IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 July 2023, accepted 17 September 2023, date of publication 22 September 2023,
date of current version 29 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3317897

RESEARCH ARTICLE

3G-AN: Triple-Generative Adversarial Network Under Corse-Medium-Fine Generator Architecture

CARLOS AVILÉS-CRUZ AND GABRIEL J. CELIS-ESCUDERO
Electronics Department, Autonomous Metropolitan University-Azcapotzalco, Mexico City 02200, Mexico

Corresponding author: Carlos Avilés-Cruz (caviles@azc.uam.mx)

This work was supported by the Electronics Department of the Basic Science and Engineering Division under Grant ELE18-6.

ABSTRACT In recent years, Generative Adversarial Networks (GANs) have gained worldwide interest and have marked a breakthrough in deep learning, encouraging detailed studies in generating artificial images. A new Generative Adversarial Networks (GAN) is proposed to unveil how Human visual perception takes place, focusing on how human beings perceive images, firstly, coarse structures and then their details. The network called 3G-AN consists of three generation stages and a single Discriminator. In this paper, a novel three-branch generator is proposed, which takes into account Coarse, Medium, and Fine structure of a given image. *Coarse* RGB decomposition image provides the general structure, while *Medium* RGB stage provides general-fine structure. Finally, *Fine* RGB decomposition provides fine details of the image. The proposal is tested on MNIST, CIFAR10, and Celebrity faces databases, generating realistic images with almost no anomalies. The RGB decomposition into coarse, medium, and fine, allows to understand the composition of an image from a structural point of view. The qualitative analysis carried out in this research paper outperforms the six most competitive models existing in the literature.

INDEX TERMS GAN, artificial intelligence, deep learning, fake images.

I. INTRODUCTION

Artificial intelligence has made significant advances in the last 20 years, both in the field of machine learning and deep learning. Supervised learning has been a major focus of research and development, however, unsupervised learning is still an open problem for researchers. Recently, deep learning techniques (based on artificial neural networks) have opened an important beta in unsupervised techniques, particularly with Generative Adversarial Networks (GANs) [1]. GANs are the most common learning model for both supervised and unsupervised learning. In theory, GANs adopt a supervised learning approach to perform fake data generation. The principle of GANs, the training of two simultaneous networks, can be summarized as: the generator network denoted by **G** and the discriminator network denoted by **D**. The latter is a binary classifier that learns to classify real and generated data as genuinely as possible. Conversely, **G** confuses **D** by generating real data. These two networks are sectioned, and finally, **G** produces realistic data, and **D** specializes to predict fake data.

GANs have started to be used in different fields, mainly related to two- and three-dimensional images, however, their use has been extended to other fields such as audio and video [2], [3], [4]. Thus, GANs have been applied in image combination [5], image manipulation [6], image enhancement and inpainting [7], Content Based Image Retrieval (CBIR) [8], changing faces over time [9], face completion [10], human poses [11], face expression recognition based on generative adversarial networks [12], object detection [13], 3D image synthesis [14], texture synthesis [15], sketch synthesis [16], image-to-image transition [17]; speech and language synthesis [18], music generation [19], and applications to video [20]. Finally, there are also efforts to identify fake

The associate editor coordinating the review of this manuscript and approving it for publication was Bo Pu.

images that have been generated by GAN methods [21], [22], [23], [24] where authors present algorithms used to create deepfakes and, methods to detect them.

There are some works that have some proximity to our proposal, by namely, the so-called Triple-GAN [25], Triple-BigGAN [26], and Triple Discriminators - Equipped GAN [27]. These proposals use three elements in a GAN network, a generator, a discriminator, and a classifier. However, these works do not address the triple generation in a GAN network. Other works that use 3 or more discriminators or generators focus on a region of the image [28], [29], multi-resolution image [30], and Hyper-spectral image [31] to discriminate or generate. It should be noted that generators are implemented with exactly the same settings (kernel size, deconvolution, normalization, activation), having as inputs another image or part of the image.

In this paper, a new approach to fake image generation based on GAN scheme is presented, proposal called 3G-AN (Three-Generative Adversarial Network) consists of three generation stages and a single Discriminator. 3G-AN was designed over three-branch generator scheme, which takes into account *Coarse*, *Medium*, and *Fine* structure of a given image. Two views are presented for the *Coarse*, *Medium*, and *Fine* decomposition from a structural point of view, i.e., what elements contribute structurally to each layer. On the other hand, an analysis is presented from the combination of the RGB color components of each *Coarse*, *Medium*, and *Fine* stage. We emphasize that *Coarse* RGB decomposition image provides the general structure, while *Medium* RGB stage provides general-fine structure. Finally, *Fine* RGB decomposition provides fine details of the image. The proposal is tested on MNIST, CIFAR10, and Celebrity faces database, generating realistic faces with almost no anomalies.

The rest of the paper is organized as follows. In section II and III, the methodology and its implementation are presented. Results are shown in section IV. Finally, conclusions and future works are given in section V.

II. METHODOLOGY

The proposed methodology is based on the general structure of a GAN network (see Fig. 1), three *Generators* and a *Discriminator*. The *Discriminator* is trained with image databases (described later), the output of the *Discriminator* will be either zero or one, indicating that the image is "Real" or "Fake". In the case of the *Generator* comprising of three stages (Coarse, Medium, and Fine, and a weighted summation) starts from a vector of 100 randomly initialized values. As iteration proceeds, the values of the vector will be adapted until it starts generating images as close as possible to the training base. In the ideal case, the images of the *Generator* will be sufficiently similar to the training images (fake images). *Discriminator* will not be able to differentiate between real and fake images. It is precise at this point that the vector is taken, corresponding to a "genetic fingerprint" V_E that gives rise to a particular type of image, i.e. man, woman, smiling man, or smiling woman.

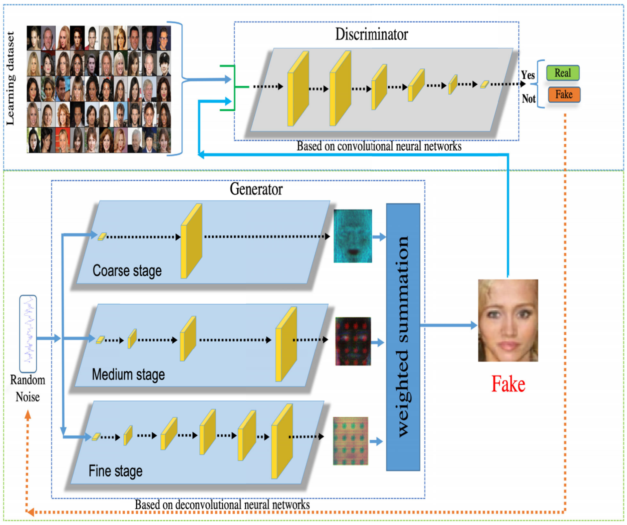


FIGURE 1. General 3G-AN structure.

