GENERATIVE AI

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Candidate Name

(Dhanushree Y | AP21110011491)

(Vaishnavi PS | AP21110011543)

(Hima Teja G | AP21110011548)

(Abhi Shasank S | AP21110011499)

(Mahitha V | AP21110011552)

Under the Guidance of

(Prof. Radha Guha)

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

Nov, 2023

Certificate

Date: 3-Dec-23

This is to certify that the work present in this Project entitled "**GENERATIVE AI**" has been carried out by **Hima Teja** under **Prof. Radha Guha**'s supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **School of Engineering and Sciences**.

Supervisor

(Signature)

Prof. Radha Guha

Professor,

SRM University AP.

Acknowledgements

We would like to express my sincere gratitude to the people and organizations for their support and guidance in the completion of my Undergraduate Research Opportunities Project (UROP).

We are also deeply indebted to my faculty mentor, Prof. Radha Guha, for her constant encouragement, feedback, and supervision throughout the project. She helped me to define the scope, objectives, and methodology of the project and provided me with the necessary resources and tools. She was always available for consultation and guidance whenever we faced any difficulties or challenges in the project.

We would like to thank all my friends who supported me during this project. They motivated me to overcome the obstacles and challenges that we encountered along the way. They also provided me with constructive criticism and suggestions to improve my work. We are grateful to all of them for their invaluable contribution to my project.

Lastly, we are extremely grateful to SRM University, Andhra Pradesh, for providing me with a platform to grow both academically and personally. The support and kindness from all the faculty members in the CSE Department have played a pivotal role in the success of my project.

Table of Contents

Abstract

In recent times supervised learning with convolutional networks (CNNs) has explored more than unsupervised learning. In this work we tried to work on deep convolutional generative adversarial networks (DCGANs) which is a class of CNN.

Image Generation is a task of generating new images using existing or similar images. GAN' s (Generative Adversarial Network) are type of machine learning model that is well suited for image generation. DCGAN (Deep Convolutional Generative Adversarial Networks) uses the image generated by GAN and gives feedback regarding the quality of fake images to GAN. DCGAN mainly consists of two networks namely Generator and Discriminator.

These two networks are trained together in adversarial manner such that the generator becomes better creating more realistic fake images that fool the discriminator. In this paper we generated fake images using DCGAN by giving an input dataset.

DCGAN have a wide rage of applications including like generating new images from anything you imagine, augmenting datasets which can be helpful for image recognition systems, medical imaging where we can generate realistic images looking similar to scans which can help doctors, fashion designing which can help fashion designing to generate new designs.

Researchers are trying to explore potential of DCGAN beyond image generation such as Image editing and manipulation which enhance details of an image, add filters, erase unnecessary noise and do generative fills. They are also trying to explore on video generation, drug discovery and natural language processing.

Abbreviations

GAN	Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
CNN	Convolutional Neural Network

List of Tables

List of Figures

List of Equations

1. Introduction

Generative AI is a part of artificial intelligence that focuses on creating new content, such as images, music, and text by getting trained. It accomplishes this by first identifying patterns and relationships in the available data, after which it creates new examples that are comparable yet distinct.

Generative AI and DCGANs (Deep Convolutional Generative Adversarial Networks) are closely related concepts that have revolutionized the field of image generation. Generative AI encompasses a broad range of techniques that enable machines to produce new, original content, such as images, music, and text. DCGANs, a specific type of generative adversarial network (GAN), have emerged as a powerful tool for generating realistic and high-quality images.

A generator and a discriminator are two neural networks that are competed against one another in DCGANs. Making fake images that resemble the actual photos in the training dataset is the generator's responsibility and it’s the discriminator’s job to differentiate between real and fake images.

1.1 Generator

The Generator produces new images. Through an iterative process, Generator tries to improve its ability to create realistic data. Generator does not have access to real images.

1.2 Discriminator

The discriminator in a DCGAN is a neural network that consists of convolutional layers followed by a fully connected layer and a sigmoid activation function. The convolutional layers extract image features and fully connected layer sums up these features to a single value and sigmoid activation functions returns 0 or 1 depending on input image.

