COMPUTOLOGY: Journal of Applied Computer Science and Intelligent Technologies
Volume 3, Issue 2, Jul-Dec 2023, pp. 69-97
DOI: 10.17492/computology.v3i2.2304
https://www.journalpressindia.com/cjacsit

Ensemble WGAN (EWG): Advancing Image Synthesis and Deepfake Detection with Heterogeneous Discriminator Approach

Preeti Sharma[1]*, *Manoj Kumar*[2] *and Hitesh Kumar Sharma*[1]

ABSTRACT

In the present time, deepfakes pose a big threat to the security of our society. Concerns regarding these fake images being used for malevolent reasons on social networking sites have increased. As a solution it, this paper proposed a new model called EWG (Ensemble WGAN) which helps to detect deepfake using its unique ensemble architecture. The EWG model is an expansion of the WGAN architecture that improves deepfake detection and GAN training issues. It employs a voting ensemble of three unique discriminators and a single generator. The approach works with generator weights updated by the best discriminator on each epoch. The model dynamically selects the best discriminator based on a unique diverse loss function that combines adversarial loss and the SSIM metric, boosting diversified performance. Leveraging the "Indian Actor Images Dataset" and "5-Celebrity Faces," the EWG model achieves remarkable deepfake detection accuracy of 98.480% and 96.417%, with computation times of 1813.251 and 2197.011 seconds. Furthermore, it mitigates GAN training challenges like mode collapses, gradient penalties, and convergence and provides superior image quality, surpassing basic WGAN and other state-of-the-art methods. The EWG model demonstrates its dependability and potential for countering deepfakes and improving GAN capabilities.

Keywords: *Deep Learning; Digital Forensics; Generative Adversarial Networks (GAN); Ensemble GAN Model; Deepfake.*

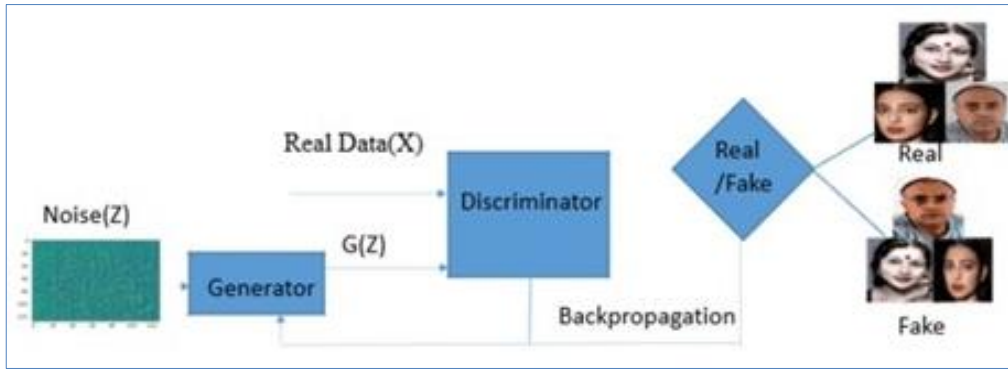
1.0 Introduction

Generative Adversarial Networks (GANs) are a subclass of generative models

[1]*School of Computer Sciences, UPES, Dehradun, Uttarakhand, India*
[2]*Engineering and Information Sciences, University of Wollongong, Dubai, UAE*
**Corresponding author e-mail: preetiii.kashyup@gmail.com*

that excel in extracting new, convincing visual information from the probability distribution of a dataset. These are a particular kind of generative model that generates synthetic images through adversarial training. Deepfake is an application of GAN for creating fake images and videos that are difficult to identify with the naked eye. With the fast growth of advanced computing capabilities and GAN advancements, spotting deep fakes is becoming more and more challenging. GAN variants are currently employed to generate still pictures from text data, move images from still images, boost image resolution, and modify images. Applications of this technology vary from anomaly detection to chess game enhancements[1]. The initial GAN model was invented and released in 2014 by Ian J. Good fellow and his team[2]. Radford et al. [3]presented the Deep Convolutional Generative Adversarial Network (DCGAN) in 2016. DCGAN, which enhances and stabilizes training effectiveness for all GANs, is created by combining CNN and the Generative Adversarial Network (GAN). Instead of maximizing likelihood as in traditional generative models, GAN advances the adversarial learning between the generator and the discriminator. Generative adversarial networks occur in a variety of various forms, each of which functions slightly differently and helps in obtaining optimized different result[4]. Numerous GAN modifications are made to the architecture side or the loss function side in order to boost efficacy[5][6][7]. The basic GAN model is shown in Figure 1.

Figure 1: Basic GAN Model



Instead of the advancement of GAN, there are a few common problems like vanishing gradients, mode collapse and convergence that are specific to GAN training. Vanishing gradients is the condition that occurs when the model impede or stop learning entirely. Mode collapse occurs when the generator distributes diverse inputs to the same

class at the output, producing instances that are extremely non-diversified. Instable, oscillating, and divergent behaviour while training the generator and discriminator leads to non-convergence [8]. Numerous GAN approaches have been created to overcome these primary issues with GAN training. One of the approach that machine learning (ML) practitioners uses is to strategically combine or fuse the models called Ensembles [9]. In comparison to a single GAN, an ensemble of GANs can more precisely reflect the distribution of normal data and consequently help in manipulation detection. Evolutionary algorithms also working to find solution of these training issues. Modelling, optimization, and design are major areas where evolutionary algorithms have also been incredibly successful. An evolutionary algorithm's core function is to compare potential solutions and choose the best ones based on fitness[10].

From the discussion of various techniques, it is required to answer four important research questions which forms the basis of our proposed research:

Question 1: Can a new GAN architecture be developed that substantially improves GAN training issues?

Question 2: Do a GAN model offers an optimized method for detecting deep fakes in images shared on social media?

Question 3: Would the ensemble approach offer a way to overcome the significant constraint of working with challenging data set in deep learning models?

Question 4: Can the SSIM integrated loss technique act as a feasible solution to generate high-quality images for GAN?

As an answer of these questions a new technique called Ensemble WGAN (EWG) is proposed in this paper that uses a variety of input samples to produce high-definition images while also improving training. Deepfake is classified in the discriminator section using various CNNs and majority voting ensemble approaches. Every epoch, the model selects the best discriminator by utilizing a separate loss function and a single generator with three distinct CNN-based discriminators. In two datasets, Indian Actor Images and 5 Celebrity Faces, the model achieves good accuracy (98.480% and 96.417%) and optimizes values (10.4, 32.71) and (11.5, 37.52), respectively, outperforming previous techniques. The model also integrates an improved metric known as SSIM, which evaluates the validity and quality of generated images by taking structural features, contrast, and brightness into consideration.

1.1 Contributions

i. Proposing EWG model as an enhanced WGAN based voting ensemble model that works incredibly well to generate good quality images and improve GAN

pathologies up to a considerable extent.

ii. A new upgraded approach to WGAN using minimizing new objective Function $\mathbf{E} * = \mathbf{min} \sqcup\_(\mathbf{i} = \mathbf{1})\mathbf{\hat{}3}\ \mathbf{ei}\{\mathbf{g}, \mathbf{di}\}$ to perform pairwise evaluation of separately trained networks.

iii. Locating a new diverse Loss function "$\mathbf{E} * + \boldsymbol{\alpha} * \mathbf{SSIM} = \mathbf{min}\ (\mathbf{W\_loss1}, \mathbf{W\_loss2}, \mathbf{and\ W\_loss3}) + \boldsymbol{\alpha} * \mathbf{SSIM}$" which is required to be minimized to promote the generated images to be visually and structurally similar, and adversarial convincing. It further helps to classify deepfake by penalizing image that deviate from original one in both pixel and structural information.

iv. Implementing SSIM as an integrated GAN metric assists in detecting expected deepfakes by calculating similarity between two images. SSIM supports EWG model by accounting for the images' structural details, brightness, and contrast.

v. An improved technique that outperforms the existing GAN models with optimised values of evaluating parameters IS, SSIM, FID and total computation time function leading to great performance and creation of real like pictures while mitigating the GAN pathologies.

Rest of paper is organised with subsequent sections. Sections 2 presented related research of the domain. Section 3 defines the proposed methodology including the details about the dataset and defined architecture. Section 4 demonstrates the experimentation section. Section 5 defines about the result and discussion section. It demonstrates various result graphs showcasing the optimised values of Accuracy, Loss function, GAN mode collapse, convergence, and gradient penalty improved graphs. At the end, section 6 presents the conclusion of the research paper.

2.0 Related Work

In this section, we have some previous studies of GAN and its variants that have been designed to reduce training instability and improve generative performance. Following that, we provide a brief overview of WGAN with evolutionary techniques which is used as basic architecture for implementation of our proposed model.

