A Deep Convolution Generative Adversarial Network for the Production of Images of Human Faces

Noha Nekamiche, Chahnaz Zakaria, Sarra Bouchareb, Kamel Smaïli

A Deep Convolution Generative Adversarial Network for the Production of Images of Human Faces

Noha Nekamiche[1], Chahnez Zakaria[1], Sarra Bouchareb[2], and Kamel Smaïli[3]

[1] École nationale Supérieure d'Informatique, Algiers, Algeria
{hn_nekamiche, c_zakaria}@esi.dz
[2] Laboratoire LIMPAF Bouira university, Algeria
sa.bouchareb@univ-bouira.dz
[3] Loria, Campus Scientifique, Vandoeuvre Lès-Nancy, France
smaili@loria.fr

Abstract. Generative models get huge attention by researchers in different topics of artificial intelligence applications, especially generative adversarial networks (GANs) which have demonstrated good performance in data generation. In this paper, we would like to explore the potential of this class of models in producing human faces images. For that, we will use Deep Convolutional Generative Adversarial Network (DCGAN). Since that, the evaluation of GANs is still difficult even with the existing metrics like Inception Scores (IS), Mode Score (MS), Kernel Inception Distance (KID), Fréchet Inception Distance (FID), Multi-Scale Structural Similarity (MS-SSIM), etc. Thus, the best possible evaluation remains that carried out by human evaluators. This is why we propose a new hybrid measure combining qualitative and quantitative evaluation, we called this measure: Measuring the Quality of the Features of an Image (MEQFI). The images produced with the DCGAN method were trained on three well known datasets from the literature and the results were evaluated with MEQFI.

Keywords: Image generation · Generative Adversarial Network · Deep Convolutional Generative Adversarial Network · Batch Normalization.

1 Introduction

Generative adversarial networks (GANs) are a class of generative models proposed by Ian J. Goodfellow et al. [6], which consist of two adversarial players, a Generator that produces new data from random input and a Discriminator which classifies the produced and the real data. Both the Generator and the Discriminator try to maximize their success and minimize the success of the other one. This approach is inspired from a learning paradigm that was introduced by Schmidhuber in 1990 [23]. In this paradigm, two different neural networks compete each other in minmax game. However, the simultaneous training of the two adversarial networks makes the learning process noisy, unstable and hard to optimize. This leads to problems such as vanishing gradients [6] because of the adversarial training, mode collapse [5] which happens when the Generator produces a limited set of examples and the Discriminator is blocked in the local minimum and consequently it can't make any progress, especially when the model's parameters are destabilized. Several works have been developed to overcome these problems and improve the quality of the data generated, we find those which improve the architecture of the GAN by using additional classifiers or by adopting semi-supervised learning to guide the learning process and those which focus on modifying the objective function depending on the application domain. GANs have made great achievements since their introduction in image synthesis [2] [21], in style transfer [12], in image super-resolution [4] [15] [24] and in image-to-image translation [11]. Recently, applications have been extended to a wide range of fields, not only image generation, but also language and speech processing, security analysis, malware detection and chess program. One of the main challenges in machine learning is the collecting data and sizing datasets appropriately for relevant training. Thus, GANs can help to manage this problem by generating synthetic data to achieve data augmentation and dataset balancing.
In this paper, we focus on face generation tasks by using Deep Convolutional Generative Adversarial Networks (DCGAN) [20] and we introduce a human evaluation metric that calculates the score of each generated image based on human feedbacks. This paper is structured as follows: in Section 2 and 3 we summarize the previous related works done on GANs. Then we present the characteristics of the datasets

used for our study, the experiment configurations as well as the experimental results with analysis in section 4. After that, we review the used evaluation metrics and illustrate our evaluation metric in section 5. Finally, we draw the conclusion from the model that we have studied in section 6.

