Optimized Gaussian Convolutional Neural Network Framework for Enhanced Detection of Deepfakes in Digital Media

Ahmed Alhussen[1,*]

[1]Department of Computer Engineering, College of Computer and Information Sciences,
Majmaah University, Al-Majmaah 11952, Saudi Arabia
Email: aa.alhussen@mu.edu.sa

Abstract

With the latest developments in computer vision, processing, accurate deepfakes (DF) require powerful tools. Recent research has developed a useful technique for identifying DFs in networks. The inter-frame differences of the gathered media streams, however, are beyond the scope of most methods. In this research, an Optimized Gaussian Convolutional Neural Network Framework for Enhanced Detection of Deepfakes in Digital Media (OGCNN-DDF-DM) is proposed. Initially the input images are gathered using the Face Forensics++ (FF++), and Deep Fake Detection Challenge dataset (DFDC) datasets. Then the Multi-Window Savitzky-Golay Filter (MWSGF) is used to improve quality of the DF images and reduce noise. Afterwards, Simple Contrastive Graph Clustering (SCGC) achieves segmentation. Here, the image's facial regions are segmented. Then, the texture features are extracted using Revised Tunable Q-Factor Wavelet Transform (RTQWT) is introduced. The extracted features are fed to Gaussian Convolutional Neural Network (GCNN) to categorize the image as real or fake. Finally, Gooseneck Barnacle Optimization Algorithm (GBOA) is proposed to improve the GCNN classifier. Performance parameters including accuracy, precision, recall, specificity, ROC, and computation time are examined. The introduced method attained an accuracy of 99.6% and the precision of 98.9% on the FaceForensics++ dataset, and 99.5% and 98.6% on the DFDC dataset, respectively.

Keywords: DeepFake Detection; Multi-Window Savitzky-Golay Filter; Simple Contrastive Graph Clustering; RTQWT; Hybrid Optimization; Gooseneck Barnacle Optimization Algorithm; Optimized Gaussian Convolutional Neural Network

1. Introduction

Data in computer vision, voice generation and recognition, and the development of multi-agent systems in sectors have all shown that artificial intelligence has made significant advances in a number of areas [1] [29] [30]. The DFs were a result of the transformation of multimedia processing brought about by generative DL approaches [2–3]. When applied to a person's image, DFs are a type of synthetic media where new content mimics the original media. The term "deepfake" was first used in 2017 on the internet forum Reddit, according to reports [4-5]. In the majority of scenarios, deepfakes take advantage of deep learning technologies including generative adversarial networks (GANs), as well as technology for face swapping, audio editing, and video graphics [6]. Identity theft, cyber extortion, impersonation schemes, fake news, hate speech, vote rigging, financial fraud, cyberbullying, and the exploitation of fictitious pornographic movies of celebrities for blackmail and democratic process manipulation are just a few of the cybercrimes that have been committed through DFs [7] [8][9]. Given that deepfakes are increasingly being used to pose a danger to people and organizations, detecting deepfake media is presumably a recently emerging challenge in digital forensics.

An upsurge in crimes connected to deepfakes has been noted [10], and according to a Sensity analysis, obscenity was included in over 96% of deepfakes. South Korea, India, Canada, the United States, and the United Kingdom are among the most adversely affected countries [11–12]. In one instance in 2019 alone, two hackers deceived a CEO into thinking they had an audio recording of the director in order to extort $243,000. They attest to the need for additional equipment development for the identification of deepfakes and the control of their misuse. In order

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

to combat crimes linked to deepfakes, facial recognition technology is crucial. A biometric security system authenticates an individual by comparing their facial data to a database that has been set up [13–14]. Face recognition software used by security and law enforcement organizations is also crucial to the identification of cybercrimes. These systems focus on the chin, ears, lip curves, cheekbone structure, eye socket depth, and inter-eye distance when comparing 2- and 3-D face images [15]. In face recognition technology, DL (deep learning) networks are used to recognize and learn faces, sometimes referred to as facial patterns. These networks are highly skilled in representation learning and convert face information into some kind of numerical input [16–17]. DL-based models may efficiently distinguish core biometric characteristics because they are recognized by many neuronal connections [18–19]. When it comes to face recognition, the models—that were trained on massive face databases—are significantly more effective than humans [20].

Existing DF detection models tend to have limited generalization between datasets and new DF generation methods. Handcrafted features or patterns of certain artifacts are overemphasized by most and, thus, are susceptible to adversarial attacks and changing techniques of manipulation from the adversary. Convolutional Neural Networks (CNNs), while effective, sometimes fail to capture subtle artifacts or statistical inconsistencies in manipulated content. Furthermore, conventional optimization methods including stochastic gradient descent (SGD) and its extensions may suffer from suboptimal performance due to problems related to slow convergence, inadequate hyper parameter optimization, and getting stuck in local minimizes. These challenges limit the robustness and scalability of the current solutions. These limitations of existing models motivate us to do this research work.

This work's novelty is rooted in the incorporation of a GCNN framework and the GBOA for advanced DF detection. In the GCNN, Gaussian-based filtering and probabilistic modeling are used to detect subtle spatiotemporal artifacts that are often missed by traditional CNNs. In the meantime, the GBOA offers a nature-inspired metaheuristic optimization approach, which outperforms conventional strategies regarding hyperactive parameter tuning, convergence rate, and avoiding local minima. This special combination allows a strong and adaptive method of DF detection, overcoming issues in generalization, scalability and resistance to changing manipulation methods.

This paper's major contribution could be stated up as described below,

This manuscript presents a design an optimized framework for DF detection in digital media called “OGCNN-DDF-DM”.

- ✓ In order to improve the input images, the Multi-Window Savitzky-Golay filter (MWSGF) is introduced as filtering for the noise. With this smoothing process, an image data is made smoother.
- ✓ The framework consists of a new segmentation step by means of Simple Contrastive Graph Clustering (SCGC). SCGC is dedicated to segmentation of facial parts of the image, extraction of interesting characteristics of interest-like eyes, mouth, etc and so on.
- ✓ This paper suggests the RTQWT for extract texture features. The optimized GCNN is used as the classifier in the system. GCNN utilizes the learned features to classify images as real or artificial.
- ✓ To further enhance the performance of the GCNN classifier, the Gooseneck Barnacle Optimization Algorithm (GBOA) is proposed. Face Forensics++ and the DFDC datasets are used for checking the validity of the proposed work.

The manuscript's remainder part will be separated into subsequent sections: The review of literature is described in section 2, the innovative methodology for the proposed work is presented in section 3, the outcomes and discussions are provided in section 4, and the conclusion is presented in section 5.

2. Literature review

Several research works presented in the literatures were based on DF detection utilizing DL; few of them were reviewed here,

In 2023, Khalid, et.al [21] have presented an interpretable and generalized graph neural network for DF detection. In order to detect hyperrealist Dg content, a unique architecture based on GNNs is suggested. The presented model consists of two main modules to modify and communicate information between all nodes: FFN, which comprises linear layers for node feature transformation, and GraphNet, which collections and updates graph information via graph convolution layers. The Celeb-DF, FF++, World Leaders dataset (WLRD), and the varied DFDC dataset are used to evaluate the method's efficacy.

In 2024, Soudy, et.al [22] have discribed DF detection utilizing CNN and convolutional vision transformers. Preprocessing, detection, and prediction are the three parts of the system that is being described. Frame extraction,

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

face detection, alignment, and feature cutting are all examples of preprocessing. The eye and nasal feature detection phase makes use of CNNs. For face detection, a CNN in conjunction with a vision transformer is also utilized. Using a majority voting method, the prediction component combines the output of the three models implemented to various features to provide three distinct forecasts. The DFDC and FF++ datasets are used to train the suggested approach on a variety of face images.

In 2023, Nawaz, et.al [23] have suggested ResNet-Swish-Dense54: a deep learning approach for DF detection. For dependable and accurate DF detection, a DL-based method called ResNet-Swish-Dense54 was employed. First, input video frames were used to extract human faces. The extracted faces were then provided to the ResNet-Swish-Dense54 model to classify the content as either altered or real. The robustness of the suggested method was verified through experimentation, evaluating the model on the challenging DFDC, FF++, and CelebDF datasets. It provides high accuracy and low specificity.