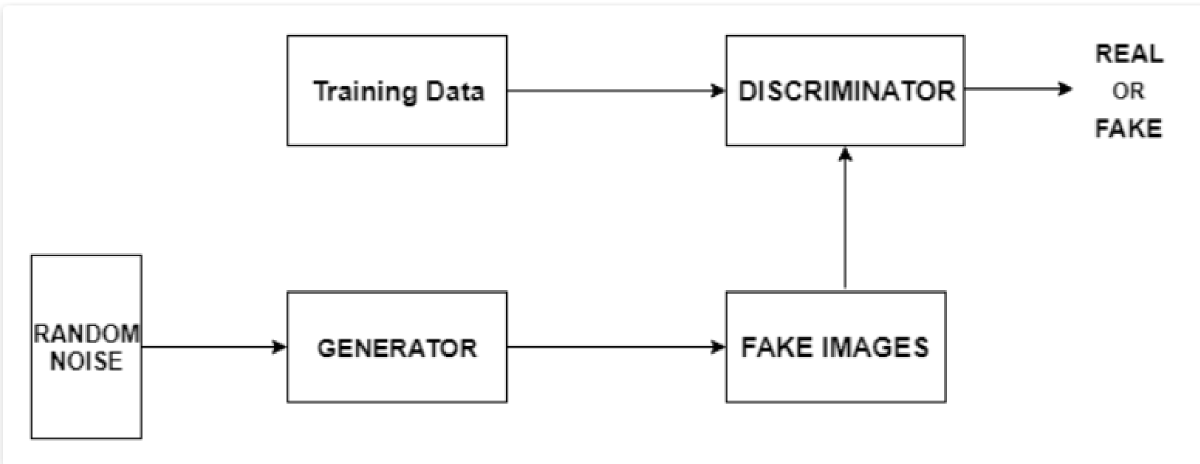

Figure 1. Structure of DCGAN

The generator is compelled to acquire more realistic image-making skills as a result of this adversarial process. The discriminator must improve in parallel with the generator. The learning process is driven by the competition between the two networks, which ultimately results in the generation of high-quality images.

DCGANs have excelled in generating realistic and diverse human faces. Their ability to capture the intricacies of facial features, expressions, and textures has made them a valuable tool for creating synthetic faces for various applications, including: Face editing and Stylized Face Generation.

2. Methodology

The methodology of this project encompasses several stages, starting from taking a sample dataset to the final stage which is generating more realistic images. This process involves data preprocessing, feature extraction, model training, and evaluation. Each stage is critical to ensure the accuracy and effectiveness of the image generation.

2.1 Description of Dataset

It Contains Celeb faces which are used to train the Generator for generating fake images. It also contains 4 CSV files .The first file contains landmarks of every image id like nose_x, leftmouth_x…., righteye_y etc. The second file contains Bounding box co-ordinates for which every image has x and y co-ordinates, width and height. Third file contains Partitions for which each image has partition number 0/1/2 which were used for training, Validation and testing respectively. The last file contains Attributes of celeba like images having features like wearing necklace, black hair, sideburns, smiling etc. which take values -1(if not present) /1 (if present).

2.2 Libraries, Activation and Loss functions

It uses many python classifiers like NumPy which is used in image classification, pandas which is used for preprocessing of data, identifying patterns; matplotlib to visualize landmarks, bounding boxes and some other libraries according to the requirement.

We used many activation functions like Leaky ReLu for every convolutional layer which improves image quality, stabilize training prevents dying state where all activations are zero, we use Adam Optimizer to update weights through back propagation Algorithm. Loss function is calculated by using binary cross entropy.