Returning to the Goodfellow notation [1], 3GAN cost function is defined as

$$\underbrace{min}_{3G}\ \underbrace{max}_{D}\ V(D, 3G) \text{ for GAN:}$$

$$\begin{aligned} V(D, 3G) &= \mathbb{E}_{X\sim p_{data(x)}}[logD(x)] \\ &\quad + \mathbb{E}_{Z\sim p_z(z)}[log(1 - D(3G(z)))] \\ &= \mathbb{E}_{X\sim p_{data(x)}}[logD(x)] + \mathbb{E}_{Z\sim p_z(z)}[log(1 \\ &\quad - D(\{\alpha \cdot G^C(z) + \beta \cdot G^M(z) + \gamma \cdot G^F(z)\}))], \end{aligned} \tag{1}$$

where:

$D(x)$ = Discriminating network stage for real images.

$\mathbb{E}_x$ is the expected value over real images.

$3G(z)$ is the image generated by whole generator stage, given a noise vector z

$D(3G(z))$ is the estimator of the probability that the artificial image is real.

$\mathbb{E}_z$ is the expected value over all false images $G(z)$.

$G^C(z)$ is the image generated by coarse stage, given a noise vector z

$G^M(z)$ is the image generated by medium stage, given a noise vector z

$G^F(z)$ is the image generated by fine stage, given a noise vector z

$G^C(z)$, $G^M(z)$, and $G^F(z)$ are $\in \mathbb{R}^{W\times H\times 3}$

$W \times H$ are height and width image dimension

α, β, and γ are weighted constants $\in [0, 1]$, and $\alpha + \beta + \gamma = 1$.

Z is a random noise vector following a standard normal distribution, having *mean* = 0 and *standard deviation* = 1.

Gradient optimization of equation 1 taking over a Mini-batch of size N_B can be expressed as:

$$\nabla_{\Theta}(D, 3G)\left[\frac{1}{N_B}\sum_{X}[log\ D(x)] + \frac{1}{N_B}\sum_{Z}\left[log(1 - D(\{\alpha \cdot G^C(z) + \beta \cdot G^M(z) + \gamma \cdot G^F(z)\}))\right]\right]. \quad (2)$$

In order to get a minimum error from equation 1 and using gradient estimation according to equation 2, the minimum error can be reached when $p(X) = p(Z)$. The set of original images X should be the same as the set of generated images (fake) Z. In such condition, the fake generated images can trick the discriminator, passing themselves off as original ones.

The whole training procedure takes into account Θ as trainable parameter for the Discriminator and Generator stages.

A. PROPOSED 3G-AN STRUCTURE

Proposed 3G-AN architecture is based on 3-stage for *Generator* task, and 1-stage for *Discriminator*.

Discriminator is comprised of eleven layers; four 2D convolutions, four LeakyReLu operations, one Flattening, one Dropout and one Dense operations. Summary *Discriminator* model is presented in Fig. 2 where it can be seen that the InputLayer are images of 64 × 64 × 3 size. Image features are then convolutioned and downsampled through 2D convolution (stride = 2), and LeakyReLu as, $((64\times64) \Rightarrow (32 \times 32) \Rightarrow (16 \times 16) \Rightarrow (8 \times 8) \Rightarrow (4 \times 4) \Rightarrow flatten(1024) \Rightarrow dropout(1024) \Rightarrow dense(1))$.

Regarding *Generator* task, it is conformed through 3 stages called *Coarse*, *medium*, and *Fine*. As it can be seen in Fig. 1 (bottom part), the same input noise vector ($dimension = 100$) passes throughout three deep learning models (see Fig. 3).

1) ***Coarse task***: For the first one (right side of Fig. 3), input vector is Densely linked to a (64 × 64 × 3) 12,288 vector, after that, a batch normalization and an activation functions (LeakyReLu) are applied. After LeakyReLu function, a reshape is applied to a (64 × 64 × 3) dimension. Three more operations are applied to, a 2D convolution ($stride = 1$, filter = (5 × 5)), a batch normalization and, a LeakyReLu operation. The output of this *Coarse task* is an image of size (64 × 64 × 3) in RGB format.
2) ***Medium task:*** For the second one (medium side of Fig. 3), an input vector is Densely linked to a vector of 14,400 elements. After dense operation, a batch normalization and a LeakyReLu operations are applied. Then, reshaping is applied to (4 × 4 × 900), besides, two deconvolution2D-batchNormalization-LeakyReLu tasks are applied, output image size is (64 × 64 × 3).
3) ***Fine task:*** The third task (Fine) (left side of Fig. 3), starts linking input random vector Z to a vector of 16,384 elements (Dense operation). After Dense operation, a batch normalization and a LeakyReLu operations are applied. Then, reshaping is developed to a (4 × 4 × 1024) dimension. Besides, three deconvolution2D-batchNormalization-LeakyReLu tasks are applied, output image size is (64 × 64 × 3).

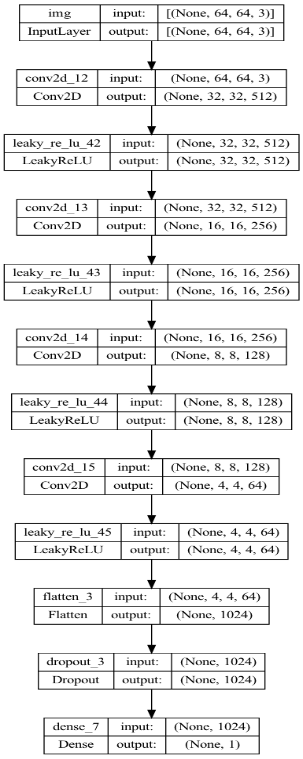

img | input: | [(None, 64, 64, 3)]
InputLayer | output: | [(None, 64, 64, 3)]
conv2d_12 | input: | (None, 64, 64, 3)
Conv2D | output: | (None, 32, 32, 512)
leaky_re_lu_42 | input: | (None, 32, 32, 512)
LeakyReLU | output: | (None, 32, 32, 512)
conv2d_13 | input: | (None, 32, 32, 512)
Conv2D | output: | (None, 16, 16, 256)
leaky_re_lu_43 | input: | (None, 16, 16, 256)
LeakyReLU | output: | (None, 16, 16, 256)
conv2d_14 | input: | (None, 16, 16, 256)
Conv2D | output: | (None, 8, 8, 128)
leaky_re_lu_44 | input: | (None, 8, 8, 128)
LeakyReLU | output: | (None, 8, 8, 128)
conv2d_15 | input: | (None, 8, 8, 128)
Conv2D | output: | (None, 4, 4, 64)
leaky_re_lu_45 | input: | (None, 4, 4, 64)
LeakyReLU | output: | (None, 4, 4, 64)
flatten_3 | input: | (None, 4, 4, 64)
Flatten | output: | (None, 1024)
dropout_3 | input: | (None, 1024)
Dropout | output: | (None, 1024)
dense_7 | input: | (None, 1024)
Dense | output: | (None, 1)

FIGURE 2. Proposed *Discriminator* of 3G-AN model.

4) ***Weighted summation:*** The output of each *Coarse*, *Medium*, and *Fine* stages are summed weighted to generate the final image. According to the theory of face human perception [32], [33], Coarse-to-fine theories of vision propose that the coarse information carried by the low spatial frequencies (LSF) of visual input guides the integration of finer, high spatial frequency (HSF) detail.