In 2024, Sekar, et.al [24] have presented DFD utilizing an optimal DL model with multi head attention-based feature extraction scheme. Here, The Viola–Jones technique is used to first identify the face regions from the gathered data. The identified facial regions are then resized and normalized as part of the preprocessing step. The Butterfly Optimized Gabor Filter is then used to learn texture features. Residual Network-50 with Multi Head Attention is then used to extract the spatial features. The data is then classified as either true or fraudulent utilizing the Optimal LSTM, and the network is optimized using the Enhanced AOA. Four benchmark datasets, including FF ++, DFDC, CDF, and WDF (Wild deepfake), are used to assess the suggested method. It provides high sensitivity and low accuracy.

In 2023, Patel, et.al [25] have presented an enhanced dense CNN architecture for DF image detection. The suggested research introduced new and enhanced deep-CNN (D-CNN) architecture for DF detection, demonstrating decent excellent and accuracy generalizability. Images from various sources were gathered to train the model, thereby enlightening its overall generalization abilities. The rescaled images were fed into the D-CNN model, and to enhance the presented model's learning rate, the binary cross-entropy and Adam optimizer were employed. It offers low computation time.

In 2024, Almestekawy, et.al [26] have introduced DF detection: improving performance with DL feature fusion and spatiotemporal texture. The suggested method integrates deep learning-based characteristics with a variety of spatiotemporal textures. A Siamese design makes use of an improved 3D CNN with a spatiotemporal attention layer. Numerous assessments are conducted on the reproducibility of results, feature relevance, and control factors. Four datasets such as Celeb-DF, FF++, DeepfakeTIMIT, and FaceShiftes are used to evaluate the suggested method. It provides high accuracy and low precision.

In 2024, Huda, et.al [27] have presented the Fake-checker: A fusion of texture features and deep learning for DF detection. Here, Inception V3 was used to extract the deep feature vector of 2048-D, and a Directional Magnitude Local Hexadecimal Pattern was suggested for extracting 320-D texture features. After feature fusion, a balanced representation was obtained by reducing the feature dimensions to 320 using Principal Component Analysis. The XGBoost model for identifying authentic or fake frames was then trained using the suggested attributes. The DFDC and FF++ datasets were used to test the model. It offers low error rate and low sensitivity. Table 1 presents the review of the literature.

Table 1: Literature survey table

Author(s)	**Objective**	**Techniques**	**Advantages**	**Drawbacks**
Khalid, et al. (2023)	Presented an explainable and generalized GNN for DF detection.	Graph neural network and GraphNet	Achieved high accuracy	Provides low sensitivity
Soudy, et al. (2024)	Developed a system using ML for DF detection.	Convolutional vision transformers and CNNs	It provides high precision	Error rate is high
Nawaz, et al. (2023)	Introduced ResNet-Swish-Dense54 for DF detection.	ResNet-Swish-Dense54	It provides high accuracy	Provides low specificity.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

Sekar, et al. (2024)	multi-head attention-based DL	Butterfly Optimized Gabor Filter, ResNet-50 and LSTM	It provides high sensitivity	Accuracy is low
Patel, et al. (2023)	Dense CNN	deep-CNN	Low computation time	It provides high error rate.
Almestekawy, et al. (2024)	Enhanced DF detection by integrating spatiotemporal texture	3D CNN and spatiotemporal attention layer	It provides high accuracy	Low precision.
Huda, et al. (2024)	Developed Fake-checker for detecting DF	Inception V3 and XGBoost	Low error rate.	It attains low sensitivity.

3. Proposed methodology

In this section, OGCNN-EDDI-DM framework is proposed for DF image detection. This work aims at optimized architectures of GCNN for the robust detection of DF images. The block diagram of proposed model is demonstrated in Figure 1.

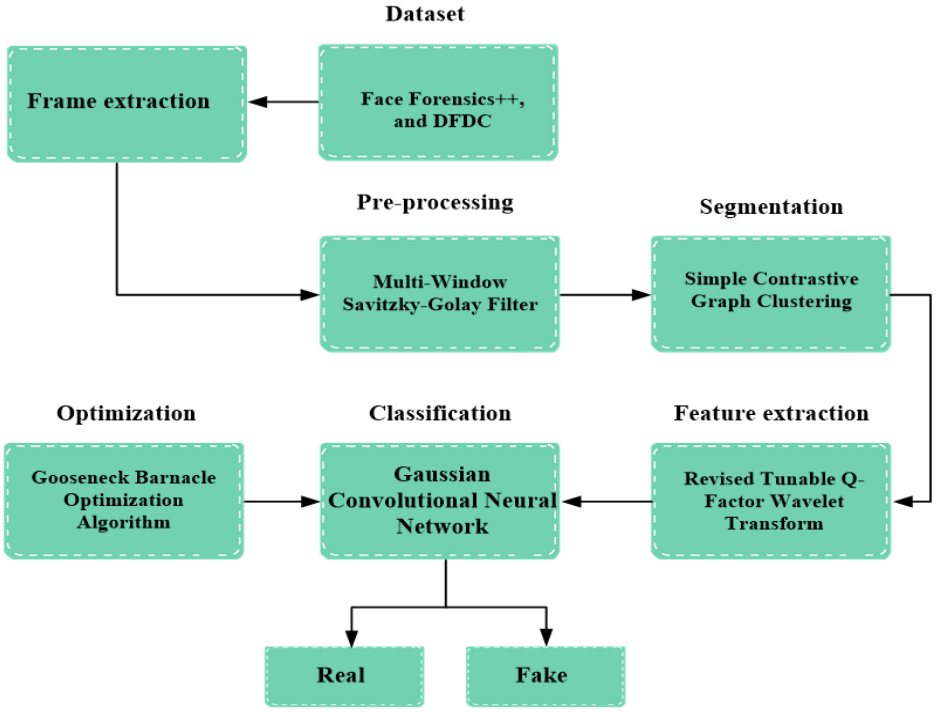

Figure 1. Block diagram of proposed OGCNN-EDDI-DM model

3.1 Video acquisition

In this method trains the models utilizing benchmark datasets include Face Forensics++ and the DFDC dataset (Deep Fake Detection Challenge).

FF++: A widely used and large database of visual alterations is the FF++ dataset (available at: https://www.kaggle.com/datasets/hungle3401/faceforensics). 1000 real and 4000 modified samples of various subjects make up the FF++ dataset. Many forensic techniques, including DeepFakes, FaceSwap, Face-Reenactment, and Neural Textures, are used to modify the samples.

Deep Fake Detection Challenge dataset (DFDC): There are five thousand real and modified videos in the DFDC dataset (can be downloaded from https://www.kaggle.com/c/deepfake-detection-challenge/data). The paid actors are the source of the real footage, while several DF, GAN-based, and non-learned techniques were utilized to generate the fakes. Facial manipulation techniques, such as Face2Face and DF, are used to create fake videos. Different acquisition circumstances (i.e., indoor and outdoor), lightning conditions (i.e., day and night), the

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

distance between the subject and the camera, changes in attitude, etc. are all taken into account by DFDC. DFDC is diverse in a number of ways, including age, gender, and skin tone.

3.2 Frame Extraction

The input videos' frames are obtained utilizing the OpenCV library (cv2) in the proposed method. Frames are extracted at regular intervals, specifically one frame out of every ten, using a systematic process. This interval strikes a balance between reducing computational costs and obtaining sufficient frames for analysis. The extracted frames are then saved as individual images, with real frames stored in one folder and fake frames stored in another. For processing efficiency and compatibility with various deep-learning models, the final images are standardized to a resolution of224×224.

3.3 Pre-processing using Multi-Window Savitzky-Golay Filter with SSIM, VL, and Wiener Filtering

Preprocessing is vital for deepfake detection since it helps to improve image quality as well as eliminate deformities that could negatively influence model performance. By methodically improving input images, the suggested MW-SVW (Multi-Window Savitzky-Golay with SSIM, VL, and Wiener Filtering) model assures reliable preprocessing. First, to identify subtle deepfake artifacts such as textural abnormalities and blending inconsistencies, Multi-Window Savitzky-Golay Filtering (MW-SGF) is used to smooth noise while maintaining key characteristics across different scales. Then, in order to maximize computing efficiency, the Structural Similarity Index Measure (SSIM) compares structural content, brightness, and contrast to find and eliminate duplicate pictures. After that, low-quality photos are identified by Variational Learning (VL), which uses the Laplacian operator to analyze edge sharpness and detect blurred images. Wiener Filtering, that employs statistical noise estimates and deconvolution to regain sharpness, is subsequently employed to repair blurry pictures, which helps uncover deepfake discrepancies. By ensuring that only high-quality, artifact-free images are sent to deepfake detection models, this all-encompassing method increases the precision of classification and lowers false positives. Fig. 2 manifest the architecture of the pre-processing phase.