2.3 Working

Generator produces images whereas Discriminator is a binary classifier that talks about whether the image is Fake (0) or Real (1). Discriminators give feedback and both are improved after some Epochs. We Create the discriminator such that it maps a 64x64 image to a binary classification score. After receiving all the images, fake and real, the Discriminator returns probabilities, a number in the range of 0 and 1, 1 representing a prediction of authenticity and 0 representing fake. After Discriminator predicts 1/0, we have a loss function we accordingly update weights to Generator for improving the quality of fake images. We use Adam Optimizer to update weights through back propagation Algorithm. We construct a discriminator such that real/fake image of size 64*64*3 after applying first convolutional layer it reduces to 32*32*64, where 64 is number of filters and then by applying second convolutional layer it reduces to 16*16*128 and then to 8*8*128 and then we flatten our image for classifying and then we have dense layer in which we perform classification using sigmoid function. For every convolution layer we use Leaky ReLU activation function. Generator tries to fool the discriminator, initially generator starts with random noise which is a one-dimensional vector of size 100*1, we then reshape random noise into 8*8*128(a low-resolution image) then we send it to conv2D Transpose for three times to up sample our image after every layer. Finally, we will send the up sampled image value to conv2D layer, in which we will get 64*64*3 image size. The size of the generated image from generator should be the same as size of input image. In training step, we will concatenate fake image produced by generator and real image and we give it to discriminator which will calculate loss function and optimizer update weights for generator. Loss function is calculated by using binary cross entropy.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 64)	3136
leaky_re_lu (LeakyReLU)	(None, 32, 32, 64)	0
conv2d_1 (Conv2D)	(None, 16, 16, 128)	131200
leaky_re_lu_1 (LeakyReLU)	(None, 16, 16, 128)	0
conv2d_2 (Conv2D)	(None, 8, 8, 128)	262272
leaky_re_lu_2 (LeakyReLU)	(None, 8, 8, 128)	0
flatten (Flatten)	(None, 8192)	0
dropout (Dropout)	(None, 8192)	0
dense (Dense)	(None, 1)	8193

Table 1. Discriminator model

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 8192)	827392
reshape (Reshape)	(None, 8, 8, 128)	0
conv2d_transpose (Conv2DTranspose)	(None, 16, 16, 128)	262272
leaky_re_lu_3 (LeakyReLU)	(None, 16, 16, 128)	0
conv2d_transpose_1 (Conv2DTranspose)	(None, 32, 32, 256)	524544
leaky_re_lu_4 (LeakyReLU)	(None, 32, 32, 256)	0
conv2d_transpose_2 (Conv2DTranspose)	(None, 64, 64, 512)	2097664
leaky_re_lu_5 (LeakyReLU)	(None, 64, 64, 512)	0
conv2d_3 (Conv2D)	(None, 64, 64, 3)	38403

Table 2. Generator model

The above two tables depict output shapes in every layer with corresponding output shapes.

3. Discussion

image_id	lefteye_x	lefteye_y	righteye_x	righteye_y	leftmouth_x	leftmouth_y	rightmouth_x	Rightmouth_y
000001.jpg	69	109	106	113	73	152	108	154
000002.jpg	69	110	107	112	70	151	108	153

Table 3. Landmarks table

It contains co-ordinates of images like left_eye, right_eye and many attributes x and y co-ordinates respectively.