2.1 GAN and its variants

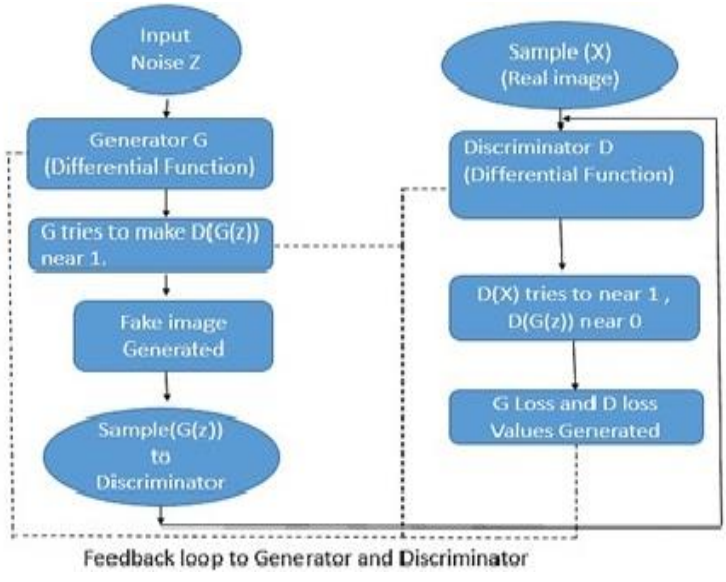
The initial GAN model, an unsupervised generative model that finds data distribution automatically, was created in 2014 by Ian J. Goodfellow et al. The "Generator" (a convolutional network) and the "Discriminator" (a deconvolutional network) are the two primary models that make up GANs. To create synthetic data for

unsupervised learning, GANs employ a supervised learning method as shown in flowchart in Figure 2. The model assumes the structures of data sets without considering predicted, labelled, or categorised outcomes. The GAN formula can be found in equation 1, which also contains its definition [2].

$$V(D,G) = Ex \sim pdata(x)[logD(x)] + Ez \sim pz(z)[log(1 - D(G(z)))] \quad \ldots(1)$$

In this context, G stands for "Generator," D for "Discriminator," $Pdata\ (x)$ denotes the distribution of real-world data, $P(z)$ denotes the distribution of generated data, x denotes a sample taken from $P\ (x)$, and $D(x)$ denotes a network representing the discriminator; $G(z)$ denotes the network representing the generator.

Figure 2: GAN Execution Flowchart



These are based on noisy contrastive estimates and the loss function used in modern GANs. In order to employ picture enhancement techniques that produce high-quality graphics, GAN using human faces has really been used since 2017 [11]. Semi-supervised and reinforced learning are also possible options, however learning and probabilistic data generation are necessary for GAN networks. Unsupervised machine learning algorithms account for the majority of GAN network implementations.

For applications including image-to-image interpretation, frame-by-frame prediction, and text-to-image synthesis, GANs have been effectively applied [12].

Though often exhibiting a trade-off in the PSNR/SSIM metrics, models are improving perceptually and can reconstruct more detailed pictures. The highest potential pixel value (L), often referred to as the dynamic range, is calculated by dividing it by the Mean Squared Error (MSE)[13], sometimes referred to as the L2 loss, between the pictures. Three further independent components—luminance, contrast, and structure—are included in SSIM [14]. The following equation may be used to get the MSE, PSNR, and SSIM between two images:

$$MSE = 1\,N \sum \|X\,(i) - XSR \quad \ldots(2)$$

$$2\,N\,i = 1;\; PSNR = 10\,log10\,L\,2\,MS \quad \ldots(3)$$

$$SSIM\,(X, XSR) = (2\,\mu X\,\mu XSR + C1)\,(2\,\sigma X\,XSR + C2)\,(\mu X\,2 + \mu XSR\,2 + C1)\,(\sigma X\,2 + \sigma XSR\,2 + C2) \quad \ldots(4)$$

This paper presents an evolutionary method for detecting deepfakes utilizing CNN-based ensemble discriminator architecture with enhanced SSIM, IS, and FID approach, improving support for the fields of mechanics, finance, and medical. The growing problem of fraudulent video generation and detection is addressed by this method.

2.2 Deepfakes

Advanced procedures like face generation, face manipulation, and face swapping called Deepfakes[15]. The core idea underlying deepfake technology is the deployment of deep convolutional generative adversarial networks (DCGANs). The deepfake technique was allegedly invented in November 2017 on the social media platform Reddit by an unidentified person shown in Figure 3.

Figure 3: Generation of Fake Images from Real Images using Deepfake

The user's source code was uploaded to GitHub, one of the two most popular sites for code sharing, in December of the same year to facilitate the developer community cooperating and advancing the idea. A few media reconstruction methods that use GAN include Cycle GAN, PGGAN, Big GAN, and Style GAN[16]. Convolutional neural networks are employed in a variety of GAN implementations to improve model quality It is done with the aid of Neural Net techniques that can detect regions and faces, as well as neatly modify them, and Specific Datasets (human faces, shapes, figures, etc.)[17].

GAN models, which are intended for image processing, identify deepfake problems using basic convolutional filters. Nevertheless, deeper networks might need consolidated data and more processing units. The notion and analysis of GAN are justified by literature. These findings included:

1. GAN model challenges on visual data reconstruction
2. database and training stability issues of GAN visual enhancements and prediction applications,
3. GAN model comparative studies and performance evaluation schemes, and
4. advances in ensemble architecture for GAN reconstruction models.

2.3 Wasserstein GAN

A generative adversarial network known as a Wasserstein GAN, or WGAN in [7],is basically used for implementing the Ensemble approach. By substituting a critic for the discriminator model, the generative adversarial network known as the WGAN improves training stability and quality. The Earth-distance Mover's approximation is minimized, and the difference between training data and generated instances is decreased. The dependability and less dependence on model design and hyperparameter settings are the advantages of the WGAN. The discriminator's loss is correlated with the generator's picture-quality output. WGAN is different from other implementations:

1. Instead of sigmoid it employs a linear activation function in the output layer of the critique model.
2. Apply a -1 label to genuine photographs and a 1 label to false images (instead of 1 and 0).
3. Apply Wasserstein loss to critic and generator model training.
4. After each micro batch update, restrict critic model weights to a certain range (e.g. [-0.01,0.01]).
5. Each cycle should include more updates to the critic model than to the generator.

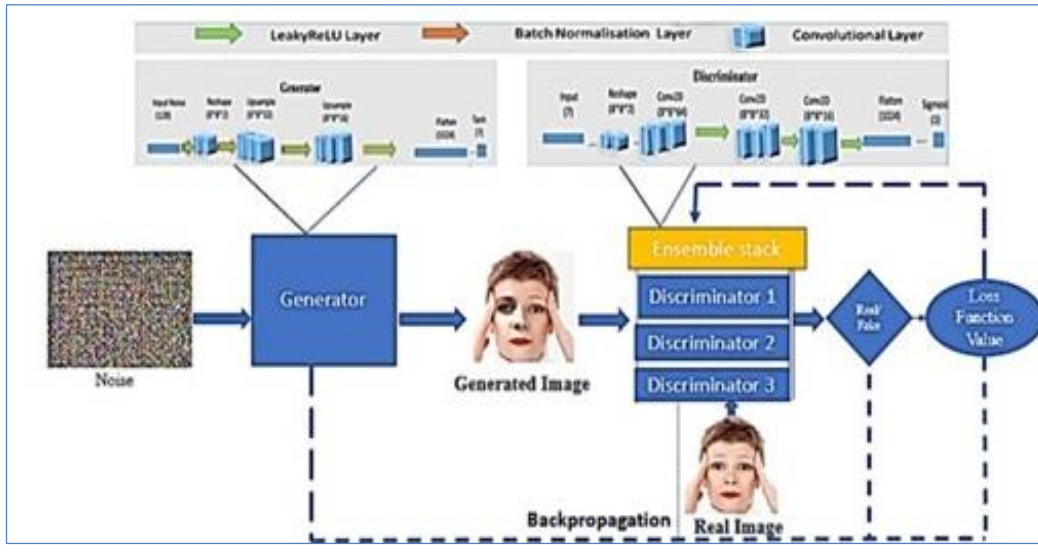
6. Employ gradient descent using the RMS Prop algorithm with a slow learning rate and no momentum (e.g., 0.00005).

3.0 Proposed Methodology

3.1 EWG model description

The suggested model is an ensemble-based extension of Wasserstein GAN (WGAN), recreating WGAN using a voting ensemble technique and an ensemble of three discriminators and a single generator. The generator's weights for a given epoch are changed by selecting the discriminator with the lowest Diverse_loss value as shown in equation 5 below. By decreasing the distance between the created and original distributions, the model employs Wasserstein distance (W_loss) to improve training consistency. Using a majority vote ensemble technique and many CNN models, the discriminator component identifies deepfake. The objective is to enhance the gradients of the generator by utilizing the weights that have been updated by a certain discriminator.

