26s, and 9.03s. Thus, the CHIODL-DIDC method is employed for a superior detection process.

Table 4. CT analysis of CHIODL-DIDC technique with existing approaches.

Model	CT (Sec)
GRAM Net	8.64
Cooccurrence	11.42
ResNet50V2	11.56
DenseNet201	9.26
MMGANGuard	9.03
Xception	8.64
DSP-FWA	11.42
B4ATT	11.56
E-TAD	9.26
MCX-API	9.03
CHIODL-DIDC	6.71

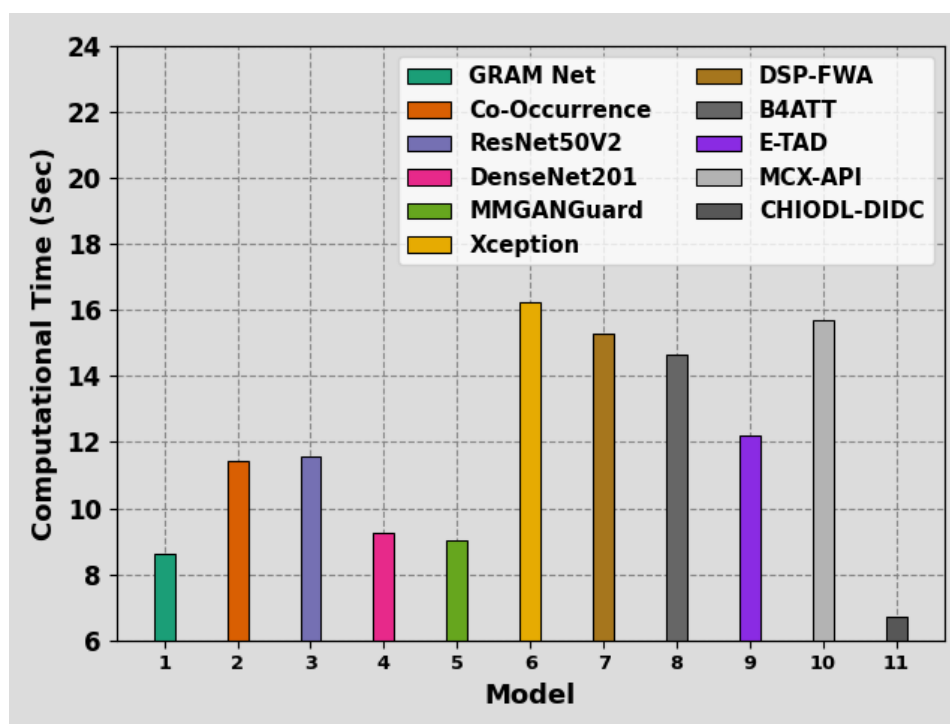


Figure 14. CT analysis of CHIODL-DIDC technique with existing approaches.

5. Conclusions

In this study, a new CHIODL-DIDC method is introduced. The CHIODL-DIDC method aims to detect and classify the existence of fake images. To perform this, the CHIODL-DIDC approach contains various stages, namely image preprocessing, MobileNetv2-based feature extractor, CHIO-based parameter tuning, and DBN-based image detection. Initially, the CHIODL-DIDC technique uses

an MF-based image filtering approach. Besides, the CHIODL-DIDC technique utilizes the MobileNetv2 model for extracting feature vectors. Moreover, the hyperparameter tuning of the MobileNetv2 technique is performed using the CHIO model. For deepfake image detection, the CHIODL-DIDC technique uses the DBN model. Finally, the DBN technique uses the BOA model for effectual hyperparameter selection. The empirical analysis of the CHIODL-DIDC technique is examined by utilizing a benchmark fake image dataset. The performance validation of the CHIODL-DIDC technique illustrates a superior accuracy value of 98.16% over other models. The limitations of the CHIODL-DIDC approach comprise potential threats with high-dimensional data and varying image quality, which may affect the accuracy of preprocessing and feature extraction stages. Furthermore, the dependence on specific tuning methods could pave the way to suboptimal accomplishment in diverse scenarios. Researchers should concentrate on improving the adaptability of the methodology to handle a wider range of image qualities and conditions and exploring alternative optimization methods to enhance overall system robustness and effectualness. Expanding the technique to comprise more advanced ML methods and integrating real-time processing capabilities may improve its applicability.

Author contributions

Wahida Mansouri: Conceptualization, validation, resources, writing-original draft preparation, supervision; Amal Alshardan: Conceptualization, validation, investigation, writing-original draft preparation, visualization, funding acquisition; Nazir Ahmad: Methodology, software, writing-original draft preparation, writing-review and editing, project administration; Nuha Alruwais: Methodology, formal analysis, writing-original draft preparation, writing-review and editing, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/51/45. Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2024R507), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Researchers Supporting Project number (RSPD2024R608), King Saud University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at Northern Border University, Arar, KSA for funding this research work through the project number “NBU-FPEJ-2024- 2899-01”.