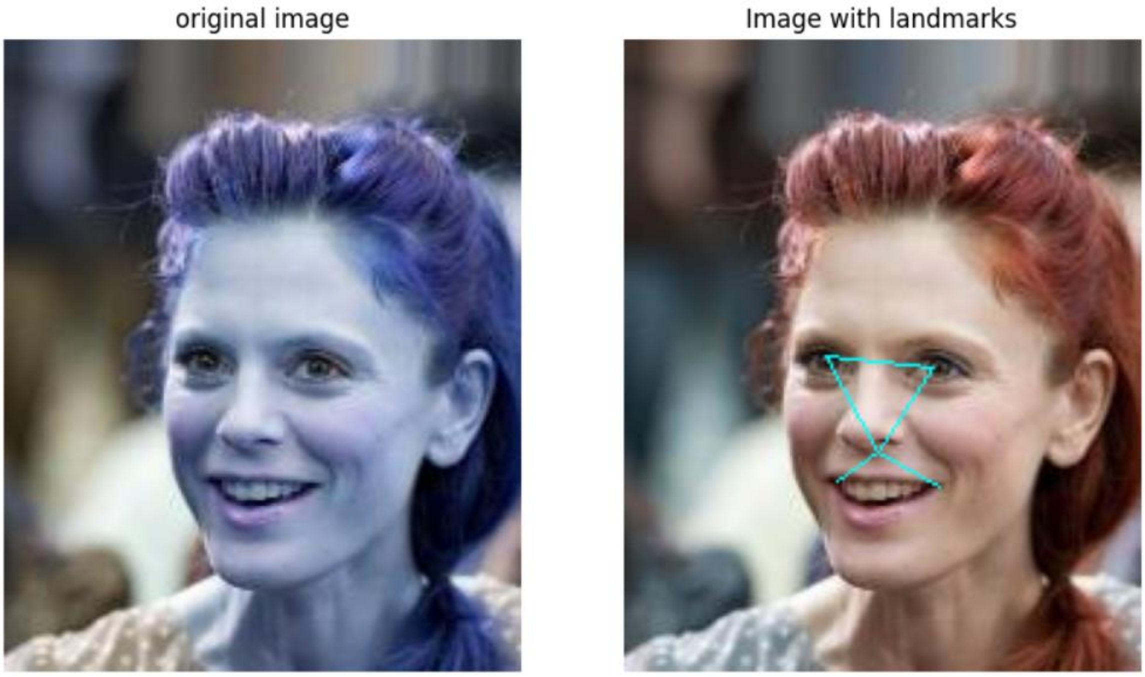

Figure 2. Landmarks plotted on image.

To get a better understanding of the landmarks of an image we plot these landmarks which is generally eyes, nose etc. in our image and in general can be edges, corners and textures.

image_id	x_1	y_1	width	height
000001.jpg	95	71	226	313
000002.jpg	72	94	221	306

Table 4. Boundary Box Co-ordinates

This table contains co-ordinates of boundary box with x, y co-ordinates and height and width of image.

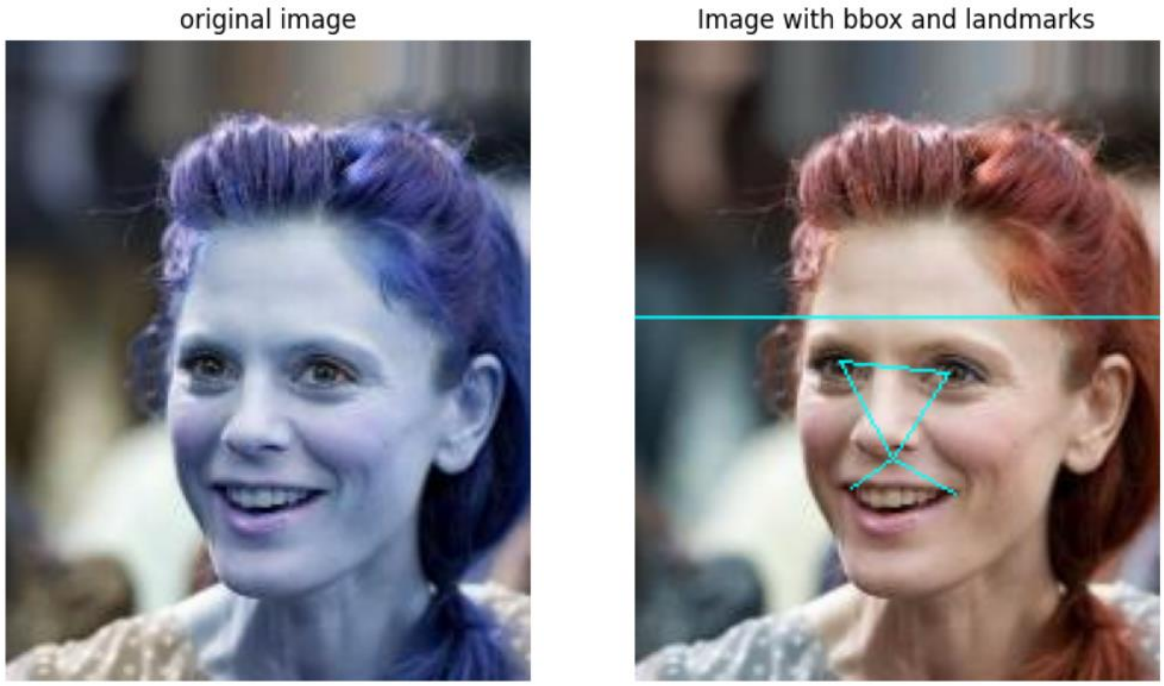

Figure 3. Boundary Box Plotted on image.

To get a better understanding of boundaries for images in the dataset, we plot these boundary boxes.