2 A recall of the Genarative Adversial Networks (GAN)

The GAN model consists of two different Deep Neural Networks. The Generator Network: takes random inputs Z to learn how to produce fake data and try to make them very similar to the real data X (training data). Thus, the objective of the Generator is to produce undistinguished data from the real one. The Discriminator Network: takes as input fake and real data and tries to learn how to distinguish between them using a loss function. Thus, the objective of the Discriminator is to recognize which data are fake and which ones are real and this is what makes the Discriminator works like a classifier.
GANs implement two neural networks, a Generator G and a Discriminator D. They both play an adversarial game where the Generator functions as a forger who at all times tries to fool the expert (the Discriminator) by generating data that is increasingly similar to that of the training set. The Discriminator is on guard not to be fooled into identifying fake data from real data. The Discriminator and the Generator work simultaneously as in a min-max game with two players D and G in which each one tries to maximize its success and minimize the success of the other one, in other words they try to reach the Nash equilibrium. In this model, the training process is split into 2 parts:

- Discriminator update. The D Discriminator is a binary classifier. Its inputs are data without a priori knowledge about the quality of true or false data. As output, it produces a probability that the data is true or false (those produced by the Generator). The weights of the Discriminator network are updated in order to reduce its classification error. After several iterations of updating the weights, D is better able to distinguish between true and false examples.
- Generator update. For the Generator, it first generates some fake examples using the noise z as input, and then these are passed to the Discriminator. Because the Generator does not know the real examples, at first it does not know how to produce real data, and the Discriminator can easily distinguish between real and fake data without much error. The Generator aims to maximize its probability of producing an example given the class label corresponding to real data. Then the Discriminator acts as what we have already explained above and after updating the parameters of G and calculating the cost, the gradient is then propagated backwards and the parameters of the Generator are updated (Figure 1).

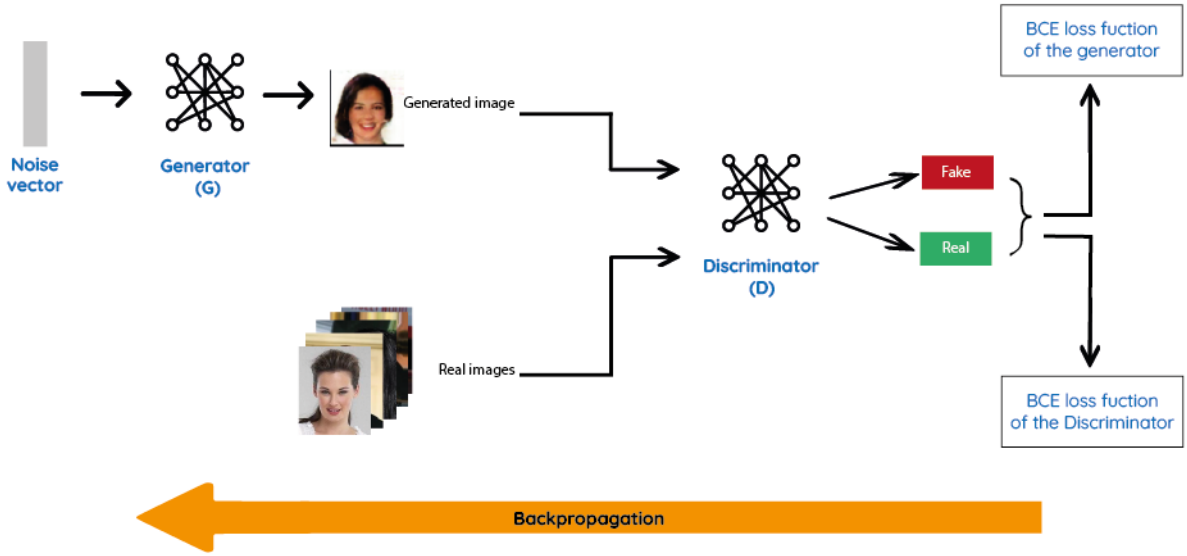


Fig. 1: GAN Architecture.

The cost function is given by the Formula 1.

$$V(D,G) = \underset{x \sim p_{data(x)}}{\mathbb{E}} [logD(x)] + \underset{z \sim p_z(z)}{\mathbb{E}} [log(1 - D(G(z))] \tag{1}$$

The first item is the expectation of the log of the Discriminator output when the data is from the real data distribution. The second parameter is the expected value of the log of the quantity of one minus the Discriminator prediction on the fake samples. The method consists in maximising $D(x)$ and minimising $D(G(z))$. So that is why, $1 - D(G(z))$ is used in order to make the Generator and the Discriminator moving in consistent directions between the two terms in the equation.