Figure 4: EWG Model Showing Modified WGAN with Ensemble Technique using Single Generator Coupled with Three Different Discriminators



To optimize learning and yield optimal outcomes, the method is combined and implemented through concurrent generator and discriminator model training. This strategy is intended to establish the following criteria's:

a. a more discriminator D (better approximating Min $W(D,G)$);
b. a Ds 's ensemble better suited to the generator's G capabilities; and
c. overcoming common training issues of existing GANs, also including local feature ambiguity and global structure failure. The perfect candidate for creating strong ensembles is chosen the best discriminator using diverse loss function. The block diagram of the proposed GAN model is shown in Figure 4.

The approach shows optimize training outcome with good quality of generated images as mentioned in Figure 4. In this work, generative models are repurposed to produce samples in accordance with a objective function $E* = minWi\ or \prod (i = 1)\text{^}3\ ei\{g, di\}$ to perform pairwise evaluation of separately trained network mentioned in equation 5,6,7 and 8.In order to improve training consistency and eliminate pixel manipulations using Wasserstein distance and clipping techniques, the researchers created a new dataset utilizing Indian Actor Images and 5-Celebrity Faces using the EWG model. Diverse_loss calculates the minimum W-loss among three discriminators using the SSIM score, with the importance of the SSIM term controlled by a coefficient called α.

$$Diverse\_loss = E* + \alpha * SSIM = min\,(W_{loss1}, W_{loss2}, and\ W_{loss3}) + \alpha * SSIM. \quad \ldots(5)$$

$$E* = minWi\ or \prod (i = 1)\text{^}3\ ei\{g, di\} \quad \ldots(6)$$

$$ei = W\_loss\ (g, di) \text{ for i=1 to 3.} \quad \ldots(7)$$

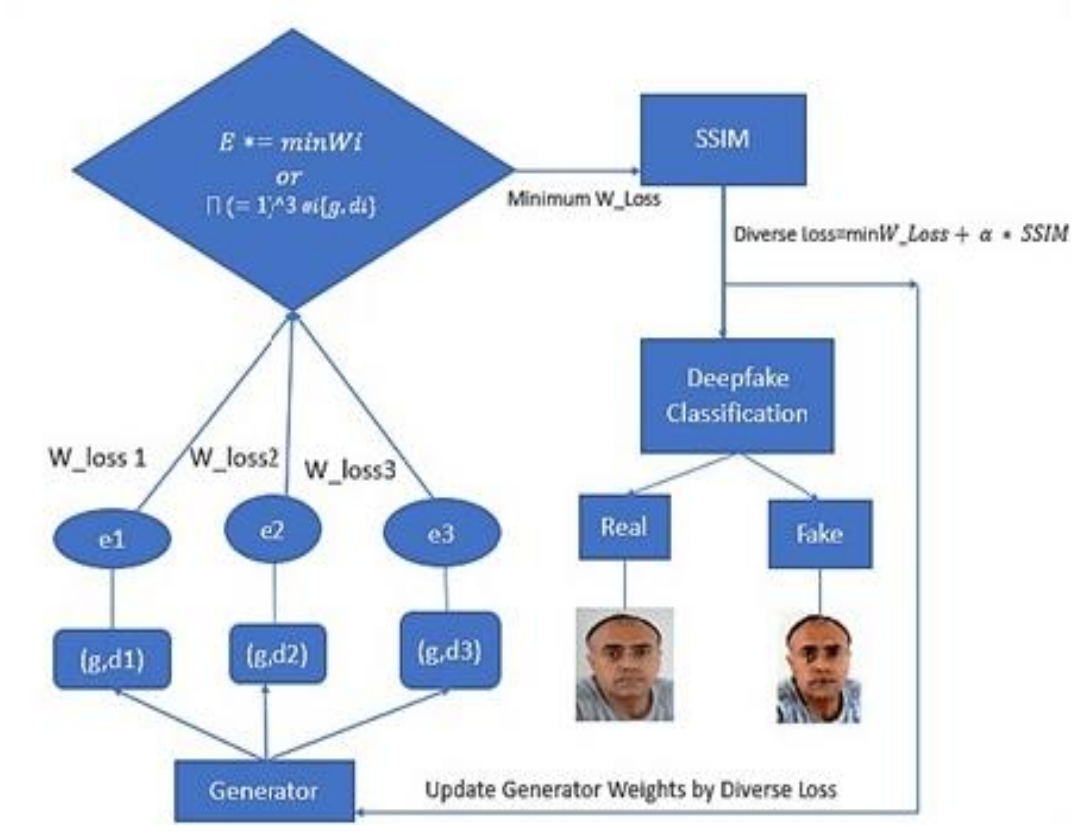
$$W\_loss = E\,[x \sim Pr][D(x)] - E[z \sim Pz][D(G(z))] \quad \ldots(8)$$

Where; E represents the optimized ensemble, ,ei represents the loss value calculated for each instance {g, di} within the ensemble. Each instance consists of a generator g and a discriminator di, and Wi represents the Wasserstein-1 or W-loss operator, which calculates the minimum value among the given set of loss values. Pr represents the actual data distribution. The noise distribution is shown by Pz. A sample of the actual data distribution Pr is represented by the variable x. Z represents a sample taken from the noise distribution. Pz. G(z) represents the generated sample obtained by running noise z through the generator G. When given a real data sample x, the discriminator D's output is represented by D(x). Given a generated sample G(z), D(G(z)) is the discriminator's output.*

The architecture of the EWG model comprising one generator and three discriminators is depicted in the Figure 5. Feedback from various discriminators is collected for the generator's training. G is trained against the best discriminator if $E* =$

$minWi\ or\ \prod (i = 1)\text{^}3\ ei\{g, di\}$ where ei represents the loss value calculated for $instance\ \{g, di\}$.

Figure 5: Proposed EWG (Ensemble WGAN) Model



3.2 Simplified Loss Function (Wasserstein Loss)

To convert the discriminator from a classifier to a critic, the WGAN model presents a novel loss function that seeks to maximize the difference in scores for true cases and minimize it for erroneous ones. By doing this, the loss for authentic and fake photos is reduced. Mathematically as the Wasserstein separation between the produced data distribution Pg and the real data distribution Pr is shown in equation 9:

$$W(\mathrm{Pr}, \mathrm{Pg}) = \inf_{\gamma \in \pi(Pr,Pg)} E(x,y) \sim \gamma^n [\|x - y\|] \quad \ldots(9)$$

Where, the set of all joint distributions γ (x, y) with marginals that are respectively Pr and Pg is denoted by the notation $\amalg (Pr, Pg)$.The Wasserstein distance

has a rigorous mathematical definition that was presented in [7]. If Pr to represent the real or current distribution and Pg to represent the desired or produced distribution. The cheapest transportation option is the Wasserstein distance. Infimum, or the highest value that is less than all the components in a set, is another name for this minimum loss.

3.4 Mathematical Notation of proposed GAN function

Function: Proposed GAN function.

```
def GAN (best discriminator, generator, input shape, latent dim):
# Discriminator definition:
discriminator = best discriminator (input shape)
discriminator. Compile (loss=“ binary_crossentropy,” optimizer=“Adam (lr=0.0002, beta_1=0.5)”
# Generator definition:
generator = generator (latent dim)
# GAN model definition:
gan = Sequential ()
gan. Add (generator)
gan. Add (discriminator)
gan. Compile (loss='binary_crossentropy', optimizer=Adam (lr=0.0002, beta_1=0.5)
return gan.
```

3.5 Algorithm

Algorithm: EWG: Ensemble WGAN Model.

Input: Real samples and generated samples are entered.

Output: Discriminator Loss and Generator Loss.

1. Set the Generator, D1, D2, and D3 to their initial values.
2. Set batch size = 128 and epochs=1000.
   For each batch:
   For every epoch:
3. Sample a batch of real samples.
4. Use the Generator to create a batch of fake samples.
5. Determine each discriminator's Wasserstein loss:
   -Determine the values of W1 = loss (D1, real samples) and W1_fake = loss (D1, generated samples).
   - Determine the values of W2 = loss (D2, real samples) and W2_fake = loss (D2, generated samples).
   - Determine the values of W3 = loss (D3, real samples) and W3_fake = loss (D3, generated samples).
6. Pick the discriminator that has the smallest Wasserstein loss:
   -If W1 > W2, and W1 > W3, use D1 for additional calculations.