Conflict of interest

The authors declare that they have no conflict of interest. The manuscript was written through the contributions of all authors. All authors have approved the final version of the manuscript.

Data availability statement

The data supporting this study’s findings are openly available in the Kaggle repository at <https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces> [33].

References

1. S. Solaiyappan, Y. X. Wen, Machine learning based medical image deepfake detection: A comparative study, *Mach. Learn. Appl.*, **8** (2022), 100298. <https://doi.org/10.1016/j.mlwa.2022.100298>
2. T. Zhang, Deepfake generation and detection, a survey, *Multimed. Tools Appl.*, **81** (2022), 6259–6276. <https://doi.org/10.1007/s11042-021-11733-y>
3. C. C. Hsu, C. Y. Lee, Y. X. Zhuang, *Learning to detect fake face images in the wild*, In 2018 international symposium on computers, consumers, and control (IS3C), IEEE, 2018, 388–391. <https://doi.org/10.1109/IS3C.2018.00104>
4. H. Chi, M. Peng, *Toward robust deep learning systems against deepfake for digital forensics*, In Cybersecurity and HighPerformance Computing Environments, Chapman and Hall/CRC, 2022, 309–331.
5. M. Tanaka, S. Shiota, H. Kiya, A detection method of operating fake images using robust hashing, *J. Imaging*, **7** (2021), 134. <https://doi.org/10.3390/jimaging7080134>
6. Z. Liu, X. Qi, P. H. S. Torr, *Global texture enhancement for fake face detection in the wild*, In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, 8060–8069. <https://doi.org/10.1109/CVPR42600.2020.00808>
7. J. Yang, A. Li, S. Xiao, W. Lu, X. Gao, Mtdnet: Learning to detect deepfakes images by multi-scale texture difference, *IEEE T. Inf. Foren. Sec.*, **16** (2021), 4234–4245. <https://doi.org/10.1109/TIFS.2021.3102487>
8. L. Nataraj, T. Manhar, S. Chandrasekaran, A. Flenner, J. H. Bappy, A. K. R. Chowdhury, et al., Detecting gan generated fake images using cooccurrence matrices, *arXiv Preprint*, 2019. <https://doi.org/10.2352/ISSN.2470-1173.2019.5.MWSF-532>
9. J. Pu, N. Mangaokar, L. Kelly, P. Bhattacharya, K. Sundaram, M. Javed, et al., *Deepfake videos in the wild: Analysis and detection*, In: Proceedings of the Web Conference, **2021** (2021), 981–992. <https://doi.org/10.1145/3442381.34499>
10. M. Masood, M. Nawaz, K. Mahmood, A. Javed, A. Irtaza, H. Malik, Deepfakes generation and detection: Stateof-the-art, open challenges, countermeasures, and way forward, *Appl. Intell.*, **53** (2023), 3974–4026. <https://doi.org/10.1007/s10489-022-03766-z>
11. R. D. Sushir, D. G. Wakde, S. S. Bhutada, Enhanced blind image forgery detection using an accurate deep learning based hybrid DCCA and ADFC, *Multimed. Tools Appl.*, **83** (2024), 1725–1752. <https://doi.org/10.1007/s11042-023-15475-x>
12. S. T. Suganthi, M. U. A. Ayoobkhan, N. Bacanin, K. Venkatachalam, H. Štěpán, T. Pavel, Deep learning model for deep fake face recognition and detection, *PeerJ Comput. Sci.*, **8** (2022), e881. <https://doi.org/10.7717/peerj-cs.881>
13. S. Ghosh, S. Kayal, M. Malakar, A. Sengupta, S. Srimani, A. Das, *FaceDig: A deep neural network-based fake image detection scheme*, In: Emerging Electronic Devices, Circuits and Systems: Select Proceedings of EEDCS Workshop Held in Conjunction with ISDCS, Singapore: Springer Nature, 2022, 395–404. https://doi.org/10.1007/978-981-99-0055-8_33
14. E. Hashmi, S. Y. Yayilgan, M. M. Yamin, S. Ali, M. Abomhara, Advancing fake news detection: Hybrid deep learning with fasttext and explainable AI, *IEEE Access*, 2024. <https://doi.org/10.1109/ACCESS.2024.3381038>