3 Related Works concerning the variants of GANs

Several works have been done to improve the quality of the GANs, we can summarise them into two main variants:

3.1 Architecture-variant

This category regroups works that improve the quality of the generated images by changing the architecture of the Generators and Discriminators, like: DCGAN which we will study in this paper, LAPGAN [3] which combines the model of conditional GAN (cGAN) with Laplacian pyramid (LP) framework which is constructed from the Gaussian pyramid (GP). Or by changing the latent space like cGAN [18] that uses label information into the Generator and the Discriminator in order to control the type of the generated data. The semi-supervised GAN (SSGAN) [19] has been inspired from the principle of semi-supervised learning and the ability of GANs to produce unlabelled data, so its Discriminator becomes a multi-class semi-supervised classifier with Softmax and Sigmoid activation functions for the classification of the fake and real examples. In [26], they used a modified architecture of GAN according to the application domain. CycleGAN [26] is used to achieve image-to-image translation when unpaired training data is not available. In [15], the method based on GAN generates high-resolution images from low-resolution images by upsampling, also it extends the loss by adding content loss.

3.2 Loss-variant

This category regroups works that improve the performance of GANs by changing the type of the loss function, like: WGAN [1] which uses Earth-Mover (EM) or Wasserstein distance instead of using BCE loss function and here the Discriminator is used to fit the Wasserstein distance. In [1], they proved that WGAN overcome the vanishing gradient problem and remove the mode collapse partially. Others propose to add additional penalisation to the loss function or any type of normalisation applied to the network, like: WGAN-GP [7] which is a variant of WGAN with gradient penalty.

4 Deep Convolutional GAN: a method adopted for human faces images producing

Since fully connected layers reduce the quality of generated images in the GAN model, in [20], the authors introduced the concept of Deep Convolutional GAN (DCGAN) that permits to generate high-quality images. The authors replaced the pooling layers with:

- Strided Convolutional layers in the Discriminator to down-sample the images.
- Fractionally–strided convolutional layers in the Generator to up-sample the images.

They removed the fully connected hidden layers, they also used batch normalization [10] in order to help the training process to become more stable by normalizing the inputs of each layer. The general architecture of the DCGAN is illustrated in the figure 2:

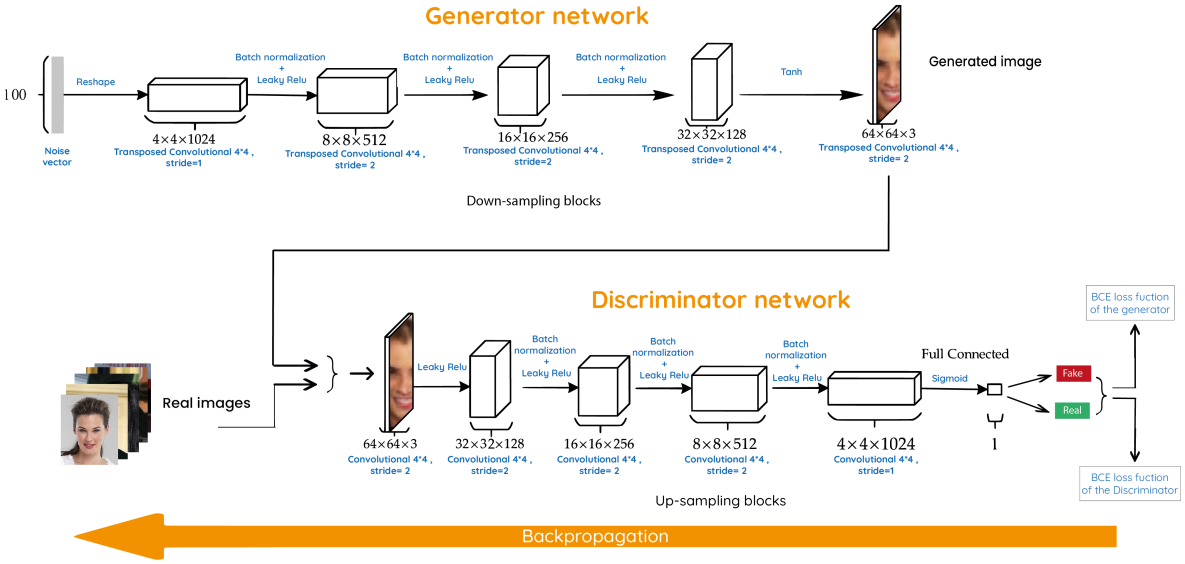


Fig. 2: DCGAN Architecture.