- If W2 > W1 and W2 > W3, use D2 for additional calculations.
- If W3> W1 and W3> W2, D3 for additional calculations.

7. Calculate SSIM, IS, FID and Total Computational Time.
8. Calculate diverse loss = $min\ (W1, W2, and\ W3)\ +\ \alpha\ *\ SSIM$.
9. Update Generator weights using diverse loss:
   - Calculate generator loss= loss (Diverse loss, generated samples)
   - Adjust the weights in the Generator based on G_loss.
10. Update the Selected Discriminator:
    - Using the chosen Discriminator, compute the Wasserstein loss for real and fake samples.
    - Adjust the weights of the Discriminator to account for the Wasserstein loss.
11. Repetition of steps 3 to 10 for the batch's allotted number of iterations.
12. Repeat steps 2 to 11 for all specified epochs.

End: Training Process completed.

4.0 Experimentation

The proposed model is implemented using Jupyter (Anaconda framework) and specifically created for the parameter updates between the generator network and discriminator network, in contrast to the standard parallel framework. The input image is frequently a three channel RGB with size 64x64x3-pixels. The images have been acquired from Indian Actor Images and 5-Celebrity Faces datasets. The input image is frequently a three channel RGB with size (64x64x3) -pixel shown in Figure 6. The array of input images of both the datasets are mentioned below in Figure 7.

Figure 6: Input Images Size (64x64x3) used for Execution of Proposed Model

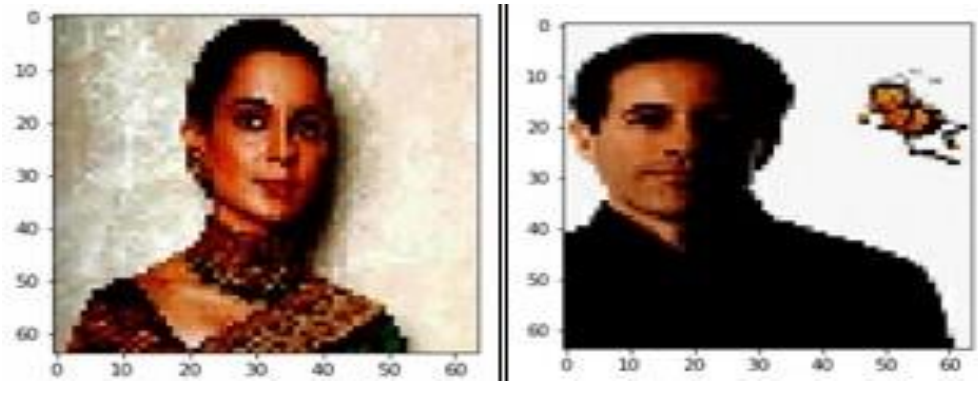
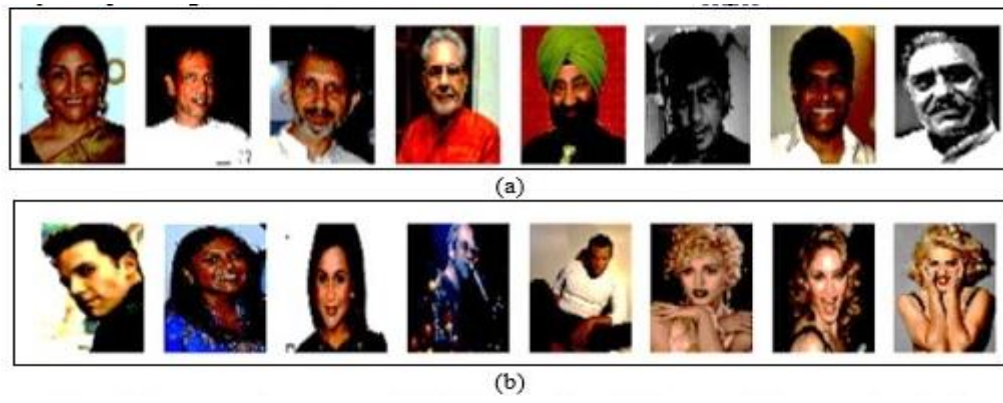


Figure 7: (a) Real Images from Indian Actor Images and (b) 5-Celebrity Faces Datasets used for Execution of Proposed Model [18][19]



4.1 Training

The generator is trained with a Gaussian random number sample, by inverting the labels and loss functions, and by producing batch norm statistics. Stride in convolutional layers is used to downsample the discriminator model, whilst deconvolutional layers can be used for upsampling. With deterministic spatial pooling functions, the network learns its own spatial downsampling. Convolutional layers are flattened and sent straight to the output layer by the generator, which also creates its own spatial upsampling and discriminator. The Gaussian input vector is converted into a multi-dimensional tensor, and the final convolution layer is flattened and fed into a single sigmoid output.To enhance image quality, the discriminator uses low random noise and label smoothing. Training and generation use a 50% dropout rate, and activations are standardized by batch normalization. After normalizing the inputs to the range [-1,1], tanh is employed in the generator output. All models include batch norm layers, with the exception of the discriminator's input and output. In classification tasks, GANs are trained using Binary Cross Entropy (BCE).

4.2 Dataset

To demonstrate the model's interoperability in various circumstances, two datasets were newly created using 93 images of each celebrity from the Indian Actor images and the 5-celebrity images dataset, respectively. The Indian celebrity dataset proved to be a more effective for proposed model. The data set is freely accessible for download at: https://www.kaggle.com/datasets/dansbecker/5-celebrity-faces-dataset [18]

and https://www.kaggle.com/datasets/iamsouravbanerjee/indian-actor-images-dataset [19] respectively.

4.3 Input parameters

The GAN model's architecture and behaviour are defined by these parameters during training and generation shown in Table 1.

Table 1: Major input parameters used in the Implementation.

Parameters	Value
Batch Size	128 pictures
Learning Rate	0.0002
Momentum (Beta)	0.5
Optimizer	Adam's Optimizer
Epoch	1000
Loss Function	Wasserstein Loss
Dataset	Indian Actor Images and 5-Celebrity Faces datasets
Evaluation Parameters	IS, SSIM, FID and Total Computational Time

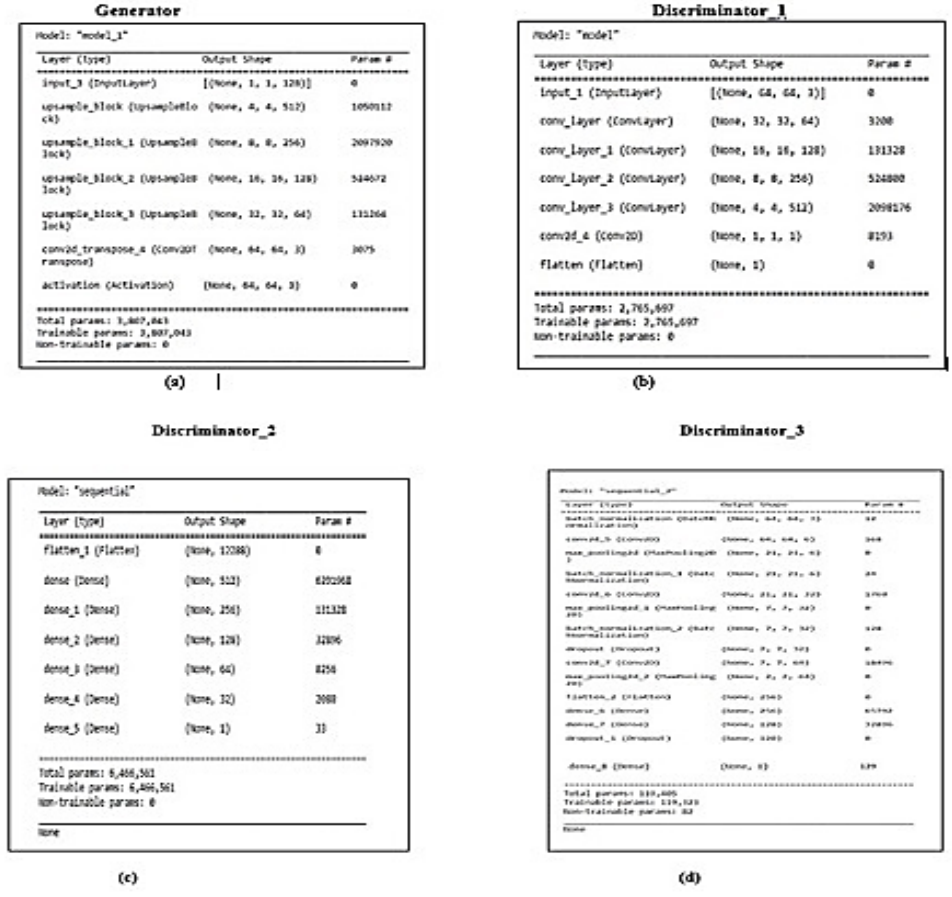
The hyperparameters scheme followed for EWG model training is:

1. **Hp drop rate:** The difficulty, user involvement, and overall generator-discriminator minimax game experience are all considered throughout the design and balancing of the game. To create balance and dynamics, the HP loss rate is adjusted through iterative testing and player feedback. The float variable hp drop rate, which has a range of 0-0.9, affects the dropout layer drop rate of the discriminator.
2. **Batch norm:** A popular method for ensuring training stability and convergence in GAN models, especially in complicated architectures, is batch normalization. The issue complexity, dataset characteristics, and intended GAN model performance all influence the batch normalization option. The Boolean variable batch norm controls the layers of batch normalization.
3. **Activation function:** In GAN models, the activation function is essential since it establishes the nonlinearity of each neuron's output range. ReLU uses the activation function specified by string variables, which minimizes the vanishing gradient problem and speeds up the convergence of the model.