The *Weighted summation* equation is expressed in equations 3 and 4 (See Figure 3, three lines before the end).

$$Image = \Big[\alpha \cdot Coarse(R, G, B) + \beta \cdot Medium(R, G, B) + \gamma \cdot Fine(R, G, B)\Big], \quad (3)$$

Stand $\alpha = 0.5, \beta = 0.3,\ and\ \gamma = 0.2$

$$Image = \Big[(0.5) \cdot Coarse(R, G, B)$$

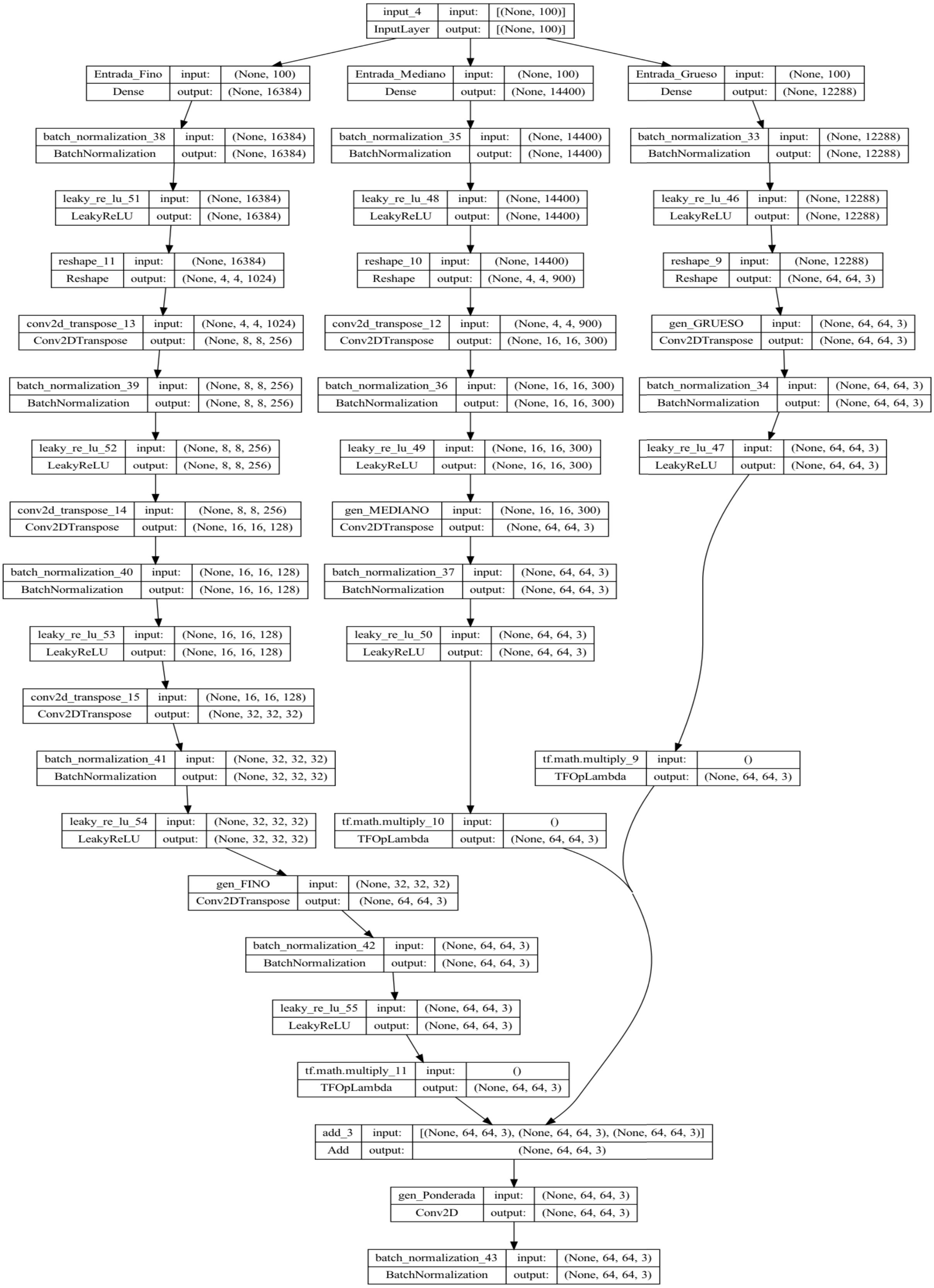

input_4 | input: | [(None, 100)]
InputLayer | output: | [(None, 100)]
Entrada_Fino | input: | (None, 100)
Dense | output: | (None, 16384)
Entrada_Mediano | input: | (None, 100)
Dense | output: | (None, 14400)
Entrada_Grueso | input: | (None, 100)
Dense | output: | (None, 12288)
batch_normalization_38 | input: | (None, 16384)
BatchNormalization | output: | (None, 16384)
batch_normalization_35 | input: | (None, 14400)
BatchNormalization | output: | (None, 14400)
batch_normalization_33 | input: | (None, 12288)
BatchNormalization | output: | (None, 12288)
leaky_re_lu_51 | input: | (None, 16384)
LeakyReLU | output: | (None, 16384)
leaky_re_lu_48 | input: | (None, 14400)
LeakyReLU | output: | (None, 14400)
leaky_re_lu_46 | input: | (None, 12288)
LeakyReLU | output: | (None, 12288)
reshape_11 | input: | (None, 16384)
Reshape | output: | (None, 4, 4, 1024)
reshape_10 | input: | (None, 14400)
Reshape | output: | (None, 4, 4, 900)
reshape_9 | input: | (None, 12288)
Reshape | output: | (None, 64, 64, 3)
conv2d_transpose_13 | input: | (None, 4, 4, 1024)
Conv2DTranspose | output: | (None, 8, 8, 256)
conv2d_transpose_12 | input: | (None, 4, 4, 900)
Conv2DTranspose | output: | (None, 16, 16, 300)
gen_GRUESO | input: | (None, 64, 64, 3)
Conv2DTranspose | output: | (None, 64, 64, 3)
batch_normalization_39 | input: | (None, 8, 8, 256)
BatchNormalization | output: | (None, 8, 8, 256)
batch_normalization_36 | input: | (None, 16, 16, 300)
BatchNormalization | output: | (None, 16, 16, 300)
batch_normalization_34 | input: | (None, 64, 64, 3)
BatchNormalization | output: | (None, 64, 64, 3)
leaky_re_lu_52 | input: | (None, 8, 8, 256)
LeakyReLU | output: | (None, 8, 8, 256)
leaky_re_lu_49 | input: | (None, 16, 16, 300)
LeakyReLU | output: | (None, 16, 16, 300)
leaky_re_lu_47 | input: | (None, 64, 64, 3)
LeakyReLU | output: | (None, 64, 64, 3)
conv2d_transpose_14 | input: | (None, 8, 8, 256)
Conv2DTranspose | output: | (None, 16, 16, 128)
gen_MEDIANO | input: | (None, 16, 16, 300)
Conv2DTranspose | output: | (None, 64, 64, 3)
batch_normalization_40 | input: | (None, 16, 16, 128)
BatchNormalization | output: | (None, 16, 16, 128)
batch_normalization_37 | input: | (None, 64, 64, 3)
BatchNormalization | output: | (None, 64, 64, 3)
leaky_re_lu_53 | input: | (None, 16, 16, 128)
LeakyReLU | output: | (None, 16, 16, 128)
leaky_re_lu_50 | input: | (None, 64, 64, 3)
LeakyReLU | output: | (None, 64, 64, 3)
conv2d_transpose_15 | input: | (None, 16, 16, 128)
Conv2DTranspose | output: | (None, 32, 32, 32)
batch_normalization_41 | input: | (None, 32, 32, 32)
BatchNormalization | output: | (None, 32, 32, 32)
tf.math.multiply_9 | input: | ()
TFOpLambda | output: | (None, 64, 64, 3)
leaky_re_lu_54 | input: | (None, 32, 32, 32)
LeakyReLU | output: | (None, 32, 32, 32)
tf.math.multiply_10 | input: | ()
TFOpLambda | output: | (None, 64, 64, 3)
gen_FINO | input: | (None, 32, 32, 32)
Conv2DTranspose | output: | (None, 64, 64, 3)
batch_normalization_42 | input: | (None, 64, 64, 3)
BatchNormalization | output: | (None, 64, 64, 3)
leaky_re_lu_55 | input: | (None, 64, 64, 3)
LeakyReLU | output: | (None, 64, 64, 3)
tf.math.multiply_11 | input: | ()
TFOpLambda | output: | (None, 64, 64, 3)
add_3 | input: | [(None, 64, 64, 3), (None, 64, 64, 3), (None, 64, 64, 3)]
Add | output: | (None, 64, 64, 3)
gen_Ponderada | input: | (None, 64, 64, 3)
Conv2D | output: | (None, 64, 64, 3)
batch_normalization_43 | input: | (None, 64, 64, 3)
BatchNormalization | output: | (None, 64, 64, 3)