15. A. Boyd, P. Tinsley, K. W. Bowyer, A. Czajka, *Cyborg: Blending human saliency into the loss improves deep learning-based synthetic face detection*, In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, 6108–6117. <https://doi.org/10.1109/WACV56688.2023.00605>
16. S. Li, V. Dutta, X. He, T. Matsumaru, Deep learning based one-class detection system for fake faces generated by GAN network, *Sensors*, **22** (2022), 7767. <https://doi.org/10.3390/s22207767>
17. S. M. Dwivedi, S. B. Wankhade, Deep learning based semantic model for multimodal fake news detection, *Int. J. Intell. Eng. Syst.*, **17** (2024). <https://doi.org/10.22266/ijies2024.0229.55>
18. C. Mallick, S. Mishra, M. R. Senapati, A cooperative deep learning model for fake news detection in online social networks, *J. Amb. Intel. Hum. Comp.*, **14** (2023), 4451–4460. <https://doi.org/10.1007/s12652-023-04562-4>
19. I. Zhang, D. Zhao, C. P. Lim, H. Asadi, H. Huang, Y. Yu, et al., Video deepfake classification using particle swarm optimization-based evolving ensemble models, *Knowl.-Based Syst.*, **289** (2024), 111461. <https://doi.org/10.1016/j.knosys.2024.111461>
20. P. Chen, M. Xu, J. Qi, DeepFake detection against adversarial examples based on D-VAEGAN, *IET Image Process.*, **18** (2024), 615–626. <https://doi.org/10.1049/ipr2.12973>
21. K. Omar, R. H. Sakr, M. F. Alrahmawy, An ensemble of CNNs with self-attention mechanism for DeepFake video detection, *Neural Comput. Appl.*, **36** (2024), 2749–2765. <https://doi.org/10.1007/s00521-023-09196-3>
22. L. Yang, W. Shu, Y. Wang, Z. Lian, *Integration model of deep forgery video detection based on rPPG and spatiotemporal signal*, In International Conference on Green, Pervasive, and Cloud Computing, Singapore: Springer Nature, 2023, 113–127. https://doi.org/10.1007/978-981-99-9893-7_9
23. A. A. Hasanaath, H. Luqman, R. F. Katib, S. Anwar, FSBI: Deepfakes detection with frequency enhanced self-blended images, *Hamzah and KATIB, RAED FAROUQ and Anwar, Saeed*, 2024. <https://doi.org/10.2139/ssrn.4869258>
24. P. K. Rangarajan, M. Sukesh, D. M. Abinandhini, Y. Jaikanth, *Detecting AI-generated images with CNN and Interpretation using Explainable AI*, In 2024 IEEE International Conference on Contemporary Computing and Communications (InC4), IEEE, **1** (2024), 1–6. <https://doi.org/10.1109/InC460750.2024.10649158>
25. H. Ilyas, A. Javed, K. M. Malik, AVFakeNet: A unified end-to-end dense swin transformer deep learning model for audio-visual deepfakes detection, *Appl. Soft Comput.*, **136** (2023), 110124. <https://doi.org/10.1016/j.asoc.2023.110124>
26. T. Ige, C. Kiekintveld, A. Piplai, Deep learning-based speech and vision synthesis to improve phishing attack detection through a multilayer adaptive framework, *arXiv Preprint*, 2024. <https://doi.org/10.20944/preprints202402.1557.v1>
27. B. Liu, B. Liu, M. Ding, T. Zhu, *Detection of diffusion model-generated faces by assessing smoothness and noise tolerance*, In 2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), IEEE, 2024, 1–6. <https://doi.org/10.1109/BMSB62888.2024.10608232>
28. A. Noor, Y. Zhao, R. Khan, L. Wu, F. Y. Abdalla, Median filters combined with denoising convolutional neural network for Gaussian and impulse noises, *Multimed. Tools Appl.*, **79** (2020), 18553–18568. <https://doi.org/10.1007/s11042-020-08657-4>
29. A. D. Raha, M. Gain, R. Debnath, A. Adhikary, Y. Qiao, M. M. Hassan, et al., Attention to Monkeypox: An interpretable Monkeypox detection technique using attention mechanism, *IEEE Access*, 2024.

30. H. Selim, A. Y. Haikal, L. M. Labib, M. M. Saafan, MCHIAO: A modified coronavirus herd immunity-Aquila optimization algorithm based on chaotic behavior for solving engineering problems, *Neural Comput. Appl.*, 2024, 1–85. <https://doi.org/10.1007/s00521-024-09533-0>
31. M. Guo, R. Lv, Z. Miao, F. Fei, Z. Fu, E. Wu, et al., Load forecasting and operation optimization of ice-storage air conditioners based on improved deep-belief network, *Processes*, **12** (2024), 523. <https://doi.org/10.3390/pr12030523>
32. R. Zhou, S. Qiu, M. Li, S. Meng, Q. Zhang, Short-term air traffic flow prediction based on CEEMD-LSTM of Bayesian optimization and differential processing, *Electronics*, **13** (2024), 1896. <https://doi.org/10.3390/electronics13101896>
33. <https://www.kaggle.com/datasets/xhlulu/140k-real-and-fake-faces>
34. S. A. Raza, U. Habib, M. Usman, A. A. Cheema, M. S. Khan, MMGANGuard: A robust approach for detecting fake images generated by GANs using multi-model techniques, *IEEE Access*, 2024. <https://doi.org/10.1109/ACCESS.2024.3393842>
35. J. Gao, M. Micheletto, G. Orrù, S. Concas, X. Feng, G. L. Marcialis, et al., Texture and artifact decomposition for improving generalization in deep-learning-based deepfake detection, *Eng. Appl. Artif. Intel.*, **133** (2024), 108450. <https://doi.org/10.1016/j.engappai.2024.108450>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)