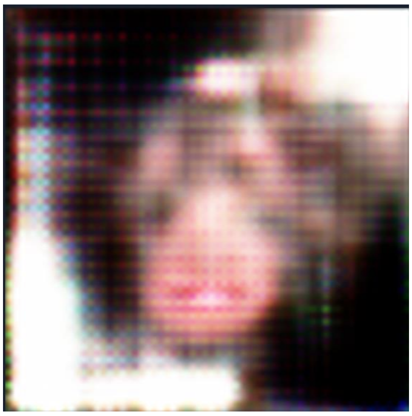

Figure 4. Generated Fake Image sample

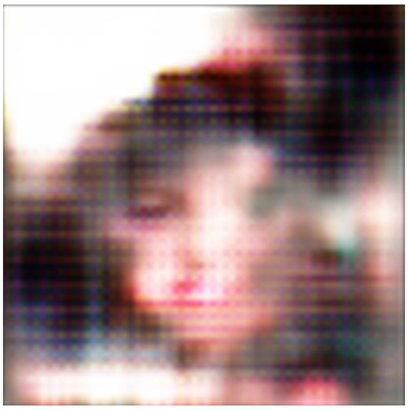

Figure 5. Generated Fake Image sample

Figure 6. Generated Fake Image sample (at 8^{th} Epoch)

These were the sample images we tried to generate using DCGAN. As we increase the epoch count the quality of image generated will increase and look even more realistic.

Equations used:

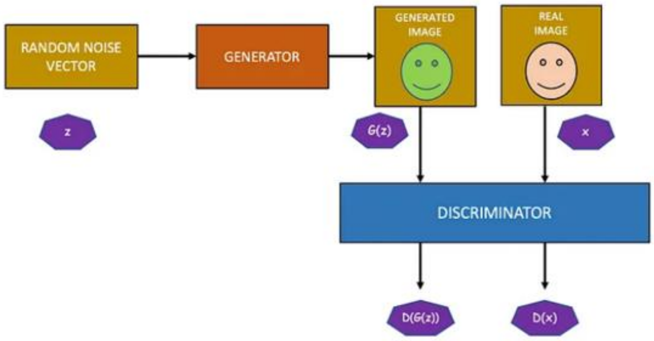

Figure 7. Description for Equations

Loss equation for Discriminator: **L(Discriminator) = max[log(D(x)) +log(1-D(G(z))]**

Loss equation for Generator: **L(Generator) = min[log(1-D(G(z)]**

Where, D(x) is output of discriminator for real input image.

z is input noise vector provided by generator.

G(z) is output of generator.

D(G(z)) is output of discriminator with generated data as input.

Performance and Loss Evaluation:

In the context of Deep Convolutional Generative Adversarial Networks (DCGANS), evaluating the performance of the generator and discriminator Involves assessing both the loss values and the generated samples' quality. The two main components to consider are the generator loss and the discriminator loss.

Generator Loss:

The generator loss represents performance of generator. It is a measure showing the difference between real and generated images. The generator tries to minimize its loss. Common loss functions for generator are cross entropy or mean squared error. The lower the loss of the generator, the better is its performance.

Discriminator loss:

Discriminator loss measures the performance of discriminator showing how well it can distinguish between real and fake images. Discriminator also tries to minimize its loss and loss function used for discriminator is binary cross entropy.