FIGURE 3. **Proposed 3-*Generators* of 3G-AN model (*Coarse-Medium-Fine*).**

FIGURE 4. **Some examples of MNIST dataset.**

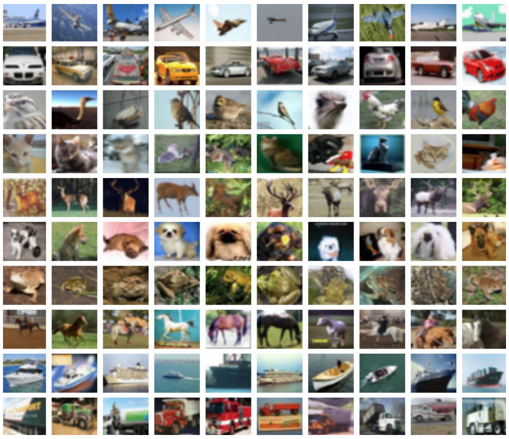
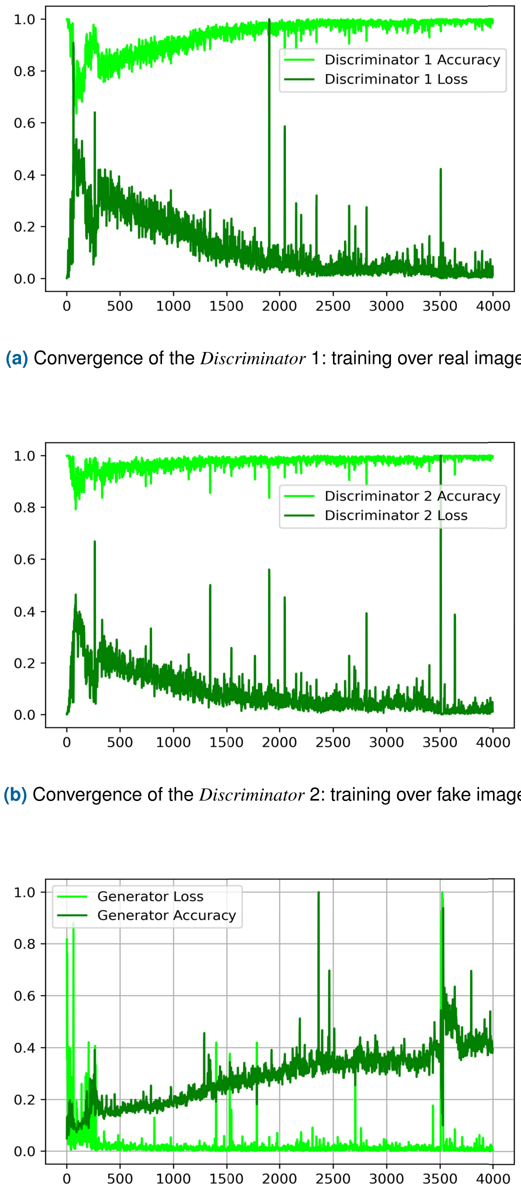


FIGURE 5. **Some examples of CIFAR-10 dataset.**

FIGURE 6. **Some examples of celebrity face dataset.**

$$
\begin{aligned}
&+ (0.3) \cdot \mathit{Medium}(R, G, B)\\
&+ (0.2) \cdot \mathit{Fine}(R, G, B)\Bigg]. \qquad (4)
\end{aligned}
$$

The *Weighted summation* layer output is an image size of (64 × 64 × 3).



(a) Convergence of the *Discriminator* 1: training over real images.

(b) Convergence of the *Discriminator* 2: training over fake images.

(c) Convergence of the *Generator*.

FIGURE 7. **Convergence plots of the proposed 3G-AN model.**

B. IMAGE DATABASE

In order to validate the proposed model, three image datasets were used: MNIST [34], Cifar-10 [35], and CelebA [36], as follows:

- ***MNIST*** [34] is a widely dataset used in the field of pattern recognition, image processing, and machine learning. MNIST contains ten hand-written digits (from 0 to 9), which is divided into 60, 000 digits for training and 10, 000 digits for testing, each dataset containing a label type. The height and width of the image are 28 × 28 pixels (see Fig. 4 as an example).

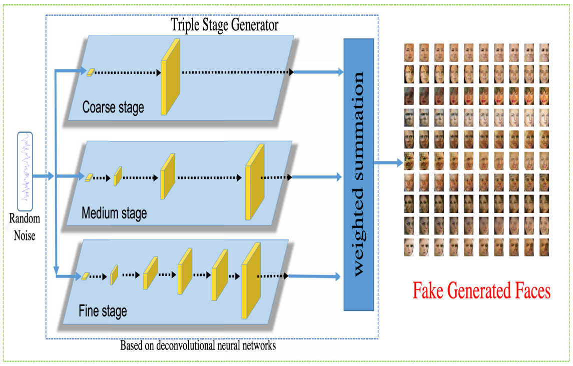


FIGURE 8. Generator of images from 3G-AN model.