4.1 Datasets

To evaluate our DCGAN model, we used three databases of famous faces, the first is *CelebA* , which was initially collected by researchers from the Chinese University of Hong Kong MMLAB. This dataset is a large-scale face attribute database, it contains more than 200,000 images of more than 10,000 celebrities, it covers many images of different races, ages, genders, hairstyles, eye colors, etc. We summarise them in 40 binary attributes (see table 1). The second is *CelebA-HQ* [13], which is an improved version of CelebA in terms of image quality. The third is the Labeled Faces in the Wild *LFW* dataset [9], which was created and maintained by researchers at the University of Massachusetts. It contains 13,233 images of 5,749 people annotated with the same 40 binary attributes as in the CelebA database.

Index	Definition	Index	Definition	Index	Definition	Index	Definition
1	5o'ClockShadow	11	Blurry	21	Male	31	Sideburns
2	ArchedEyebrows	12	BrownHair	22	MouthSlightlyOpen	32	Smiling
3	Attractive	13	BushyEyebrows	23	Mustache	33	StraightHair
4	BagsUnderEyes	14	Chubby	24	NarrowEyes	34	WavyHair
5	Bald	15	DoubleChin	25	NoBeard	35	WearingEarrings
6	Bangs	16	Eyeglasses	26	OvalFace	36	WearingHat
7	BigLips	17	Goatee	27	PaleSkin	37	WearingLipstick
8	BigNose	18	GrayHair	28	PointyNose	38	WearingHat
9	BlackHair	19	HeavyMakeup	29	RecedingHairline	39	WearingNecktie
10	BlondHair	20	HighCheekbones	30	RosyCheeks	40	Young

Table 1: List of the 40 binary attributes extracted from the images of CelebA database.

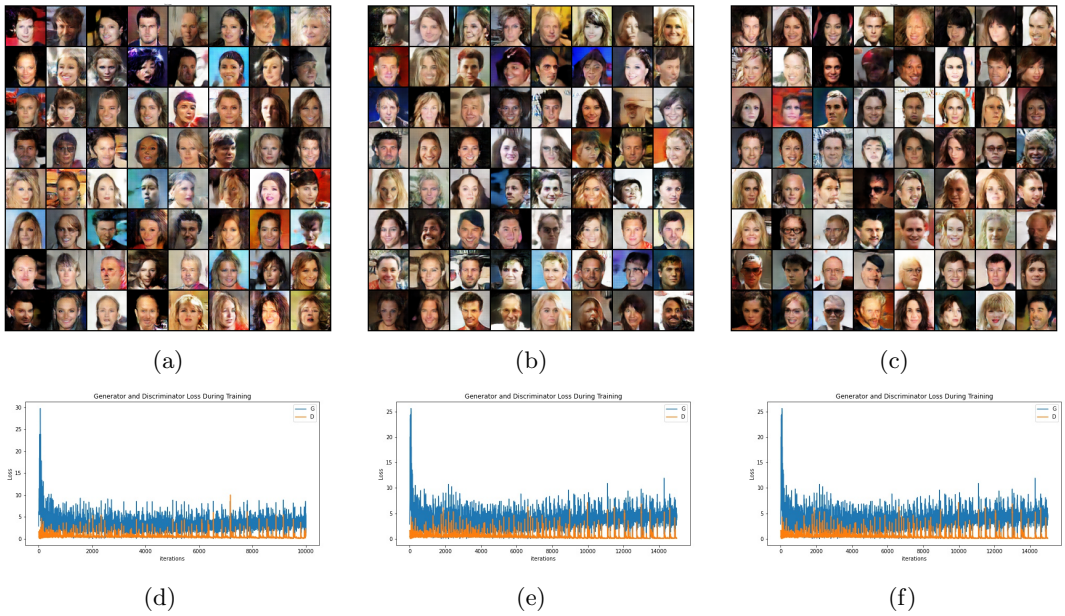
4.2 Configuration

For the implementation of DCGAN Model, we use the following hyperparameters: batch size = 128,image size =64×64, noise vector =100, number of features in both of the Generator and the Discriminator 64, and we use the Adam optimizer [14] to update both G and D neural networks with a learning rate of 0.0002 and β_1 =0.5. We also used the LeakyRelu activation function [17] [25] in the generator and discriminator, instead of Relu, because it prevents the neural network from getting stuck in the death state, where the neural network only produces zeros for all outputs, so that the learning process no longer learns.