4.4 Layers architecture used by generator and 3-discriminators in EWG model

It defines the structure of the networks' discriminator and generator. The generator and discriminator architectures of a GAN frequently employ the convolutional neural network (CNN). Figure 8 specifies for each discriminator and the generator, the quantity, size, and type of layers (e.g., fully connected, convolutional, recurrent). (a), (b), (c), and (d) shows comparable or slightly different architectural designs for generator and each discriminator and in the ensemble respectively. The generator and discriminator typically have the following layered architectures:

Figure 8: (a), (b), (c) and (d) Depicts Layered Architecture of Generator and Discriminator 1, Discriminator 2, and Discriminator 3 Respectively used in EWG Model



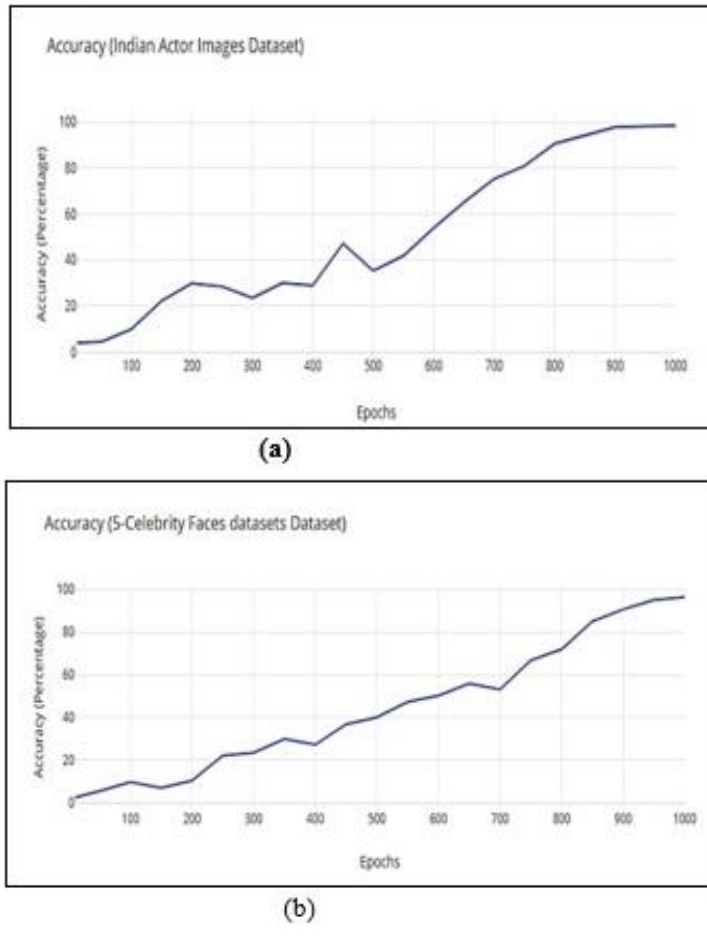
5.0 Results and Discussion

The goal of the research is to produce high-quality images by resolving non-convergence, vanishing gradients, and mode collapse in training. The model increases image diversity to optimize output. The Inception Score (IS), Fréchet Inception Distance (FID), SSIM, and total computational time function are among the evaluation parameters. With good accuracy (98.480% and 96.417%) and a total processing time of only 1813.251 seconds and 2197.011 seconds for both datasets, the model surpasses previous approaches. Additionally, the model includes an improved measure called SSIM, which computes image similarity to aid in the detection of predicted deepfakes. This aids in locating distortions, artifacts, or inconsistent patterns produced by the generator throughout the deepfake creation process. A critical metric for evaluating the degree of resemblance between real and GAN-generated images is the Structural resemblance Index (SSIM), where a score of 0.25 indicates a high probability of deepfake because it highlights the differences between the two. Figure 9 shows the quality of the images for successive epochs. It is observed that for every epoch the quality of images gets improved and generate more visible faces.

Figure 9: Generation of Images by Proposed EGW Model with Successive Epochs

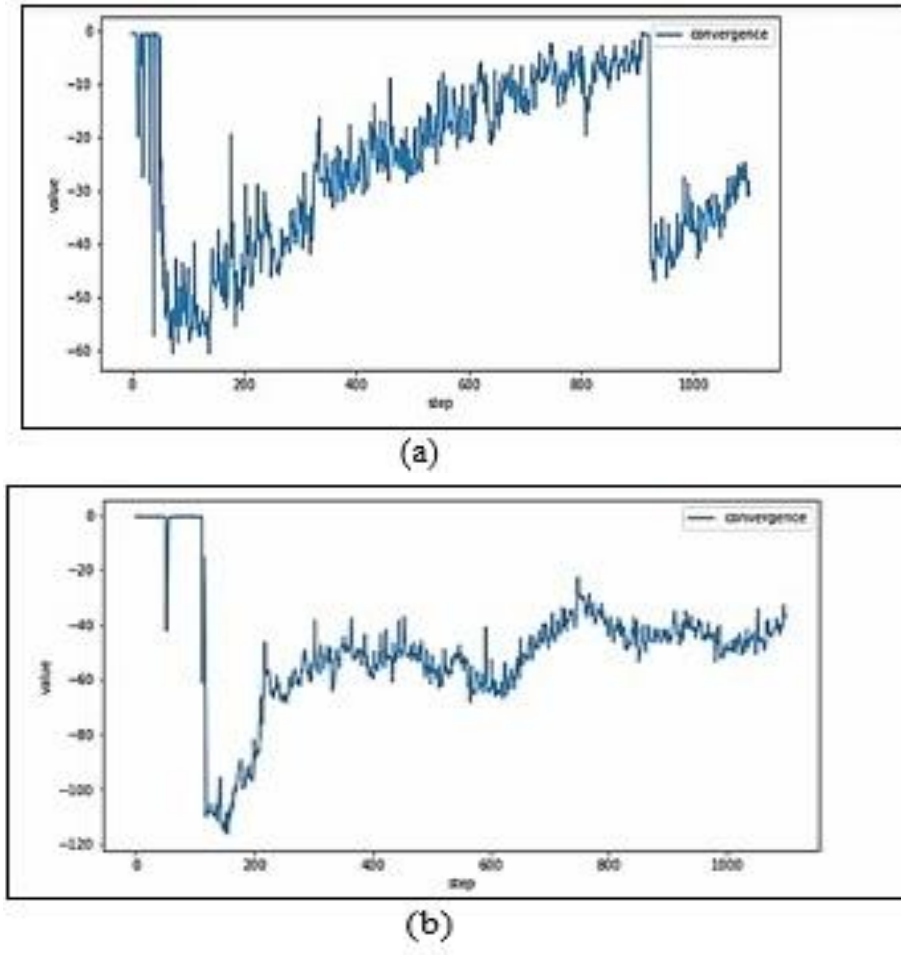
In the Indian celebrity picture dataset as well as the five celebrity image datasets, discriminator D3 performs better than the other three in terms of accuracy. It operates with three times the efficiency of the other two, increasing accuracy from 4% in the first epochs to 32%. D3's accuracy increases significantly with the size of the datasets; it reaches 98.48% for Indian celebrity photographs and 96.417% for five celebrity image datasets [Figure 10].