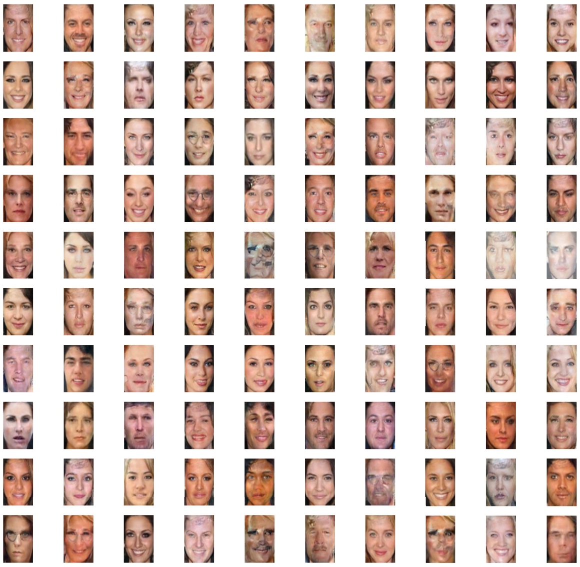


FIGURE 9. One hundred fake faces generated using 3G-AN model.

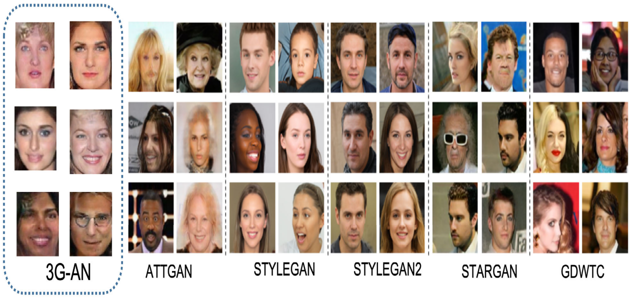
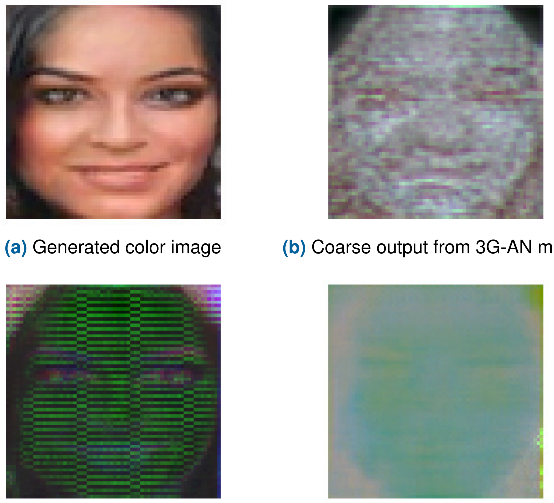


FIGURE 10. Comparation of 3G-AN model versus five competitive GANs.

- ***CIFAR-10*** [35] contains 50,000 training samples and 10,000 testing samples, where each image is in RGB format and the size of 32 × 32 pixels. The dataset contains 10 classes corresponding to 10 natural scenes *–airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck* (see Fig. 5 as an example).
- ***CelebFaces*** The sub Large-scale (celebA) dataset was used [36]. The dataset is composed of 200, 000 faces (see Fig. 6 as an example). In CelebFaces are men, women, smiling men, smiling women, blonde women, African-American women, among others.

  The *CelebFaces* image database (composed of 200K faces) was processed as follows: firstly, the face is located in each image, then, the face is cropped (using OpenFace [37]) and standardized to a size of (64 × 64) pixels, all in color, in RGB format.



(a) Generated color image

(b) Coarse output from 3G-AN model

(c) Medium output from 3G-AN model

(d) Fine output from 3G-AN model

FIGURE 11. Color RGB components for 3 output layers: Coarse (first line), Medium (second line), and Fine (third line).

C. METRICS

In order to give a measure of generated images, two metrics are applied, *mean squared error -MSE-* (see eq. 5) and *Frechet Inception Distance -FID-* (see eq. 6).

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [R(i,j) - F(i,j)]^2, \tag{5}$$

stand $R(i,j)$ real image, $F(i,j)$ fake or generated image, and $[M \times N]$ the image size.

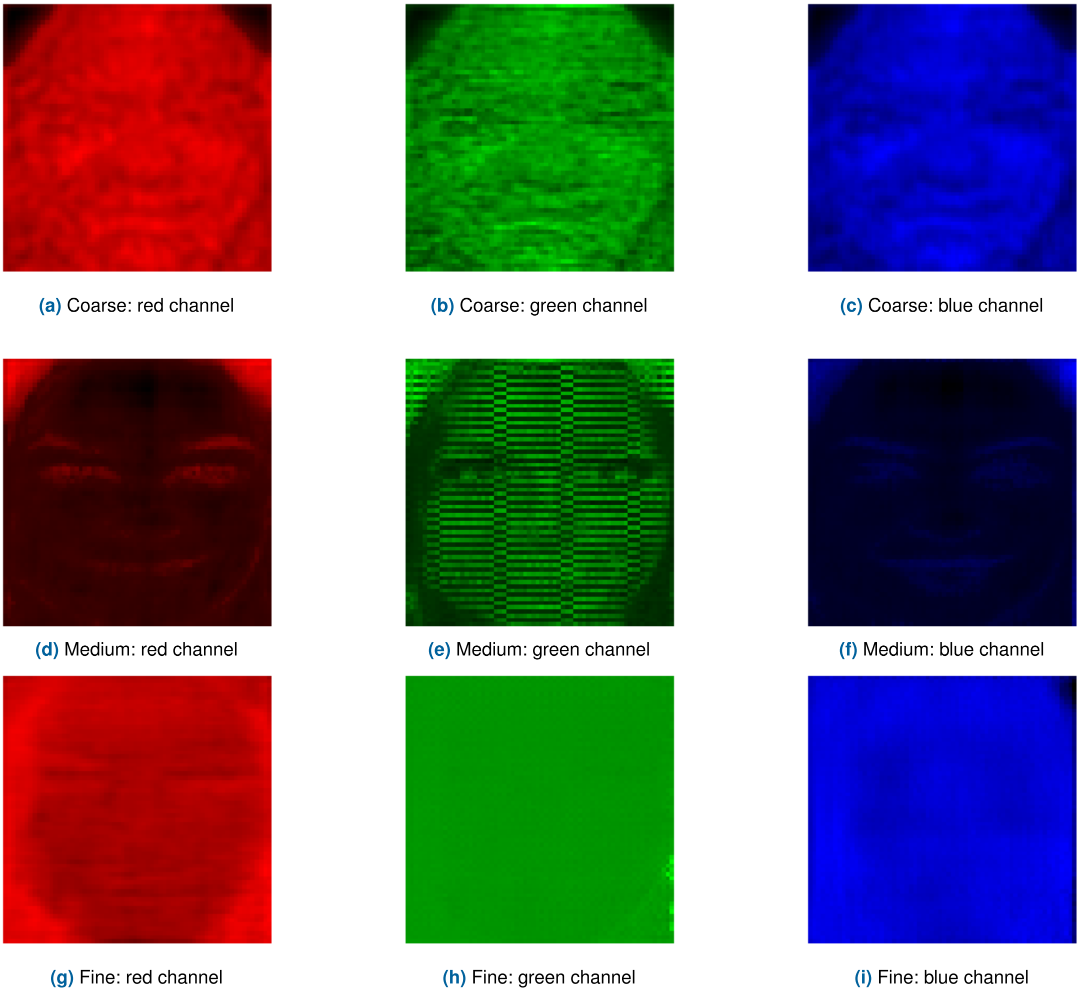
$$FID = ||\mu_R - \mu_F||^2 + Tr[\Sigma_R + \Sigma_F - 2 \cdot (\Sigma_R \cdot \Sigma_F)^{1/2}]. \tag{6}$$

stand μ_R is the mean value of the real image, μ_F is the mean value of the fake image, Σ_R is the variance/covariance matrix of the real image, Σ_F is the variance/covariance matrix of the fake image, *Tr* is the trace of a matrix (the sum of the elements along the main diagonal of the square matrix).