MWSGF: In this section, pre-processing utilizing MWSGF is presented. Since Savitzky-Golay Filter can remove unwanted noise in input images and meanwhile preserve edges, fine details, the MWSG filter can significantly reduce its contribution from the viewpoint of a DF. With the implementation of multiple window sizes, the filter is able to learn features of different scales, leading to an enhanced ability to pinpoint less apparent drifts such as blending, or textural distinctions across regions. It maintains the important information like edge and texture. The polynomial approximation is given in equation (1)

$$Z(c) = \sum_{a=0}^{m} s_a c^a \quad (1)$$

Where, $Z(c)$ represents the polynomial approximation of the image intensity at a pixel$c$, $s_a$denotes coefficients of the polynomial degree$a$, and $m$is a degree of the polynomial. It creates a smooth approximation of the noisy image and acts as the core function of the MWSG filter, preserving features like edges while reducing noise. Then the fitting error is calculated using equation (2)

$$\delta_m = \sum_{l=-M}^{M} (B(l) - z[l])^2 \quad (2)$$

Where, δ_m is the total squared error between the noisy image $B(l)$ and the smoothed image $z[l]$ over a window and $M$is half the size of the smoothing window. It quantifies how well the polynomial approximation fits the noisy image. The polynomial coefficients are optimized using equation (3)

$$\boldsymbol{\delta_m = \sum_{l=-M}^{M} (\sum_{a=0}^{m} k_a l^a - z[l])^2} \quad (3)$$

Where, k_a indicates the coefficients of the polynomial fit within the local window and l^a represents polynomial basis function. Then the filtered output is given in equation (4)

$$y(a) = \sum_{l=-M}^{M} \omega_l z[a-l] \quad (4)$$

Equation (4) computes the final de-noised image $y(a)$ by combining the smoothed pixel intensities $z[a-l]$ using weightsω_l. Hereby MWSGF removes noise from the input image. The pre-processed image is nourished to segmentation stage.

Following this, SSIM metrics identify and remove duplicate images by comparing structural information, luminance, and contrast patterns between image pairs. The pipeline then employs VL to detect blurred images by analyzing edge sharpness and local intensity variations. For images identified as blurred, the Wiener filter performs deconvolution using statistical estimations of noise and blur kernels to restore image clarity. This comprehensive approach ensures optimal image quality for subsequent Deepfake detection tasks while maintaining computational efficiency.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

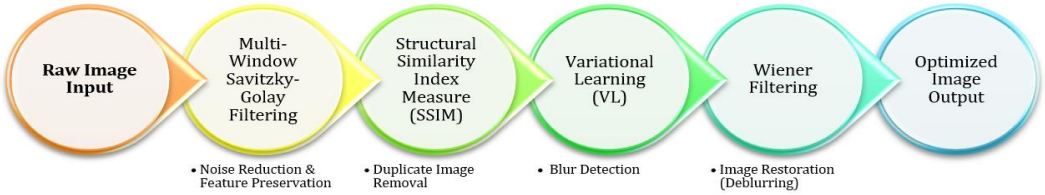

Figure 2. Pipeline of Pre-processing Stage

SSIM for Duplicates Removal: It is a metric used to measure the similarity between two images. It is often applied to identify duplicate or highly similar images by quantifying structural content similarity as defined in Eq. (3), where $\boldsymbol{m}$ and $\boldsymbol{n}$ indicate two images being compared, $\boldsymbol{\mu_m}$, $\boldsymbol{\mu_y}$, and $\boldsymbol{\sigma_m^2}$, $\boldsymbol{\sigma_n^2}$ state mean intensity and variance of images $\boldsymbol{m}$ and $\boldsymbol{n}$, respectively, $\boldsymbol{\sigma_{mn}}$ refers to covariance between images $\boldsymbol{m}$ and $\boldsymbol{n}$, and $\boldsymbol{C_1}, \boldsymbol{C_2}$ addresses small constants to stabilize the division when the denominator is close to zero.

$$SSIM(m,n) = \frac{(2\mu_m\mu_y+C_1)(2\sigma_{mn}+C_2)}{(\mu_m^2+\mu_n^2+C_1)(\sigma_m^2+\sigma_n^2+C_2)} \quad (5)$$

VL for Blur Detection: It is a method used for blur detection in images. It measures the sharpness of an image by calculating the variance of the Laplacian operator, which highlights edges. Besides, the blurry image is indicated by a low variance i.e., lack of edges as expressed in Eq. (6), in which $\boldsymbol{L_i}$ refers to Laplacian value at pixel $\boldsymbol{i}$, $\boldsymbol{n}$ addresses total number of pixels in the image, and $\boldsymbol{\mu}$ indicates mean of the Laplacian values across the image which is estimated as per Eq. (7).

$$VL = \frac{1}{n}\sum_{i=1}^{n}(L_i - \mu)^2 \quad (6)$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} L_i \quad (7)$$

Steps involved in VL are:

- To apply the Laplacian Operator by computing the Laplacian of the image to detect regions with significant intensity changes (edges).
- To calculate variance by measuring the variance of the Laplacian values.
- To set a threshold by comparing VL to a predefined threshold. If VL is below the threshold, the image is considered blurry.

De-Blurring via Wiener Filter: It is a technique for image restoration, specifically designed to reduce blur and noise (Qin & Zhang, 2024). It aims to reconstruct an image by minimizing the mean squared error (MSE) between the estimated and the original images. The observed blurred image $\boldsymbol{G(x,y)}$ initially go through a Fourier Transform to both sides as illustrated in Eq. (8), where $\boldsymbol{G(u,v)}, \boldsymbol{I(u,v)}, \boldsymbol{H(u,v)}$, and $\mathrm{N}(\boldsymbol{u},\boldsymbol{v})$ indicate Fourier Transform applied de-blurred image, observed blurred image, degradation function, and additive noise in order.

$$G(u,v) = I(u,v)H(u,v) + N(u,v) \quad (8)$$

Now, use the Wiener filter to estimate $\hat{I}(u,v)$, the original image in the frequency domain as explained in Eq. (9), where $H^*(u,v)$ addresses complex conjugate of $H(u,v)$, $S_n(u,v)$ points to power spectral density of noise, $S_f(u,v)$ indicates power spectral density of the original image, $|H(u,v)|^2$ signifies magnitude squared of the degradation function, and $\frac{S_n(u,v)}{S_f(u,v)}$ stands for Noise-to-signal ratio (NSR).

$$\hat{I}(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2+\frac{S_n(u,v)}{S_f(u,v)}} G(u,v) \quad (9)$$

At this point, convert $\hat{I}(u,v)$ back to the spatial domain using the Inverse Fourier Transform as defined in Eq. (10).

$$\hat{I}(x,y) = F^{-(\hat{I}(u,v))} \quad (10)$$

3.4 Segmentation using Simple Contrastive Graph Clustering

In this section, Segmentation utilizing Simple Contrastive Graph Clustering (SCGC) is deliberated. Here, SCGC is used to segment facial regions. SCGC is beneficial for facial region segmentation, because it is based on the

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

graph-wise representations to well model the relationship between facial landmarks. SCGC, with the help of the embedding vectors of pixels as graph nodes and the contrastive learning, can capture both local features and global spatial structures, thus leading to a better performance in segmentation. Its robustness to perturbation, occlusion, and a wide range, of facial variability is further ameliorated by use of contrastive loss. The similarity measure K_{rp} between feature components r and $p$is given in equation (11)

$$K_{rp} = B_r^{v1} \cdot \left(B_p^{v2}\right)^Q, \rightleftarrows \rightleftarrows \rightleftarrows \rightleftarrows \; \forall r, p \in [1, M] \tag{11}$$

Where, B_r^{v1} and B_p^{v2} are feature representation from one view and second view and Q indicates normalization vector. It used to calculate similarity between input deepfake image pixels in different feature spaces. Similarity measures are critical for graph construction in which nodes are image regions and edges are weighted similarity. The contrastive loss function is given in equation (12)

$$C = \frac{1}{M^2} \sum_r \left(K - \widehat{H}\right)^2 \tag{12}$$

Where, Mindicates the number of regions, $K$is the similarity matrix, $\widehat{H}$is a target similarity which represents the desired relationship between features and $C$is the total cost over all embeddings. Then aggregate features from both views for final segmentation using Equation (13)

$$B = \frac{1}{2}(B^{v1} + B^{v2}) \tag{13}$$

Where, $B^{v1}$and B^{v2}features from one embedding space and another embedding space. The output highlighting face areas like eyes, lips, and skin. Then the output is fed to feature extraction stage.