4.3 Results

For the test we use 12800 images from each dataset: CelebA, CelebA-HQ (only female faces) and LFW by testing several values of epochs: 100, 150 and 200.
In Figures 3a, 3b and 3c, we give the illustration of the images produced by training the DCGAN on CelebA. While we analyse the loss function of Figures 3d, 3e and 3f, we can remark that after the first epochs that the Generator learns to fool the Discriminator by creating more or less real images. The overall shape of the Generator's curve is decreasing, where the values are ranging from a loss of about 29,91 to 3,34. But unfortunately, the loss value gets stuck in the range $[1.2 \dots 6.11]$. As for the discriminator from the beginning, it manages to distinguish the fake data, nevertheless for certain batches, it experiences some difficulties and is fooled by the Generator. And finally, regarding to these experiments, it is preferable to use only 100 epochs, increasing the number does not help to improve the training.

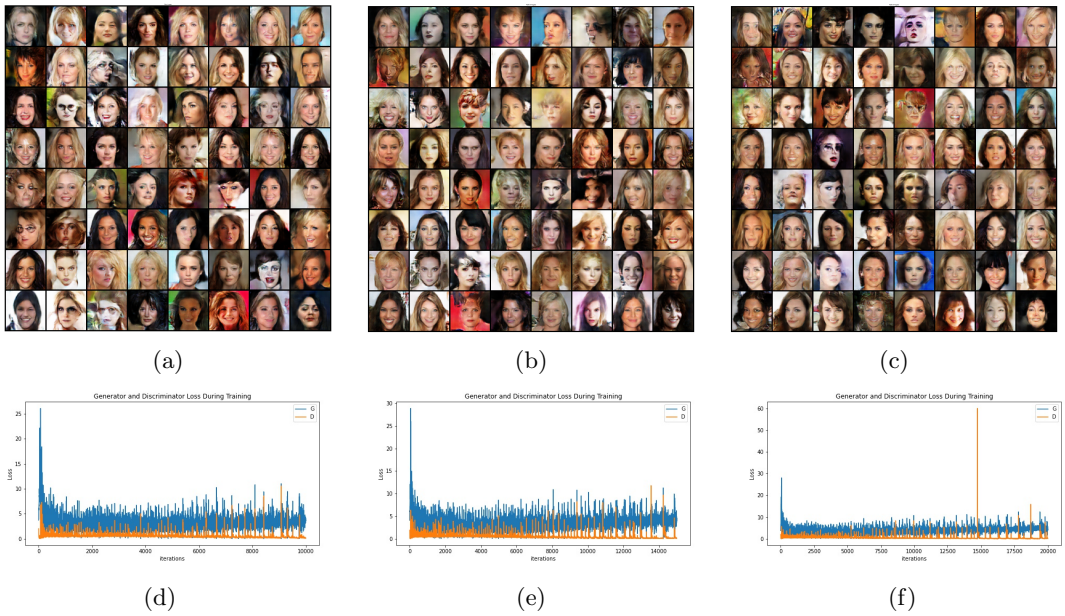
Concerning the experiment of Figure 4 with a high quality images of women faces, the aspect of the images are more accurate, and we can remark also as for the experiment of Figure 5 that for 100 epochs the results are better than the other two other experiments with 150 and 200 epochs. During the training, the initial generated images are not very good, the Discriminator was able to classify them easily, but as the training progresses the generation process becomes better and the quality of the synthesised images is more accurate. This makes the classification task difficult for the Discriminator because the generated images become more similar to the real ones.
For a general comment concerning the previous experiments, based on the loss functions of the Generator and Discriminator of each dataset ((d), (e) and (f) of Figure 3, 4 and 5), the process oscillates in a narrow range, this is due to the adversarial training and the fact that the system reaches the Nash equilibrium, which is a state where the Discriminator can not distinguish between real and fake images. But it is difficult to achieve this equilibrium because of the simultaneous training of the two adversarial networks. We also found that the Discriminator's loss value decays towards zero faster than the Generator's loss value.