Figure 10: Accuracy Graphs Generated by EWG Model for Best Chosen Discriminator (D3) for (a) Indian Actor Images and (b) 5-Celebrity Faces Datasets



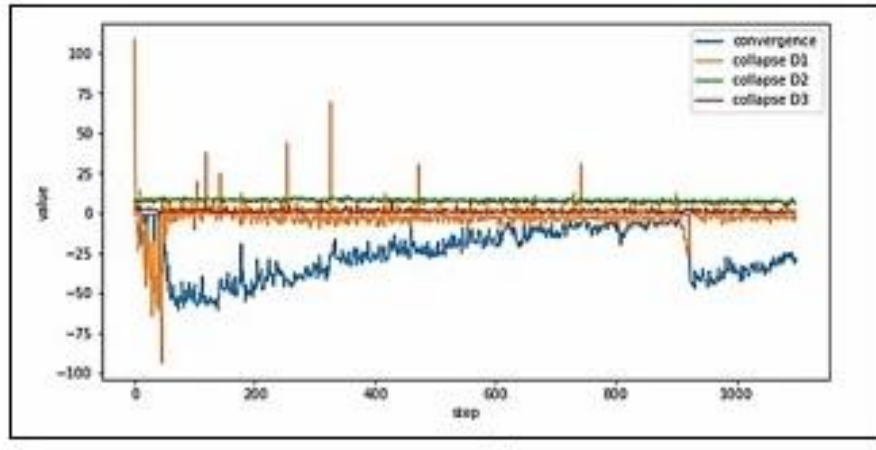
The convergence range to a upper bound value 0.4 increasing from – 0.25 shown in Figure 11. Gradient Penalty shows a drop from to 12.5 to 0.5 and get saturate at the final range. Also, it improves quite well with mode collapse problem getting fixed it to a value 0.7 dropping from 120 for Indian celebrity images dataset and 0.87 dropping from 380 to 0.9 for 5 Celebrity Image dataset as shown in Figure 12 and 13. These are the significant values achieved by the model for both datasets using discriminator 3.

Figure 11: Convergence Graphs Generated by Proposed EWG Model for (a) Indian Actor Images and (b) 5-Celebrity Faces Datasets

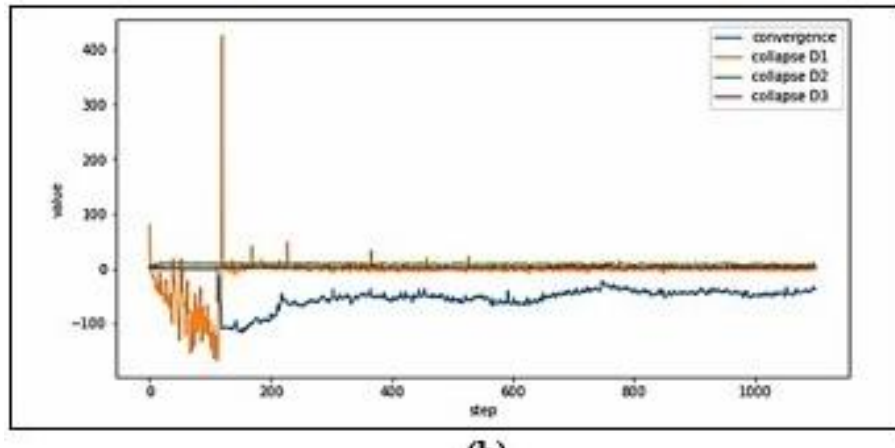


GAN networks' vanishing gradient issue is solved with activation functions like ReLU, which also encourage sparse activations (e.g., lots of zero values). We've used the generator and discriminator leaky ReLU. For the generator model, ReLU is advised, but not for the discriminator model. Instead, a variant of ReLU known as Leaky ReLU, which permits values lower than zero, is favoured in the discriminator.

Figure 12: Mode Collapse and Convergence Graphs Generated by Proposed EWG Model for (a) Indian Actor Images and (b) 5-Celebrity Faces Datasets

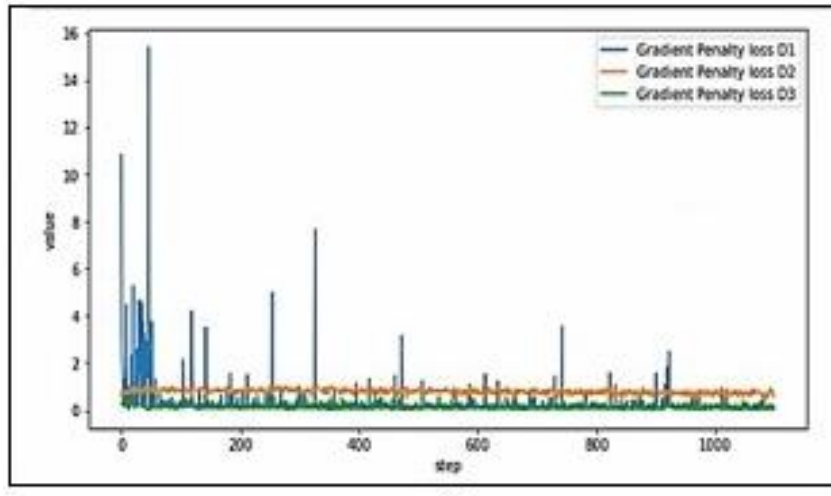


(a)

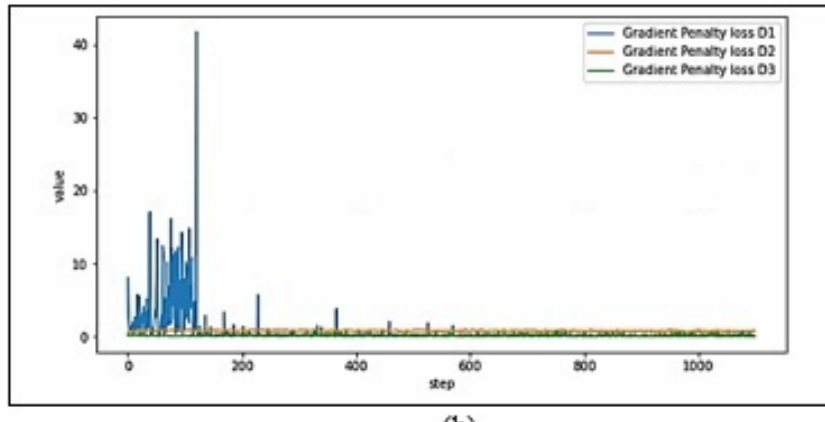


(b)

Figure 13: Gradient Penalties Generated by Proposed EWG Model for (a) Indian Actor Images and (b) 5-Celebrity Faces Datasets



(a)



(b)

The computational performance of model is analysed by calculating the total time taken by each epoch for execution. It is notified that the proposed model has taken very less time for execution for both the datasets. For Indian celebrity dataset each epoch get executed in around 17.274 seconds while for 5 Celebrity dataset it comes around 18.331 seconds initially. The total time taken by model for execution of 1000 epochs for both dataset is around 1813.251 seconds and 2197.011 seconds respectively. The graphical analysis of achieved results for computational analysis is presented in the Figure 14 and Figure 15.

Figure 14: Computational Time Graphs Bar Graph (a) and Pie Chart (b) Generated by Proposed EWG Model using Indian Actor Images

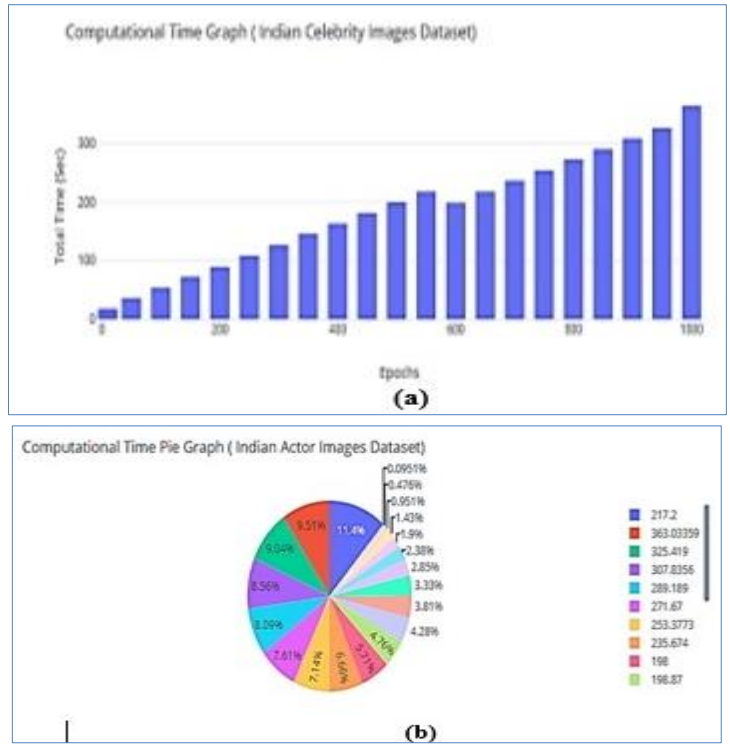
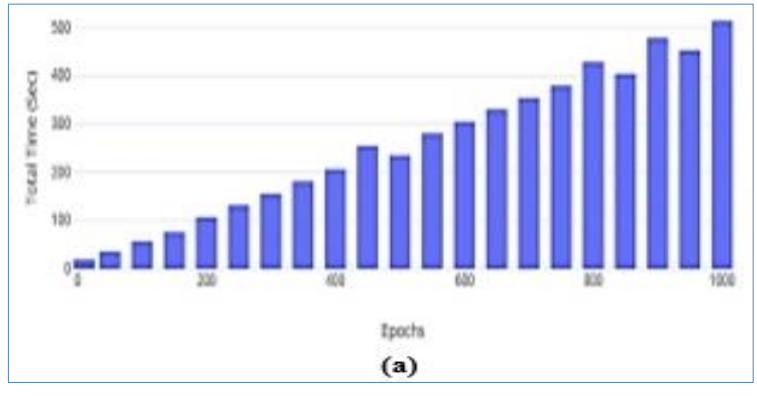
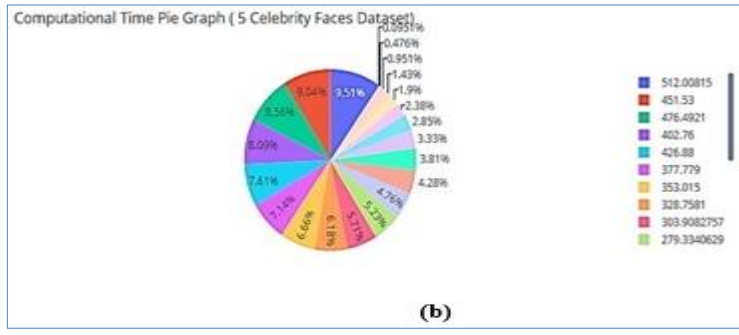