D. 3G-AN TRAINING PARAMETERS

The training parameters of the GAN network were as follows.

a) loss = 'binary_crossentropy',
b) optimizer = Adam(lr = 0.0002, beta = 0.5),
c) metrics = ['loss, accuracy'],
d) batch_size = 512,
e) epochs = 4, 00,
f) sample_period = 200.


(a) Coarse: red channel

(b) Coarse: green channel

(c) Coarse: blue channel

(d) Medium: red channel

(e) Medium: green channel

(f) Medium: blue channel

(g) Fine: red channel

(h) Fine: green channel

(i) Fine: blue channel

FIGURE 12. **Color RGB components for the 3 output stages from proposed 3G-AN model: Coarse (first line), Medium (second line), and Fine (thirtd line).**

The cost function of the equation (1) is optimized throughout the cross-entropy derivative.

III. SYSTEM IMPLEMENTATION

The project was implemented on an Alienware Aurora R13 Gaming Desktop 2 Gen Intel©Core^{TM}$i9-12900KF$, Windows 11 Home, NVIDIA®GeForce RTX^{TM}3090, 24 GB GDDR6X, 64 GB, DDR5, 4400 MHz, dual-channel; up to 128 GB, 512GB NVMe M.2 PCIe SSD (Boot) + 1TB 7200RPM SATA.

3G-AN was implemented in Python 3.9, using the following libraries: *Keras, 3.7.2 version* (used as a backbone), *Tensowflow* (2.1 version), *matplotlib* (3.6 version), and *numpy* (1.23 version).

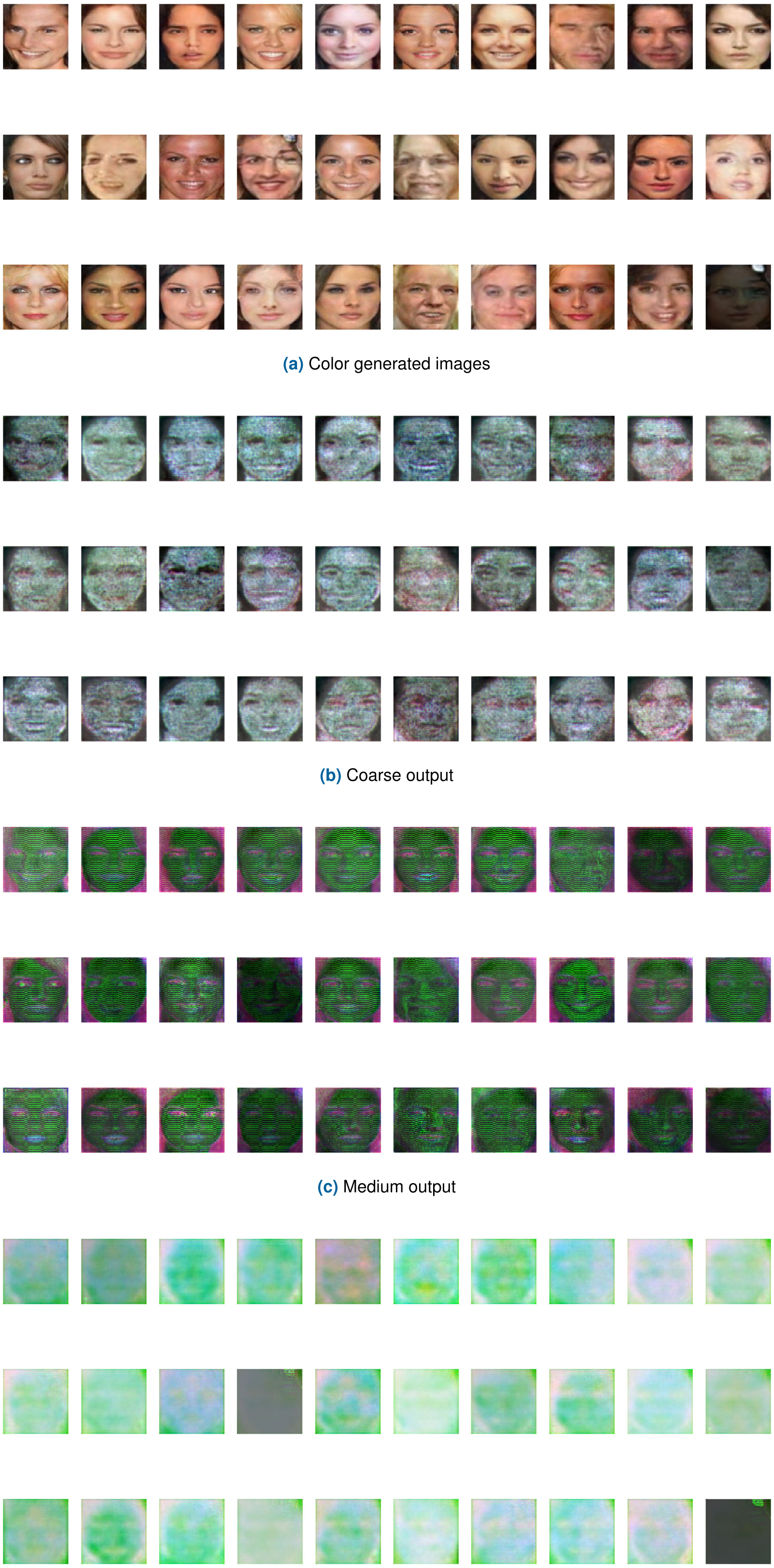
IV. RESULTS

In this section, we train and evaluate our 3G-AN model in its correct implementation and operation. After obtaining the well-trained 3G-AN model, fake image generation is performed. Then, Full model image generation is analyzed according to a *Coarse*, *medium*, and *Fine* RGB image generation. The training of the entire proposed 3G-AN network is explained as follows:

A. 3G-AN MODEL TRAINING

Once the 3G-AN model and the training database were defined, 3G-AN training was performed for the three *Generators* (Coarse, Medium, and fine) as well as for the *Discriminator*. Fig. 7 show the loss and accuracy plots. Fig. 7(a) presents accuracy and loss functions for training over real faces (*Discriminator* 1, D1), whereas, Fig. 7(b) shows metrics for training over fake images (*Discriminator* 2, D2). Finally, Fig. 7(c) provides the result figures for *Generator*.

Learning stage was done up to 4,000 epochs, in the three graphs it can be seen that the error decreases and the accuracy increases as a function of the epochs. The errors of both the *Discriminators* and the *Generator* are close to zero. Regarding the accuracy in the *Discriminators* D1 and D2, they are close to one, while for the *Generator* it is close



(a) Color generated images

(b) Coarse output

(c) Medium output

(d) Fine output

FIGURE 13. Example of 50 generated images from proposed 3G-AN model.