3.5 Feature extraction-using RTQWT

In this section, feature extraction using RTQWT is discussed. Using RTQWT, the texture features are extracted. As a texture analysis tool, the RTQWT is also powerful enough that it can be applied to the decomposition of images into many frequency bands in a tunable manner, in accordance with an image's application. The normalized energy feature J_r^s is given in equation (14)

$$J_r^s = \frac{|z^s(r)|}{\sum_{p=1}^{G} |z^s(p)|} \tag{14}$$

Where, $z^s(r)$ indicates the wavelet coefficient of the DF image, and $\sum_{p=1}^{G} |z^s(p)|$is a total sum of the absolute values of the coefficients. This normalization actually leads to a relative energy distribution of the wavelet coefficients, such that it is possible to extract features that do not depend on the magnitude of the image. The Shannon entropy is given in equation (15)

$$O^s = -\sum_{r=1}^{G} J_r^s \, log \, J_r^s \tag{15}$$

Where, J_r^s indicates the normalized energy at position$r$and $log \, J_r^s$ is the logarithmic term to compute the entropy. Shannon entropy extracts the texture features within the wavelet coefficient distribution. The following texture features are extracted by RTQWT.

Entropy measures the randomness in pixel intensity distribution, which is given in equation (16)

$$Entropy = -\sum_{a,b}^{M-1} s(a,b) \, log \, s\,(a,b) \tag{16}$$

Where, $s(a,b)$indicates Co-occurrence matrix value for intensity levels $a$and b and Mdenotes total intensity levels.

Contrast is the change of fluorescent intensity between pixels in an image or in defined area. It measures the degree of change in the pixel values itself, which determines edge sharpness or intensity primitives. It is calculated by equation (17)

$$Contrast = \sum_{a,b}^{M-1} (a,b)^2 s(a,b) \tag{17}$$

Energy calculates the texture's smoothness or sum of squared components. Energy is calculated by equation (18)

$$Energy = \sum_{a,b}^{M-1} s(a,b)^2 \tag{18}$$

Homogeneity calculates the uniformity of pixel intensity distribution. Higher homogeneity indicates smoother textures. It is given in equation (19)

$$Homogeneity = \sum_{a,b} \frac{s(b,a)}{1+|b-a|} \tag{19}$$

Then the extracted features are fed to the detection stage.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

3.6 Deepfake detection using Gaussian Convolutional Neural Network with CasDetNet

In this section, DF detection using Gaussian Convolutional Neural Network (GCNN) is discussed. GCNN improves the network's ability to capture subtle inconsistencies like unnatural textures, blending artifacts, and lighting distortions that arise while generating DF. The GCNN enhances accuracy in detection, even with regard to high quality or photorealistic DF. GCNN applies consecutive layers that operate and process an image using a combination of operations involved in the use of convolution and pooling followed by activation functions, in order to identify slight anomalies within DF. The architecture diagram of GCNN is shown in figure 3.

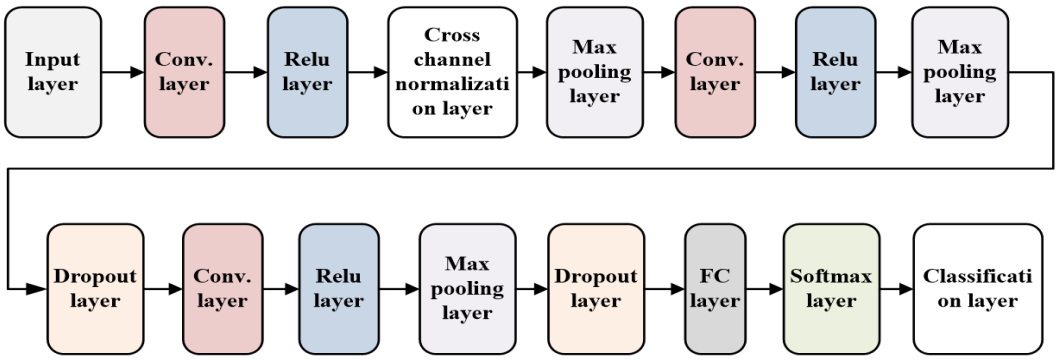

Figure 3. Architecture diagram of GCNN

There are sixteen layers in all, starting with the input layer, convolution, down sampling using Normalization, convolution, Rectified-Linear Unit (ReLU), and Pooling. Over fitting is prevented by employing the dropout layer. Softmax and fully connected layers are later used for output detection, and a classification layer is added for DF-class classification. Four convolutional layers are used in the proposed method to detect DF images. The input layer makes sure that the Bluetooth images are an appropriate size and normalizes the data. Sliding filters across the input images and calculating the dot product between the input and filter weights applies a 2D convolution. Both horizontal and vertical movement of the filters across the input images is referred to as "stride." To preserve the image's dimensions, padding is applied to it prior to the kernel sliding. ReLU, or non-saturated activation functions, is employed here to reduce the training time compared to other activation functions. It follows each convolutional layer. The ReLU activation function $g(b)$is given in equation (20)

$$g(b) = max(0, b) \tag{20}$$

Where, $b$indicates an extracted feature in convolution layer. Here, ReLU is used to process the features from image input at the GCNN convolutional layer. It helps extract high-quality features useful for distinctive between real and fake. The weighted sum of the output features from different layers $c(x)_l$ is given in equation (21)

$$c(x)_l = \frac{-(e_l^x)}{(\sum_{i=0}^{k} k{=}1)e_l^x} \tag{21}$$

Where, $k$denotes the detection output and e_l^xrepresents an exponential operation, which can be used to weight the features. It helps the GCNN focus on important cues that differentiate real and fake images. The cross-entropy loss function $U(p, o)$ is given in equation (22)

$$U(p, o) = -\sum_b \big(p(b) * log\big(p(b)\big)\big) \tag{22}$$

Where, $p(b)$indicates predicted probabilities. Cross-entropy loss is a quantity of the proposed model's performance in penalizing more heavily wrong classifications if the network is confident of its wrong predictions. The decay of weight and penalty adding to the cost-operation $c$utilizing L2 regularization is represented in the equation (23).

$$C = cos\, t\, (loss) + \beta \sum_{i=0}^{k} (b = 1) A_b^2 \tag{23}$$

Where, A_b^2represents the square of the activation values and $\beta$is regularization attribute. Here, a regularization term is added to the loss function, which encourages the proposed model to generalize better. Finally, the input images are classifying as real or fake using GCNN. The GCNN classifier analyzes the artificial intelligence-based optimization approach for its simple implementation and applicability. The GCNN optimal parameterβ has been optimized in this study utilizing GBOA. Here, GBOA is employed for modifying the GCNN's weight and bias parameters.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

3.6 Optimization using Gooseneck Barnacle Optimization Algorithm

In this section, optimization of GCNN weight parameter βusing Gooseneck Barnacle Optimization Algorithm (GBOA) is discussed. GBOA is a highly efficient optimization algorithm. Stimulated by the unique clustering and tracking behavior of gooseneck barnacles, GBOA achieves an efficient balance between exploration and exploitation during optimization. GBOA evades local minima and converges rapidly to global optimal [28]. It is very easy to use and computationally cheap due to its simple mathematical model and small calibration parameter. The stepwise procedure is given below,

Step 1: Initialization

Gooseneck barnacles are assumed the candidate solution in the proposed GBOA, and the gooseneck's position is the variable for the problem space. The gooseneck's penis could be extended seven or eight times to reach their body, allowing them to move through any dimensional space. The total number of stalked barnacles referred to as gooseneck barnacles, is the candidate of solution A for initialization as GBOA is a population-based method. This population can be expressed in equation (24)

$$A = \begin{bmatrix} (a+k)_{1,1} & (a+k)_{1,2} & \cdots & (a+k)_{1,c} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ (a+k)_{m,1} & (a+k)_{m,2} & \cdots & (a+k)_{m,c} \end{bmatrix} \quad (24)$$

Where, m and c stand for the number of dimensions or variables that need to be enhanced and the total populations, respectively. Each gooseneck has a different length$k$, which will be chosen at random because of its edible structure.

Step 2: Random generation

The input parameters are chosen at random. The selection of ideal fitness values relied on the obvious hyperparameter condition.