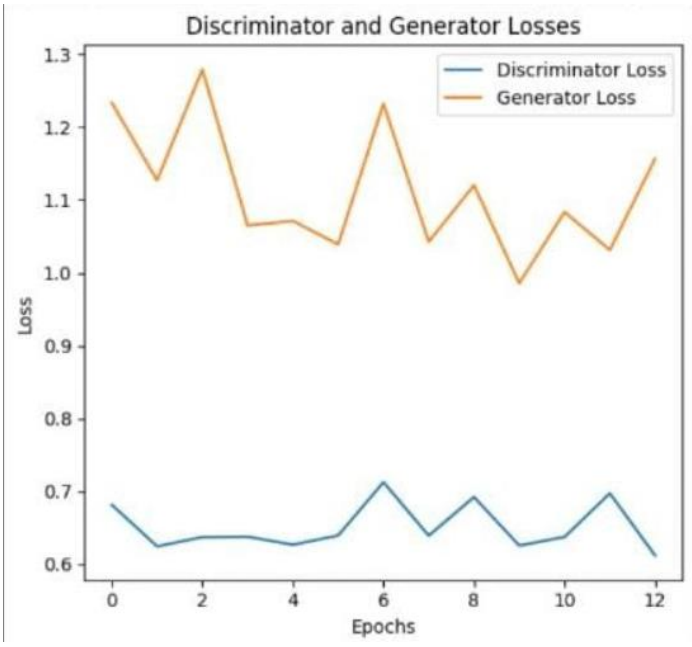

Figure 8. Losses of Generator and Discriminator

As the Generator loss is more at the starting epoch after getting feedback from discriminator it's loss is being decreased similarly to generator.

For these GAN's if they have low loss then it have more accuracy as we increase the training time or count of epochs the loss of discriminator and generator decreases as a result accuracy increases.

4. Concluding Remarks

In conclusion, our research delved into the realm of unsupervised learning, specifically focusing on the powerful capabilities of Deep Convolutional Generative Adversarial Networks (DCGANs) for image generation. Supervised learning with Convolutional Neural Networks (CNNs) has seen widespread exploration, but our emphasis on unsupervised learning aimed to harness the potential of DCGANs in generating diverse and realistic images.

The core architecture of DCGANs involves two networks – the Generator and the Discriminator – trained in an adversarial manner. This synergy enables the Generator to continually improve its ability to create images that convincingly deceive the Discriminator. Our experiments involved training DCGANs on input datasets, resulting in the generation of synthetic images with applications across various domains.

One of the primary utilities of DCGANs lies in image generation, offering the capability to produce novel images based on given datasets. This has far-reaching implications, including data augmentation for image recognition systems, realistic medical imaging for diagnostic assistance, and innovative designs for the field of fashion.

Beyond image generation, researchers are exploring the extended potential of DCGANs. Applications such as image editing and manipulation showcase the ability to enhance details, add filters, remove noise, and perform generative fills, expanding the role of DCGANs in creative processes.

It is evident that DCGANs hold promise as a versatile tool in the machine learning landscape. The exploration of their applications continues to unveil new possibilities, making DCGANs a valuable asset for researchers and practitioners seeking innovative solutions in various fields. The success of our experiments underscores the significance of unsupervised learning and the potential of DCGANs to contribute to the evolution of artificial intelligence.

5. Future Work

Deep Convolutional Generative Adversarial Networks (DCGANs) for the creation of phony images have great prospects as long as academics keep solving problems and taking fresh approaches. Here are few possible directions for further research:

Improving Translation Across Domains:

Subsequent research endeavors may concentrate on enhancing DCGANs' proficiency in transposing images across various domains. This entails translating semantic material in addition to transferring style. Sturdy techniques for producing visuals that smoothly change from one domain to another, like summer to winter or day to night, would be useful in computer vision and entertainment.

CNN Model:

We also decided to create a CNN Model which we made trained in such a way that it tells whether a given image is fake or real so that we may use it for identifying the image is Original or fake.

Future research endeavors seek to enhance the caliber and variety of produced images while tackling wider issues such as impartiality, interaction, and resilience, so rendering DCGANs more adaptable and useful across an extensive array of fields and uses.

References

1. C. Kaushik and S. N. Singh, 'Generate Artificial Human Faces with Deep Convolutional Generative Adversarial Network (DCGAN) Machine Learning Model', in International Conference on Advances and Applications of Artificial Intelligence and Machine Learning, 2022, pp. 45–55.

2. W. Liu, Y. Gu, and K. Zhang, 'Face Generation using DCGAN for Low Computing Resources', in 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), 2021, pp. 377–382.

3. D. M. Shariff, H. Abhishek, D. Akash, and Others, 'Artificial (or) fake human face generator using generative adversarial network (gan) machine learning model', in 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2021, pp. 1–5.

4. N. Nekamiche, C. Zakaria, S. Bouchareb, and K. Smaïli, 'A deep convolution generative adversarial network for the production of images of human faces', in Asian conference on intelligent information and database systems, 2022, pp. 313–326.