TABLE 1. Quantitative comparison on the three datasets, mean squared error (MSE) and frechet inception distance (FID).

Dataset	*MSE*	*FID*
CIFAR-10	0.001	1.60
MNIST	0.0001	0.66
CelebFaces	0.0015	1.20

to 0.5. In conclusion, 3G-AN Model training was developed as expected (according to the theory).

B. FAKE IMAGE GENERATION

Given a well-trained GAN model, having the weights of both the *Discrimination* and *Generation* (see Fig 8) models, one hundred images were generated from one hundred random vectors ($Z = 100 \times 100$ dimension). Fig. 9 shows one hundred fake images, as it can be seen, fake images are closely similar to original *CelebFaces*. As it can be seen in Fig. 9, there are smiling, serious, men, women, among other characteristics. Finally, the variation of light on the faces can be highlighted.

Metric evaluation

According to two metrics defined, MSE and FID (see eq. (5) and (6)), Table 1 shows measure values for three datasets, CIFAR-10, MNIST, and CelebFaces. As it can be seen, mean squared error is close to zero.

C. COMPARISON OF 3G-AN MODEL VS 5 COMPETITIVE GANs

Nowadays, different GAN-based models have been proposed to generate fake faces of human beings. Among all, the five most competitive models [24] are ATTGAN, proposed by He et al. [6], another competitive GAN model is proposed by Cho et al. GDWCT [38], Choi et al. proposed StarGAN model [39] and StarGAN-v2 model [40], Style- GAN [41], and the StyleGAN2 [42]. Fig. 10 shows six DeepFake images for each model, as you can see, proposed 3G-AN model (left side) generates only the face, showing red lips, colorful eyes, and different types and colors of hair can be seen. In 3G-AN model, fake generating images results are realistic faces with almost no anomalies.

Qualitative Analysis

Table 2 illustrates a comparison of our proposal versus the most competitive methods reported in the literature. Table 2 gives FID measure for two datasets, *CIFAR10* and *CelebFaces*. Our proposal outperforms 6 most competitive models, reaching 1.60 for *CIFAR10*, and 1.20 for *CelebFaces*. Since MNIST is not a complicated dataset to generate, the most competitive models give the same FID value.

D. RGB ANALYSIS OVER COARSE-MEDIUM-FINE FACE DECOMPOSITION

The contribution of this article is the understanding of the conformation of a face in its Coarse, medium, and Fine components, firstly, the case of a single face in its RGB components.

TABLE 2. Quantitative comparison of the two datasets using frechet inception distance (FID).

CIFAR10	
Model	*FID*
Our proposal	1.60
EDM-G++ [43]	1.77
EDM [44]	1.97
CLD-SGM [45]	2.25
NDM [46]	2.28
iDDPM [47]	2.90
VDM [48]	7.41
CelebFaces	
Our proposal	1.20
Soft Truncation-G++ [43]	1.34
Diffusion StyleGAN2 [49]	1.69
NDM [46]	1.75
Soft Diffusion [50]	1.85
Soft Truncation [51]	1.90
DDPM++ [52]	1.32

An RGB woman fake face decomposition is shown in Fig. 11. A fake *CelebFaces* type is generated from a gaussian noise vector (Z) as it is shown in Fig. 11(a). The decomposition of RGB *Coarse* is shown in Fig. 11(b), RGB *Medium* image is shown in Fig. 11(b), and RGB *Fine* image is shown in Fig. 11(d).

The *Coarse* component is expected to provide the general structure of the face, while the *Medium* and *Fine* components provide the structure and details of the face.

Another example is presented in Fig. 13 for 50 images. As it can be seen, Fig. 13 (b) shows *Coarse* components where general face structure can be remarked. Fig. 13 (c) General characteristics of *Medium* components with other details are presented in Fig. 13(c). Finally, Fig. 13 (d) contents *Fine* components, minimal face complementarity can be observed.

V. CONCLUSION AND FURTHER WORK

In this paper, a new GAN model, called 3G-AN has been proposed that consists of three generation stages and a single discriminator. The proposal has worked properly at each of its stages, it was tested on MNIST, CIFAR10, and Celebrity faces databases, generating fake images realistic with good quality and almost no anomalies, superseding the five most competitive models reported in the literature.

A gain of decomposing a face through a novel three-branch generator (Coarse, Medium, and Fine) from a given face is *Coarse* RGB decomposition face provides the general structure, while *Medium* RGB stage provides general-fine structure. Finally, *Fine* RGB decomposition provides fine details of the image.

The RGB decomposition Coarse, medium, and Fine, allows to understand the composition of an image from a structural point of view.

Our proposal outperforms 6 most competitive models for CIFAR10 and CelebFaces datasets.

As a future work, the human validity of the Coarse, Medium, and Fine decompositions will be analyzed together with neuropsychologists in vision.

REFERENCES

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[2] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–49, Nov. 2022.

[3] P. Shamsolmoali, M. Zareapoor, E. Granger, H. Zhou, R. Wang, M. E. Celebi, and J. Yang, "Image synthesis with adversarial networks: A comprehensive survey and case studies," *Inf. Fusion*, vol. 72, pp. 126–146, Aug. 2021.

[4] J. Ma, P. Saxena, and S. I. Ahamed, "A comprehensive qualitative and quantitative review of current research in GANs," in *Proc. IEEE 45th Annu. Comput., Softw., Appl. Conf. (COMPSAC)*, Jul. 2021, pp. 1675–1682.

[5] B.-C. Chen and A. Kae, "Toward realistic image compositing with adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8407–8416.

[6] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen, "AttGAN: Facial attribute editing by only changing what you want," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5464–5478, Nov. 2019.

[7] B. Dolhansky and C. C. Ferrer, "Eye in-painting with exemplar generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7902–7911.

[8] S. R. Dubey, "A decade survey of content based image retrieval using deep learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2687–2704, May 2022.

[9] Y. Liu, Q. Li, and Z. Sun, "Attribute-aware face aging with wavelet-based generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11869–11878.

[10] R. Ghanem and M. Loey, "Face completion using generative adversarial network with pretrained face landmark generator," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 2, pp. 295–305, Apr. 2021.

[11] A. Siarohin, S. Lathuilière, E. Sangineto, and N. Sebe, "Appearance and pose-conditioned human image generation using deformable GANs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1156–1171, Apr. 2021.

[12] L. Lu, "Multi-angle face expression recognition based on generative adversarial networks," *Comput. Intell.*, vol. 38, no. 1, pp. 20–37, Feb. 2022.

[13] Z. Ruiqiang, Z. Yu, and J. Xin, "Optimization of small object detection based on generative adversarial networks," in *Proc. E3S Web Conf.*, vol. 245, 2021, Art. no. 03062.

[14] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 82–90.

[15] B. U., J. N., and V. R., "Learning texture manifolds with the periodic spatial GAN," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 1, 2017, pp. 722–730.

[16] J. Yu, X. Xu, F. Gao, S. Shi, M. Wang, D. Tao, and Q. Huang, "Towards realistic face photo-sketch synthesis via composition-aided GANs," 2020, *arXiv:1712.00899*.