Step 3: Fitness function estimation

The fitness function is appraised with optimization parameter value for optimizing weight parameter $\beta$of GCNN. It is shown in equation (25),

$$Fitnessfunction = optimzing[\beta] \quad (25)$$

Step 4: Sperm casting

Models the sperm casting behaviour of barnacles, a process where individuals release sperm into the water to fertilize the eggs. In GBOA, this step consists of creating new solutions derived from existing solutions, which is used to promote the exploration of the search space. A nearby logarithmic spiral zone is defined for sperm casting is given in equation (26)

$$R((A+k)_u, (R_{water})_v) = C_u . e^s . cos(2\pi s) + (R_{water})_v \quad (26)$$

Where s is a random number in [-1, 1], k is a constant that defines the shape of the logarithmic spiral, C_u denotes the distance between the u^{th} barnacle and the v^{th} sperm area for mating and R_{water}indicates the v^{th} sperm casting region.

Step 5: Food availability assessment

It is anticipated that sperm-cast mating behavior combines a few of the patterns covered in this research. In order to mimic the movement of new offspring generation and update the position of new gooseneck barnacles in a search space, a movement vector is taken into consideration, Δ, which is specified in equation (27) and (28)

$$\delta(A+1)_{u+1} = W_u + S(A+1)_{u_{dim}} \quad (27)$$

$$(A+1)_{u+1} = (A+1)_u + \delta(A+1)_{u+1} \quad (28)$$

Using the radius of the search area and the degree range [0 359], W_u defines the wind direction, which is then added to the best solution together with the target dimensionS_{dim}. Assuming that the wind is constantly directed toward the target is the value of the dimension in the target.

Step 6: Termination

Using the assistance of GBOA, the weight parameter value$\beta$from the GCNN are optimized using GBOA, continue with step 3 frequently until fulfil halting criteria$A = A + 1$. Then, the GCNN method effectively detects DF with

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

high accuracy and low error rate. Pseudo code of Gooseneck Barnacle Optimization Algorithm is assumed in Algorithm 1.

Algorithm 1: Pseudo code of Gooseneck Barnacle Optimization Algorithm

Input: Population size, Search space boundaries, Termination criteria.

Output: GlobalBestMatingPosition

Initialize population with random positions.
Assign random lengths to each barnacle.

While termination criteria not met:
  Adjust A if out of search space.
  Evaluate fitness function.
  Update GlobalBest.
  Sort mating positions based on GlobalBest.
  For each variable in A:
    If multiple mating regions available:
      Compute movement vectors
      Update offspring positions
    Else if single mating region available:
      Update offspring positions
    End if.
  End for.
  Adjust all positions to ensure they remain within search space.
  Update GlobalBestMatingPosition

Return GlobalBestMatingPosition

4. Results and Discussion

A detailed analysis of the outcome of the proposed model with existing approaches was conducted using several performance metrics. To demonstrate the efficiency of the proposed model, a comparative analysis and performance analysis are conducted. For the proposed model, key hyperactive parameters include input image resizing of 224 x 224 pixels, and the dropout rate of 0.5. The model uses the binary cross-entropy loss and an L2 regularization penalty. Training consists of a batch size of 32, maximum 50 epochs, learning rate 1×10^{-4}, and early stopping. Figure 4 shows the output images of proposed OGCNN-DDF-DM model.

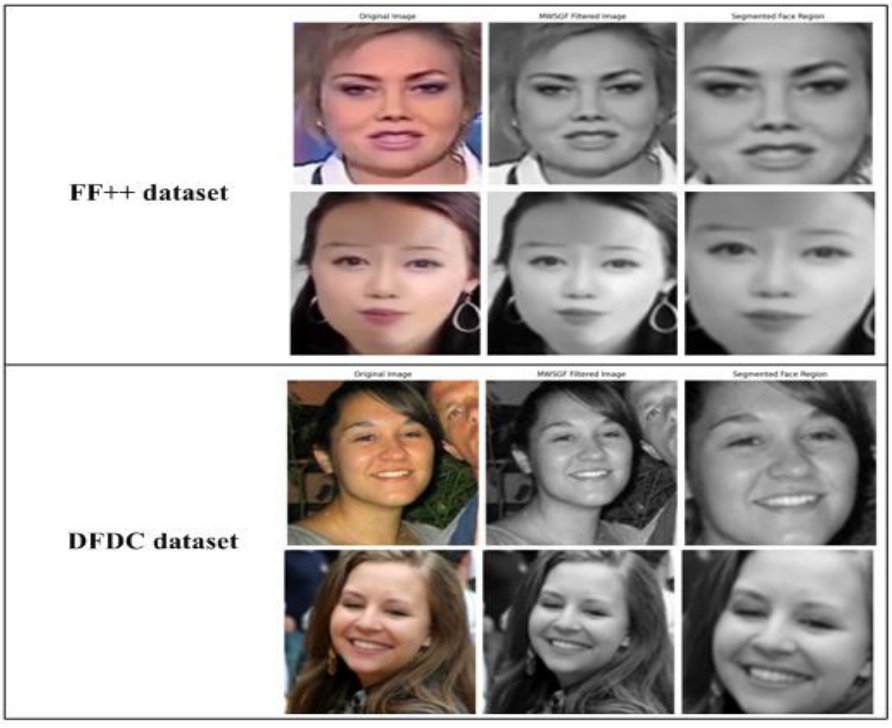

Figure 4. Output images of proposed OGCNN-DDF-DM model

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

4.1 Experimental Setup

The evaluation of each investigation experiment was conducted using the Python programming language. The neural network techniques used were built using the TensorFlow module (version 2.8.2) and the Keras module (version 2.8.0). The proposed framework works on the Linux operating system and is implemented on the NVIDIA DGX-1 working platform, which has eight V100 GPU accelerators and 32 GB of RAM each.

4.2 Performance measures

The performance of the proposed model is assessed utilizing performance metrics including accuracy, precision, F1-score, recall, specificity, and computation time.

4.1.1 Accuracy

The capacity to calculate an exact value is referred to as accuracy. The performance of a technique in each class could be explained with an indicator called accuracy. It is calculated by the equation (29)

$$\boldsymbol{Accuracy} = \frac{(\boldsymbol{TP}+\boldsymbol{TN})}{(\boldsymbol{TP}+\boldsymbol{FP}+\boldsymbol{TN}+\boldsymbol{FN})} \quad (29)$$

Where, TPindicates True positive, $TN$is a True negative, FPrepresents False positive andFN is a False negative.

4.2.2 Precision: It describes the model's capacity to accurately categorize positive predictions out of all positive predictions. This measure shows the proportion of images that were indeed DF out of all the images that the proposed model projected to be DF. It is computed utilizing equation (30)

$$\boldsymbol{Pr\,e\,cision} = \frac{\boldsymbol{TP}}{(\boldsymbol{TP}+\boldsymbol{FP})} \quad (30)$$

4.2.3 Recall: The model's recall is its volume to accurately classify positive positives. Thus, a statistic shows the proportion of images that were identified as DF among all of the actual DF images that were fed into the model. Recall is computed utilizing equation (31)

$$Recall = \frac{TP}{(TP+FN)} \quad (31)$$

4.2.4 Specificity: The percentage of accurately identified real negatives among all actual negatives is known as specificity. In other words, it measures how well the system avoids labeling very content as DF. It is computed employing equation (32)

$$Specificity = \frac{TN}{TN+FP} \quad (32)$$

4.2.5 F1 Score: In essence, the F1 score is their harmonic mean as it is a scale that combines recall and accuracy. Both false positive and false negative findings could be caused by it. Equation (33) is utilized to regulate the F1 score.

$$F1-score = 2\left(\frac{Precision \times Recall}{Precision+Recall}\right) \quad (33)$$

4.3 Performance analysis using Face Forensics++ dataset

The simulation outcomes of OGCNN-DDF-DM technique using FF++ dataset are shown in Figure 5 to 11. The proposed OGCNN-DDF-DM techniques are compared with existing IGGNN-DFD [21], DFD-CVT-CNN [22] and ResNet-Dense54-DFD [23] techniques.

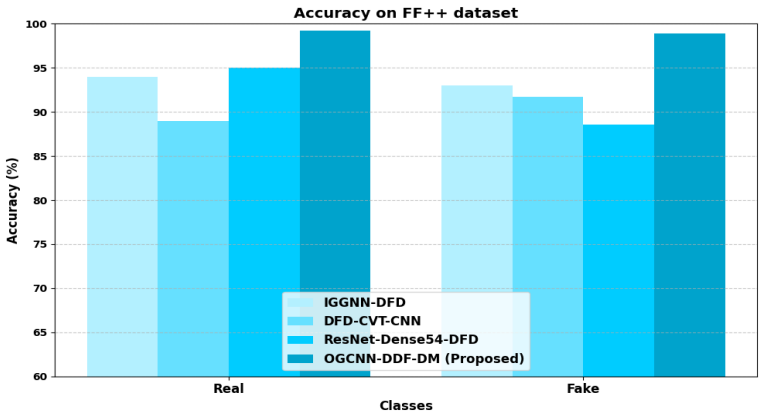

Figure 5. Accuracy analysis on FF++ dataset

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

Figure 4 presents accuracy analysis on FF++ dataset. The OGCNN-DDF-DM technique reaches in the range of 12.56%, 12.23% and 8.15% higher accuracy for real; 13.25%, 15.47% and 7.52% higher accuracy for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively. The high performance of the proposed model is due to its improved ability of the Gaussian convolution layers, which are able to capture delicate discriminative anomalies of DF while eliminating noise, so that the proposed model can perform the manipulation detection more accurately.