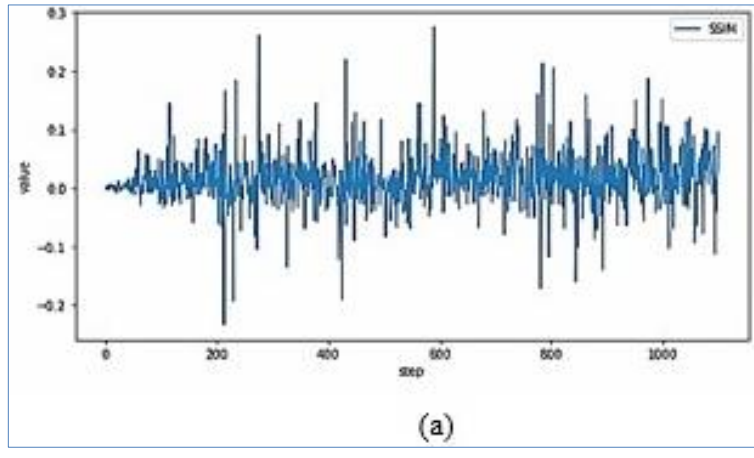
Figure 15: Computational Time Graphs Bar Graph (a) and Pie Chart (b) Generated by Proposed EWG Model using 5-Celebrity Faces

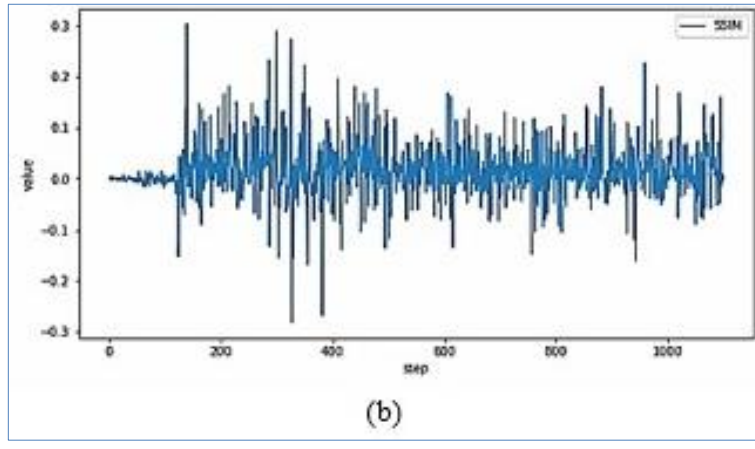




The Structural Similarity Index (SSIM) is a visual metric that measures the loss in image resolution imposed by image processing. It is proved highly useful to compare the quality of real images to those of images generated by generator using a discriminator at a time. The optimum value achieved is 0.25 and 0.3 for both datasets respectively, which indicates about high-quality reconstruction technique as shown in Figure 16.

Figure 16: SSIM Graphs Generated by Proposed EWG Model for (a) Indian Actor Images and (b) 5-Celebrity Faces Datasets





A quantitative metric called the Structural Similarity Index (SSIM) is used to automatically evaluate the perceived quality of images, which helps detect Deepfake. With a decent score of 0.25, this state-of-the-art parameter shows how similar the suggested GAN-generated images are to the genuine ones.as illustrated in Figure 17.

Figure 17: (a) Real Image used for Training EWG Model(b) Generated Images by the EWG Model

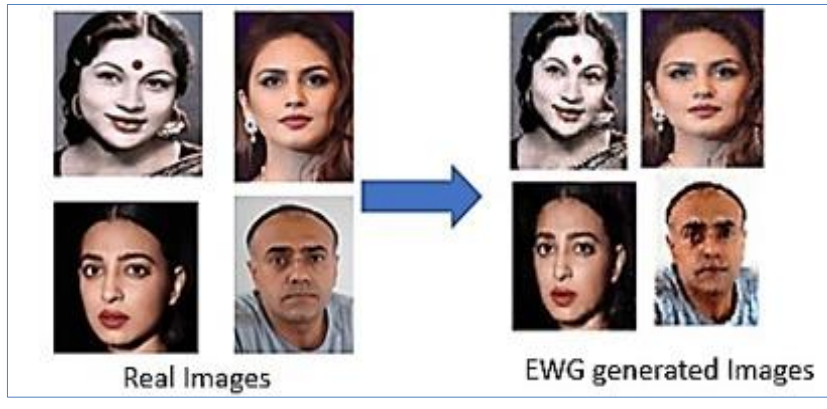


Table 2 shows the comparative score of EWG model with existing WGAN model. It compares the results of the EWG model and the existing WGAN model on basis of four evaluation parameters IS, FID, SSIM and Total computational time.

Table 2: Showing the Comparative Scores of IS, FID, SSIM and Total Computational Parameters of EWG Model with Existing WGAN Model

Evaluation Parameters	**EWG* Model (**Indian Actor Images dataset)	**EWG* Model (**5 Celebrity Faces datasets)	**Basic WGAN Model (**Indian Actor Images dataset)	**Basic WGAN Model (**5 Celebrity Faces dataset)
IS	10.4	11.5	7.06	7.18
FID	32.71	37.52	54.46	55.90
SSIM	0.25	0.3	0.0313	0.00454
Total Computational time (Seconds)	1813.251	2197.011	4970.271	4950.913

*(Note: * represent proposed EWG model)*

On the Indian Actor Images and 5 Celebrity Faces datasets, the EWG* model had a higher FID score, suggesting a greater ability to distinguish between the created and real data distributions. With a higher FID score, the WGAN model can distinguish more clearly between the produced and actual data distributions. The structural similarity between the generated samples and the real data is measured by the SSIM (Structural Similarity Index). In comparison to the WGAN model (0.0313 and 0.00454), the EWG* model has a higher SSIM score (0.25 and 0.3), indicating that the generated images are of higher quality and more akin to genuine images. The WGAN model computes in 4970.271 seconds and 4950.913 seconds, respectively, whereas the EWG* model computes at 1813.251 seconds on the Indian Actor Images dataset and 2197.011 seconds on the 5 Celebrity Faces dataset. These findings suggest that the suggested EWG model performs better at generating diverse and realistic samples.

According to the study, the suggested EWG* technique received good accuracy scores—98.480% and 96.417%, respectively—on the 5-celebrity Faces and Indian celebrity Faces datasets. On the CelebA dataset, Benny et al. likewise attained high accuracy scores.On the CIFAR10 dataset, an accuracy score of 69.51%, and 98.90% on the MNIST dataset was reported by [20], is considerably lower than that of our proposed EWG approach. On the CelebA dataset, [21] received accuracy ratings of 93.27% for baldness, 99.88% for eyeglasses, 95.68% for moustache, 94.62% for wearing a hat, and 98.62% for wearing a necktie. [22] had moderate accuracy scores. On the CIFAR10/CIFAR100 datasets, it obtained accuracy scores of 65.0/58.2%.