[17] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.

[18] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," 2018, *arXiv:1705.11001*.

[19] P. L. Tomaz Neves, J. Fornari, and J. Batista Florindo, "Self-attention generative adversarial networks applied to conditional music generation," *Multimedia Tools Appl.*, vol. 81, no. 17, pp. 24419–24430, Jul. 2022.

[20] C. Madarasingha, S. R. Muramudalige, G. Jourjon, A. Jayasumana, and K. Thilakarathna, "VideoTrain++: GAN-based adaptive framework for synthetic video traffic generation," *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108785.

[21] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey," *Comput. Vis. Image Understand.*, vol. 223, Oct. 2022, Art. no. 103525.

[22] H. S. Shad, M. M. Rizvee, N. T. Roza, S. M. A. Hoq, M. M. Khan, A. Singh, A. Zaguia, S. Bourouis, and S. K. Gupta, "Comparative analysis of deepfake image detection method using convolutional neural network," *Comput. Intell. Neurosci.*, vol. 2021, Jan. 2021, Art. no. 3111676.

[23] G. Lee and M. Kim, "Deepfake detection using the rate of change between frames based on computer vision," *Sensors*, vol. 21, no. 21, p. 7367, Nov. 2021.

[24] L. Guarnera, O. Giudice, F. Guarnera, A. Ortis, G. Puglisi, A. Paratore, L. M. Q. Bui, M. Fontani, D. A. Coccomini, R. Caldelli, F. Falchi, C. Gennaro, N. Messina, G. Amato, G. Perelli, S. Concas, C. Cuccu, G. Orrú, G. L. Marcialis, and S. Battiato, "The face deepfake detection challenge," *J. Imag.*, vol. 8, no. 10, p. 263, 2022.

[25] C. Li, K. Xu, J. Zhu, J. Liu, and B. Zhang, "Triple generative adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9629–9640, Dec. 2022.

[26] A. Gangwar, V. González-Castro, E. Alegre, and E. Fidalgo, "Triple-BigGAN: Semi-supervised generative adversarial networks for image synthesis and classification on sexual facial expression recognition," *Neurocomputing*, vol. 528, pp. 200–216, Apr. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223000346

[27] J. Zhang, J. Shi, M. Li, M. Guo, and Z. Pan, "Triple discriminators–equipped GAN for denoising of Chinese calligraphic tablet images," *Multimedia Tools Appl.*, vol. 81, no. 29, pp. 42691–42711, Dec. 2022.

[28] Y. Xu, M. Pagnucco, and Y. Song, "An edge guided coarse-to-fine generative network for image outpainting," *Neurocomputing*, vol. 541, Jul. 2023, Art. no. 126254.

[29] L. Yang, J. Wei, Z. Zuo, and S. Zhou, "MAC-GAN: A community road generation model combining building footprints and pedestrian trajectories," *ISPRS Int. J. Geo-Inf.*, vol. 12, no. 5, p. 181, Apr. 2023.

[30] C. Su, X. Wang, R. Liu, Z. Guo, S. Sang, S. Yu, and H. Zhang, "Fault diagnosis method based on triple generative adversarial nets for imbalanced data," *Meas. Sci. Technol.*, vol. 34, no. 3, Mar. 2023, Art. no. 035007.

[31] A. Qin, Z. Tan, R. Wang, Y. Sun, F. Yang, Y. Zhao, and C. Gao, "Distance constraints-based generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5511416.

[32] J. C. Peters, R. Goebel, and V. Goffaux, "From coarse to fine: Interactive feature processing precedes local feature analysis in human face perception," *Biol. Psychol.*, vol. 138, pp. 1–10, Oct. 2018.

[33] K. Petras, S. ten Oever, C. Jacobs, and V. Goffaux, "Coarse-to-fine information integration in human vision," *NeuroImage*, vol. 186, pp. 103–112, Feb. 2019.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[35] A. Krizhevsky, "Convolutional deep belief networks on CIFAR-10," Univ. Toronto, Tech. Rep., May 2012. [Online]. Available: https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf

[36] S. Yang, P. Luo, C. C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," 2015, *arXiv:1509.06451*.

[37] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, Oct. 2016.

[38] W. Cho, S. Choi, D. Keetae Park, I. Shin, and J. Choo, "Image-to-image translation via group-wise deep whitening-and-coloring transformation," 2018, *arXiv:1812.09912*.

[39] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," 2017, *arXiv:1711.09020*.

[40] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "StarGAN V2: Diverse image synthesis for multiple domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8188–8197.

[41] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2019, *arXiv:1812.04948*.

[42] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," 2020, *arXiv:1912.04958*.

[43] D. Kim, Y. Kim, S. J. Kwon, W. Kang, and I.-C. Moon, "Refining generative process with discriminator guidance in score-based diffusion models," 2022, *arXiv:2211.17091*.

[44] T. Karras, M. Aittala, T. Aila, and S. Laine, "Elucidating the design space of diffusion-based generative models," 2022, *arXiv:2206.00364*.

[45] T. Dockhorn, A. Vahdat, and K. Kreis, "Score-based generative modeling with critically-damped Langevin diffusion," 2021, *arXiv:2112.07068*.

[46] D. Kim, B. Na, S. J. Kwon, D. Lee, W. Kang, and I.-C. Moon, "Maximum likelihood training of implicit nonlinear diffusion models," 2022, *arXiv:2205.13699*.

[47] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021, *arXiv:2102.09672*.

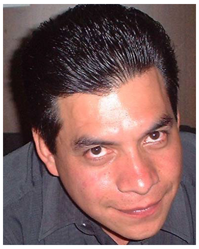
[48] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "Variational diffusion models," in *Proc. 35th Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2023, pp. 1–12.

[49] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-GAN: Training GANs with diffusion," 2022, *arXiv:2206.02262*.

[50] G. Daras, M. Delbracio, H. Talebi, A. G. Dimakis, and P. Milanfar, "Soft diffusion: Score matching for general corruptions," 2022, *arXiv:2209.05442*.

[51] D. Kim, S. Shin, K. Song, W. Kang, and I.-C. Moon, "Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation," 2021, *arXiv:2106.05527*.

[52] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2020, *arXiv:2011.13456*.



CARLOS AVILÉS-CRUZ received the degree in electronics engineering specialized in digital systems from Universidad Autónoma Metropolitana, Mexico, in 1991, and the master's and Ph.D. degrees with a specialty in digital signal processing, image and voice processing from the National Polytechnic Institute of Grenoble, France, in 1993 and 1997, respectively. He has published more than 60 peer-reviewed articles and contributed to many national and international conference proceedings, as well as coauthored two books. His research interests include computer deep learning, convolutional neural networks, computer vision, digital image processing, digital signal processing, pattern recognition, and higher order statistics. He is a member of the National System of Researchers in Mexico.

GABRIEL J. CELIS-ESCUDERO received the degree in electronics engineering specialized in digital systems from Universidad Autónoma Metropolitana, Mexico, in 2022, and the degree in electrical engineering specialized in mechatronics from the National Polytechnic Institute of Mexico. He has published three international articles. His research interests include deep learning, computer vision, digital image processing, digital signal processing, and pattern recognition.