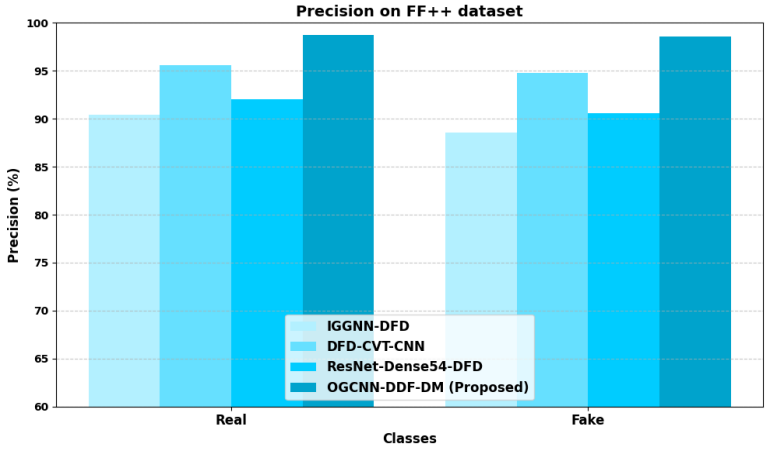

Figure 6. Precision analysis on FF++ dataset

Figure 5 presents precision analysis on FF++ dataset. The OGCNN-DDF-DM technique reaches in the range of 10.23%, 8.59% and 9.78% higher precision for real; 14.94%, 6.38% and 9.96% higher precision for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively. The cause of high precision is that GCNN and the Gooseneck Barnacle Optimization Algorithm are in harmony to prevent over fitting and make the model generalizable to the process of recognizing DF examples unseen in the past.

Figure 6 presents recall analysis on FF++ dataset. The OGCNN-DDF-DM technique reaches in the range of 5.34%, 8.92% and 12.68% higher recall for real; 17.34%, 11.43% and 16.69% higher recall for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively. The recall of proposed model is high because the GCNN is well suited for learning fine and coarse representations from fake face, and therefore can be used for detecting a variety of DF modalities.

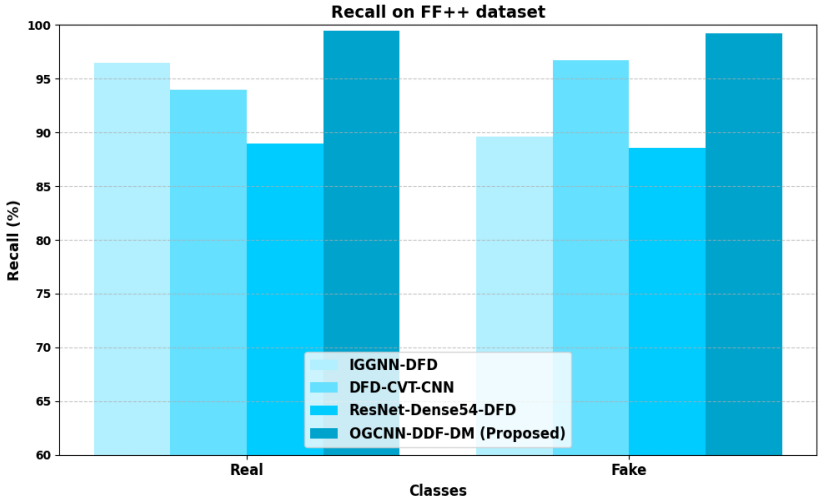

Figure 7. Recall analysis on FF++ dataset

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

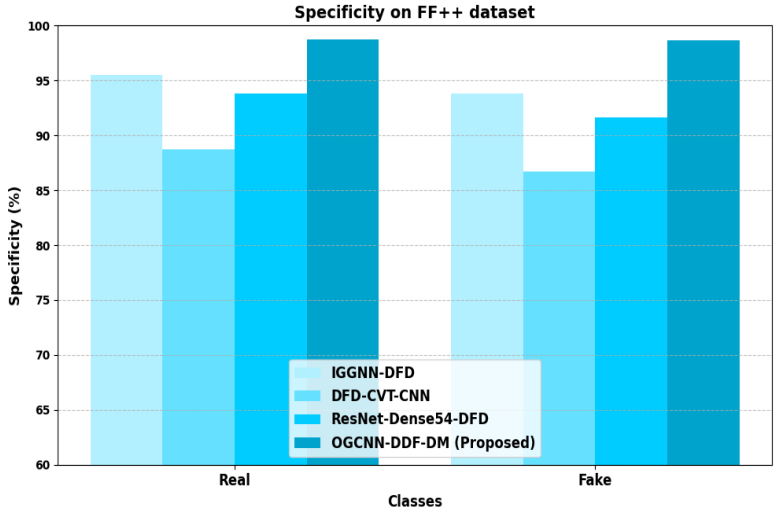

Figure 8. Specificity analysis on FF++ dataset

Figure 7 presents specificity analysis on FF++ dataset. The OGCNN-DDF-DM method reaches in the range of 7.72%, 15.38% and 8.63% higher specificity for real; 8.87%, 18.40% and 9.93% higher specificity for fake when compared with existing methods likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively. Through a reduction of false positives, the proposed model is able to reduce as much as possible false classification of real media as DF.

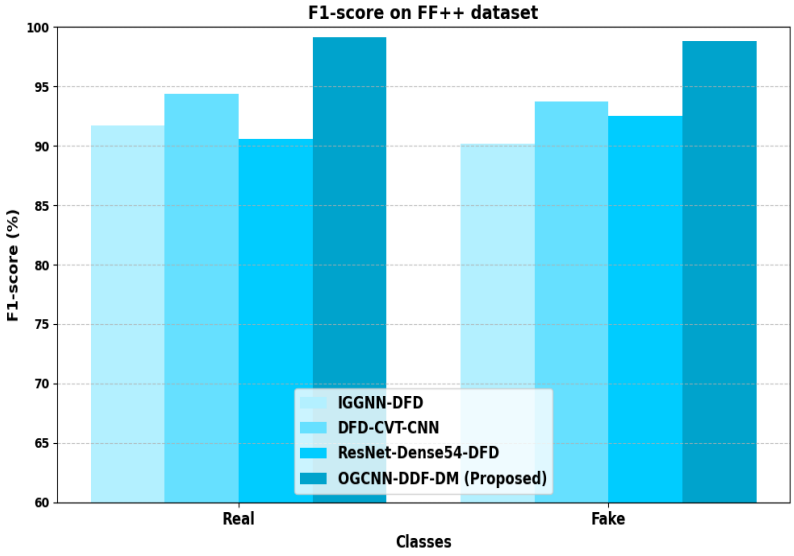

Figure 9. F1-score analysis on FF++ dataset

Figure 8 presents F1-score analysis on FF++ dataset. The OGCNN-DDF-DM technique reaches in the range of 9.97%, 6.39% and 10.84% higher F1-score for real; 10.54%, 8.83% and 9.28% higher F1-score for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

Figure 9 presents computation time analysis on FF++ dataset. The OGCNN-DDF-DM technique reaches in the range of 22.78%, 14.93% and 18.29% lower computation time when likened with existing methods likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