Table 3: **Showing the Comparative Scores of Accuracy Parameter of EWG Model (*) with Existing GAN Model**

Research	Dataset	Accuracy Score
EWG* Model	Indian celebrity Faces and 5-celebrity Faces	98.480% (Indian celebrity Faces) 96.417% (5-celebrity Faces)
[20]	CIFAR10 MNIST	69.51% 98.90%
[21] *(NFO SCC-GAN)*	CELEB A	93.27% (Bald) 99.88% (Eyeglasses) 95.68% (Mustache) 94.62% (Wearing Hat) 98.62% (Wearing Necktie)
[22]	CIFAR10 CIFAR100	65.0/58.2 (in %) 3.5/2.4 (in %)

The findings in Table 4 state the comparison of EWG model with other existing models on based of IS and FID parameters. It is noticed that the EWG Model performs better than most models indicating better image quality and diversity.

Table 4: Showing the Comparative Scores of IS and FID Parameters of EWG Model (*) with Existing GAN Model

GAN Variants	Dataset	IS	FID
EWG* Model	Indian Actor Images dataset 5 Celebrity Faces dataset	10.4 11.5	32.71 37.52
Deep Convolution GAN [23]	Celeb A	1.074	49.3
DC-GAN[24]	CIFAR 10 CIFAR 100	7.06 6.87	42.23 44.18
FC-GAN[25]	CIFAR 10	6.41	42.6
S-GAN [26]	CUB-200-2011	-	25.99 ± 4.26
BIG GAN[27]	JFT-300M	1.94	50.88

In terms of image quality, the EWG Model* performs better than the other models, achieving higher IS values (10.4 and 11.5) for both datasets. Its competitive FID scores (32.71/37.52) show that it can successfully fit real picture distributions. A lower IS value for the Deep Convolution GAN denotes reduced image variety and quality.

Optimized values of IS, FID, SSIM, total calculation time, and accuracy parameters show that the proposed EWG Model* performs better than current models in generating realistic and diverse samples. The performance of the model is contrasted with that of other models, such as BIG GAN, DC-GAN, and FC-GAN as discussed in Table 2,3 and 4. In nutshell, by utilizing a novel voting ensemble method and heterogeneous discriminators, the EWG model surpasses both IS and FID criteria in producing diverse and high-quality images. A lower computational time value denotes good efficiency and architectural design, whilst its optimal SSIM value implies high-quality reconstruction. This renders it a feasible option for diverse picture production and false detection assignments. Additional investigation and testing may shed additional light on the potential of the EWG model and its implications for practical uses.

6.0 Conclusion

It has been found that proposed EWG model works exceptionally well by utilizing its robust three discriminator-based ensemble architecture. The EWG Model's ensemble technique and heterogeneous discriminators contribute to its excellent performance, making it a strong contender for a variety of applications including image generation and deepfake detection. The development of the approach has made it possible to easily detect the manipulated images circulated on social networking sites achieving high accuracy values of 98.480% and 96.417%. The model also provides a solution to improves GAN training issues by utilizing SSIM integrated diverse loss function. It is used as an evaluation measure in the ensemble GAN model assisting in detecting expected deepfakes by calculating similarity between two images. The model is also proving computationally sound as requires only 1813.251 seconds and 2197.011 seconds for 1000 epochs. Optimized values of SSIM, IS, and FID parameters further authenticate the generation of high-quality images. It provides satisfactory answer to all stated research questions and notably confirms state-of-the-art performance to the existing models of the field. In term of future work, the model can be extended for a greater number of discriminators upgrading the technique of ensemble involved. The effective schemes and discriminator ensemble architectures can serve for more robust and efficient GAN models in commercial areas, such as, facial recognition, graphical enhancements, facial corrections, etc. The generalization of GAN for managing complex datasets with more effective assessment measures would also be a potential subject for future study in this discipline.

References

[1] Rezaei, M., Näppi, J. J., Lippert, C., Meinel, C. & Yoshida, H. (2020). Generative multi-adversarial network for striking the right balance in abdominal image segmentation. *Internationla Journal of Computer Assisted Radiology and Surgery, 15*(11), 1847–1858. Retrieved from doi:10.1007/s11548-020-02254-4.

[2] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2020). Generative adversarial networks. *Communication of ACM, 63*(11), 139-144. Retrieve from doi: 10.1145/3422622.

[3] Radford, A., Metz, L. & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. 4th International Conference Learn. Represent. ICLR 2016., pp. 1–16.

[4] Cao, Y.-J., Jia, L. L., Chen, Y. X., Lin, N., Yang, C., Zhang, B., Liu, Z., Li, X. X., & Dai, H.H. (2019). Recent advances of generative adversarial networks in computer vision. *IEEE Access, 7*(C), 14985–15006. Retrieved from doi: 10.1109/ACCESS.2018.2886814.

[5] Isola, P., Zhu, J. Y., Zhou, T. & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings - 30th IEEE Conference Computer Vision Pattern Recognition, CVPR 2017*, pp. 5967–5976. Retrieved from doi: 10.1109/CVPR.2017.632.

[6] Quan, F., Lang, B. & Liu, Y. (2022). ARRPNGAN: Text-to-image GAN with attention regularization and region proposal networks. *Signal Processing: Image Communication, 106*, 116728. Retrieved from doi: 10.1016/j.image.2022.116728.

[7] Arjovsky, M., Chintala, S. & Bottou, L. (2017). Wasserstein generative adversarial networks. *34th International Conference Machince Learning (ICML 2017)*, 1, 298–321.

[8] Khanuja, S. S. & Khanuja, H. K. (2021). GAN challenges and optimal solutions. *International Research Journal of Engineering and Technology (IRJET), 8*(10), 836–840.

[9] Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M. & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence, 115*, 105151. Retrieved from doi: 10.1016/j.engappai.2022.105151.

[10] Wu, Z., He, C., Yang, L. & Kuang, F. (2021). Attentive evolutionary generative adversarial network. *Applied Intelligence, 51*(3), 1747–1761. Retrieved from doi: 10.1007/s10489-020-01917-8.

[11] Aggarwal, A., Mittal, M. & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights, 1*(1), 100004. Retrieved from doi: 10.1016/j.jjimei.2020.100004.

[12] Wang, Y., Zhang, L. & van de Weijer, J. (2016). Ensembles of generative adversarial networks. Retrieved from http://arxiv.org/abs/1612.00991

[13] Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. (n.d.). The unreasonable effectiveness of deep features as a perceptual metric. Retrieved from https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.pdf

[14] Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. Retrieved from doi: 10.1109/TIP.2003.819861.

[15] Aduwala, S. A., Arigala, M., Desai, S., Quan, H. J. & Eirinaki, M. (2021). Deepfake detection using GAN discriminators. *IEEE 7th International Conference on Big Data Computing Service and Applications (BigDataService),* pp. 69–77. Retrieved from doi: 10.1109/BigDataService52369.2021.00014.

[16] Xie, Y., Lin, T., Chen, Z., Xiong, W., Ran, Q. & Shang, C. (2022). A lightweight ensemble discriminator for generative adversarial networks. *Knowledge-Based System, 250*, 108975. Retrieved from doi: 10.1016/j.knosys.2022.108975.

[17] Sharma, P., Kumar, M. & Sharma, H. (2022). Comprehensive analyses of image forgery detection methods from traditional to deep learning approaches: An evaluation. *Multimedia Tools and Applications*, *82*(12), 18117-18150.

[18] https://www.kaggle.com/datasets/dansbecker/5-celebrity-faces-dataset

[19] https://www.kaggle.com/datasets/iamsouravbanerjee/indian-actor-images-dataset.

[20] Yaniv, B., Galanti, T., Benaim, S. & Wolf, L. (2021). Evaluation metrics for conditional image generation. *International Journal of Computer Vision, 129*, 1712-1731.

[21] Kinakh, V., Drozdova, M., Quétant, G., Golling, T. & Voloshynovskiy, S. (2021). Information-theoretic stochastic contrastive conditional GAN: InfoSCC-GAN. arXiv preprint arXiv:2112.09653.

[22] Castro, F. M., Manuel, J., Marín-Jiménez, N. G., Cordelia, S. & Karteek, A. (2017). End-to-end incremental learning. *In Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 233-248.

[23] Kumar, M. & Sharma. H. K. (2023). A GAN-based model of deepfake detection in social media. *Procedia Computer Science, 218*, 2153-2162.

[24] Shmelkov, K., Cordelia, S. & Karteek, A. (2018). How good is my GAN? *In Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213-229.

[25] Li, C., Zi, W. & Hairong, Q. (2018). Fast-converging conditional generative adversarial networks for image synthesis. *In 2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2132-2136. IEEE.

[26] Chavdarova, T. & François, F. (2018). Sgan: An alternative training of generative adversarial networks. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9407-9415.

[27] Brock, A., Donahue, J. & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096.