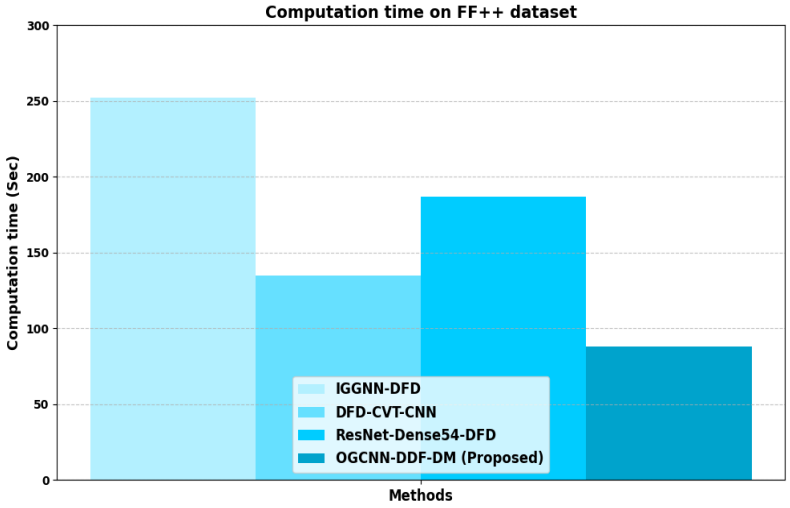

Figure 10. Computation time analysis on FF++ dataset

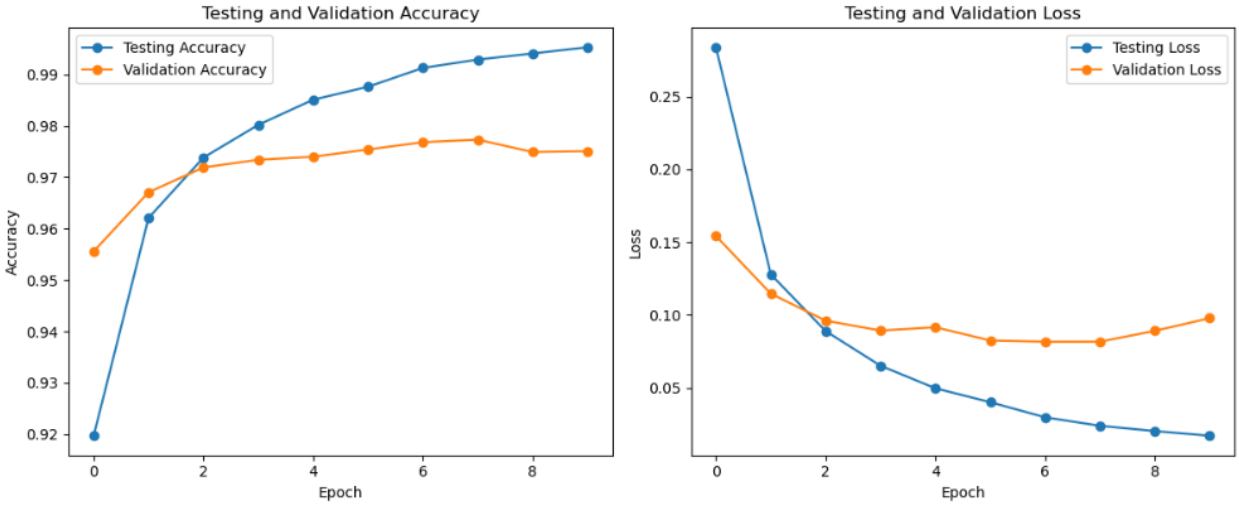

Figure 11. Accuracy vs. Loss

Figure 10 illustrates the accuracy versus loss of proposed model. The plots demonstrate the validation and training accuracy and loss across 10 epochs, with training accuracy progressively increasing to 99.9% and validation accuracy stabilizing at around 97.5%, whilst training and validation loss both decrease monotonically, suggesting robust model performance and resistance to over fitting.

4.4 Performance analysis using DFDC dataset

The simulation results of OGCNN-DDF-DM technique using DFDC dataset are shown in Table 2 to 7. The proposed OGCNN-DDF-DM techniques are compared with existing IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD techniques.

Table 2: Accuracy analysis on DFDC dataset

Methods	Accuracy (%)	
	Real	Fake
IGGNN-DFD	89.36	91.63
DFD-CVT-CNN	91.97	94.34
ResNet-Dense54-DFD	94.37	92.06
OGCNN-DDF-DM (Proposed)	99.5	99.48

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

Table 2 presents accuracy analysis on DFDC dataset. In proposed model, the GCNN applies Gaussian functions to optimize the weights and filters in the Convolutional layers, ensuring smoother transitions in feature extraction. This can improve the detection of subtle spatial features present in DFs. The OGCNN-DDF-DM technique reaches in the range of 11.58%, 18.39% and 8.26% higher accuracy for real; 19.25%, 15.47% and 7.52% higher accuracy for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

Table 3: Precision analysis on DFDC dataset

Methods	Precision (%)	
	Real	Fake
IGGNN-DFD	91.84	92.07
DFD-CVT-CNN	88.06	89.62
ResNet-Dense54-DFD	92.52	90.52
OGCNN-DDF-DM (Proposed)	98.6	98.4

Table 3 presents precision analysis on DFDC dataset. Through parameter optimization, the proposed model is optimized so that it can select the most relevant features, which minimizes errors in the classification and enhances the proposed model's precision. The OGCNN-DDF-DM technique reaches in the range of 9.87%, 19.58% and 13.36% higher precision for real; 20.58%, 16.26% and 10.52% higher precision for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

Table 4: Recall analysis on DFDC dataset

Methods	Recall (%)	
	Real	Fake
IGGNN-DFD	87.32	89.16
DFD-CVT-CNN	96.24	94.53
ResNet-Dense54-DFD	92.58	90.69
OGCNN-DDF-DM (Proposed)	98.26	98.34

Table 4 presents recall analysis on DFDC dataset. GCNNs employ Gaussian functions to improve feature extraction. This method is well suited for finding the fainter artifacts or distortions inside DFs, like texture, lighting, or edge incoherence. The OGCNN-DDF-DM technique reaches in the range of 20.25%, 9.18% and 9.32% higher recall for real; 8.14%, 6.42% and 12.36% higher recall for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

Table 5: Specificity analysis on DFDC dataset

Methods	Specificity (%)	
	Real	Fake
IGGNN-DFD	90.63	89.78
DFD-CVT-CNN	88.95	90.58
ResNet-Dense54-DFD	94.27	93.06
OGCNN-DDF-DM (Proposed)	98.51	99.02

Table 5 presents specificity analysis on DFDC dataset. GBOA is built to converge rapidly to global optima, which guarantees that the GCNN is finely tuned with regard to specificity. This reduces the bias over time for the detection of DF. The OGCNN-DDF-DM method reaches in the range of 10.24%, 12.42% and 7.51% higher specificity for real; 7.26%, 10.41% and 7.26% higher specificity for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

Table 6: F1-score analysis on DFDC dataset

Methods	F1-score (%)	
	Real	Fake
IGGNN-DFD	79.36	81.64
DFD-CVT-CNN	94.27	92.65
ResNet-Dense54-DFD	88.54	90.52
OGCNN-DDF-DM (Proposed)	98.85	98.54

Table 6 presents F1-score analysis on DFDC dataset. Having a high F1-score means the proposed model is successfully achieving a compromise between these two indicators, reducing false positives and false negatives in DF detection. The OGCNN-DDF-DM method reaches in the range of 21.30%, 5.42% and 10.74% higher F1-score for real; 17.5%, 8.52% and 8.57% higher F1-score for fake when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

Table 7: Computation time analysis on DFDC dataset

Methods	**Computation time (Sec)**
IGGNN-DFD	242
DFD-CVT-CNN	152
ResNet-Dense54-DFD	185
OGCNN-DDF-DM (Proposed)	108

Table 7 presents Computation time analysis on DFDC dataset. The ability of Gaussian functions to be utilized in GCNNs enables the network to attempt to concentrate on local features and filter noisy data to help the network achieve better representation of DFs at lower computational time than the existing models. The OGCNN-DDF-DM technique reaches in the range of 17.26%, 28.94% and 15.21% lower computation time when compared with existing techniques likes IGGNN-DFD, DFD-CVT-CNN and ResNet-Dense54-DFD respectively.

4.5 Ablation study

Table 8: Ablation study table

Experiment	**MWSGF**	**SCGC**	**RTQWT**	**GCNN**	**GBOA**	**Accuracy (%)**
Baseline	No	No	No	Yes	No	85.4
+ MWSGF	Yes	No	No	Yes	No	88.7
+ MWSGF + SCGC	Yes	Yes	No	Yes	No	92.2
+ MWSGF + SCGC + RTQWT	Yes	Yes	Yes	Yes	No	96.7
Proposed model (with all methods)	Yes	Yes	Yes	Yes	Yes	99.6

The ablation study table 8 shows the cumulative effect of incorporating each method to the model accuracy. Starting with the Baseline (GCNN only) accuracy of 85.4%, the MWSGF filter accuracy is increased by 88.7% through the elimination noise. Adding SCGC further improves accuracy to 92.2% by improving segmentation. By including RTQWT in the texture feature production, accuracy is improved by 96.7%. Specifically, the Proposed model, consisting of all methods (MWSGF, SCGC, RTQWT, GCNN and GBOA), reaches the best accuracy of 99.6%, demonstrating the rationality and good performance of all components included in it for boosting overall performance.

4.6 K-Fold Cross Validation Results

The performance validation of the proposed method's efficacy is demonstrated in Table 9. Five folds of the data set were used in the k-fold cross-validation process. The intended model, according to the results of the systematic analysis, obtained a 99% average with a standard deviation of 0.5. The findings of the cross-validation additionally determine that the model implemented is an effective one for DF detection on social media.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

Table 9: Performance validation of proposed method

K-Folds	Accuracy (%)	Loss
Fold 1	98.75	0.04
Fold 2	98.56	0.07
Fold 3	99.01	0.08
Fold 4	99.18	0.04
Fold 5	99.46	0.05
Average	99.3	0.5
Standard deviation	±0.5	

4.7 Discussion

On the basis of above analysis, it is concluded that the proposed OGCNN-DDF-DM performs significantly better than approaches used so far. The majority of the conventional approaches function effectively for a limited dataset, but they suffer decline in performance in a large dataset. Therefore, by tackling this issue, the proposed approach makes very good performance even for the big dataset. In addition, the majority of the current works takes much time to train the data. However, the proposed approach trains the data in a minimal time and resources. Thus, it is determined that the existing approach is superior to further cutting-edge techniques.

5. Conclusion

In this manuscript**,** OGCNN-DDF-DM was successfully implemented. This study determines the efficiency of deep learning models in identifying DF videos, using OGCNN and GBOA techniques. The performance of the proposed architecture was evaluated on the DFDC, and FF++ datasets in comparison to other current models. This integrative method is very effective with accuracy and precision above 99% for both the datasets. These results show how this method can be used to detect DFs reliably and uniformly, and therefore it could be a solution to the problems found in digital media authentication. Further study primarily aims to investigate transfer-learning models with hybrid optimization algorithm for the video-based DF detection.

References

[1] S. Sadiq, T. Aljrees, and S. Ullah, "Deepfake detection on social media: Leveraging deep learning and FastText embeddings for identifying machine-generated tweets," *IEEE Access*, vol. 11, 2023.

[2] A. Heidari, N. J. Navimipour, H. Dag, S. Talebi, and M. Unal, "A novel blockchain-based deepfake detection method using federated and deep learning models," *Cognitive Computation*, pp. 1–19, 2024.

[3] S. Albahli and M. Nawaz, "MedNet: Medical deepfakes detection using an improved deep learning approach," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 48357–48375, 2024.

[4] S. Suratkar and F. Kazi, "Deep fake video detection using transfer learning approach," *Arabian Journal for Science and Engineering*, vol. 48, no. 8, pp. 9727–9737, 2023.

[5] R. U. Maheshwari et al., "Advanced plasmonic resonance-enhanced biosensor for comprehensive real-time detection and analysis of deepfake content," *Plasmonics*, pp. 1–18, 2024.

[6] Y. Salini and J. HariKiran, "Deepfake videos detection using crowd computing," *International Journal of Information Technology*, vol. 16, no. 7, pp. 4547–4564, 2024.

[7] L. Cunha, L. Zhang, B. Sowan, C. P. Lim, and Y. Kong, "Video deepfake detection using particle swarm optimization improved deep neural networks," *Neural Computing and Applications*, vol. 36, no. 15, pp. 8417–8453, 2024.

[8] M. Karaköse, H. Yetış, and M. Çeçen, "A new approach for effective medical deepfake detection in medical images," *IEEE Access*, vol. 11, 2024.

[9] J. Gao et al., "Texture and artifact decomposition for improving generalization in deep-learning-based deepfake detection," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108450, 2024.

[10] A. Hashmi, S. A. Shahzad, C. W. Lin, Y. Tsao, and H. M. Wang, "AVTENet: Audio-visual transformer-based ensemble network exploiting multiple experts for video deepfake detection," *IEEE Transactions on Multimedia*, vol. 25, pp. 1234–1245, 2023.

[11] A. A. Khan et al., "Digital forensics for the socio-cyber world (DF-SCW): A novel framework for deepfake multimedia investigation on social media platforms," *Egyptian Informatics Journal*, vol. 27, p. 100502, 2024.

[12] S. Vashishtha et al., "Optifake: Optical flow extraction for deepfake detection using ensemble learning technique," *Multimedia Tools and Applications*, pp. 1–19, 2024.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025

[13] B. Yan, C. T. Li, and X. Lu, "JRC: Deepfake detection via joint reconstruction and classification," *Neurocomputing*, p. 127862, 2024.
[14] S. K. Panda, T. Diwan, O. G. Kakde, and J. V. Tembhurne, "Improvised detection of deepfakes from visual inputs using lightweight deep ensemble model," *Multimedia Tools and Applications*, vol. 82, no. 13, pp. 20101–20118, 2023.
[15] S. A. Khan and D. T. Dang-Nguyen, "Deepfake detection: Analysing model generalisation across architectures, datasets and pre-training paradigms," *IEEE Access*, vol. 11, 2023.
[16] S. R. Ahmed and E. Sonuç, "Evaluating the effectiveness of rationale-augmented convolutional neural networks for deepfake detection," *Soft Computing*, pp. 1–12, 2023.
[17] R. U. Maheshwari and B. Paulchamy, "Securing online integrity: A hybrid approach to deepfake detection and removal using Explainable AI and Adversarial Robustness Training," *Automatika*, vol. 65, no. 4, pp. 1517–1532, 2024.
[18] N. Kumar and A. Kundu, "Cybersecurity-focused deepfake detection system using big data," *SN Computer Science*, vol. 5, no. 6, p. 752, 2024.
[19] S. Kingra, N. Aggarwal, and N. Kaur, "SiamNet: Exploiting source camera noise discrepancies using Siamese network for deepfake detection," *Information Sciences*, vol. 645, p. 119341, 2023.
[20] S. Mathews, S. Trivedi, A. House, S. Povolny, and C. Fralick, "An explainable deepfake detection framework on a novel unconstrained dataset," *Complex & Intelligent Systems*, vol. 9, no. 4, pp. 4425–4437, 2023.
[21] F. Khalid, A. Javed, H. Ilyas, and A. Irtaza, "DFGNN: An interpretable and generalized graph neural network for deepfakes detection," *Expert Systems with Applications*, vol. 222, p. 119843, 2023.
[22] A. H. Soudy et al., "Deepfake detection using convolutional vision transformers and convolutional neural networks," *Neural Computing and Applications*, vol. 36, no. 31, pp. 19759–19775, 2024.
[23] M. Nawaz, A. Javed, and A. Irtaza, "ResNet-Swish-Dense54: A deep learning approach for deepfakes detection," *The Visual Computer*, vol. 39, no. 12, pp. 6323–6344, 2023.
[24] R. R. Sekar, T. D. Rajkumar, and K. R. Anne, "Deep fake detection using an optimal deep learning model with multi-head attention-based feature extraction scheme," *The Visual Computer*, pp. 1–18, 2024.
[25] Y. Patel et al., "An improved dense CNN architecture for deepfake image detection," *IEEE Access*, vol. 11, pp. 22081–22095, 2023.
[26] A. Almestekawy, H. H. Zayed, and A. Taha, "Deepfake detection: Enhancing performance with spatiotemporal texture and deep learning feature fusion," *Egyptian Informatics Journal*, vol. 27, p. 100535, 2024.
[27] N. U. Huda, A. Javed, K. Maswadi, A. Alhazmi, and R. Ashraf, "Fake-checker: A fusion of texture features and deep learning for deepfakes detection," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 49013–49037, 2024.
[28] M. Ahmed, M. H. Sulaiman, A. J. Mohamad, and M. Rahman, "Gooseneck barnacle optimization algorithm: A novel nature-inspired optimization theory and application," *Mathematics and Computers in Simulation*, vol. 218, pp. 248–265, 2024.
[29] A. J. Arunnehru et al., "Target object detection from unmanned aerial vehicle (UAV) images based on improved YOLO algorithm," *Electronics*, vol. 11, no. 15, p. 2343, 2022.
[30] A. Alhussen et al., "XAI-RACapsNet: Relevance-aware capsule network-based breast cancer detection using mammography images via explainability O-net ROI segmentation," *Expert Systems with Applications*, vol. 261, 2024.

DOI: https://doi.org/10.54216/FPA.180217
Received: August 10, 2024 Revised: November 18, 2024 Accepted: January 27, 2025