(a) (b) (c)

(d) (e) (f)

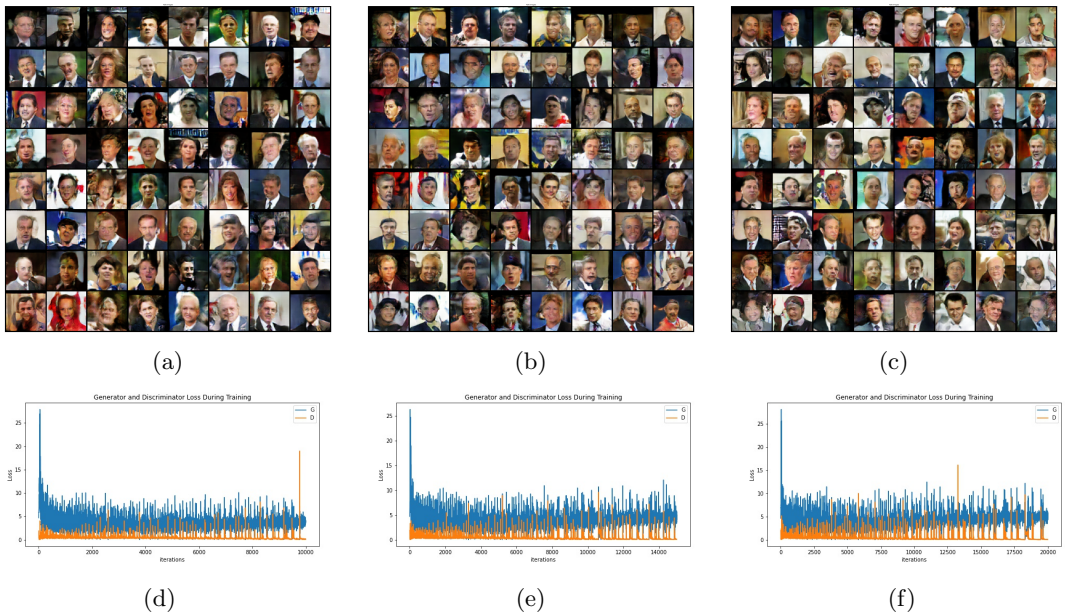
Fig. 3: Different results of our model from CelebA dataset with 100 batches of 128 images. (a), (b) and (c) represent the generated images after 100, 150 and 200 epochs respectively. (d), (e) and (f) the G (blue color) and D (orange color) losses after 100, 150 and 200 epochs respectively.



(a) (b) (c)

(d) (e) (f)

Fig. 4: Different results of our model from CelebA-HQ-females only dataset with 100 batches. (a), (b) and (c) represent the generated images after 100, 150 and 200 epochs respectively. (d), (e) and (f) the G (blue color) and D (orange color) losses after 100, 150 and 200 epochs respectively.



(a) (b) (c)

(d) (e) (f)

Fig. 5: Different results of our model from Labelled Faces in the Wild dataset with 100 batches. (a), (b) and (c) represent the generated images after 100, 150 and 200 epochs respectively. (d), (e) and (f) the G (blue color) and D (orange color) losses after 100, 150 and 200 epochs respectively.

5 Evaluation

One of the hardest tasks in this kind of process is how to find a good metric that allows to evaluate the images achieved by a process based on GANs algorithm. In the literature we identified two types of evaluation:

- Qualitative evaluation, where we will evaluate the model based on the human subjective evaluation [3] [16] and according to the feedback, one can get an idea about the quality of the produced images. This is what is done in several research areas in artificial intelligence.
- Quantitative evaluation, where a specific numerical score is calculated indicating the quality of the produced images. In this topic, we identified 24 quantitative metrics to evaluate GANs and the most used one is the Inception score (*IS*) which was proposed by Salimans et al. [22]. The idea is to apply an inception model on each generated image to get a conditional label distribution $P(y|x)$, the ideal images are those which have a distribution obeying to a low entropy. The inception score has the lowest value of 1 and the highest value of the number of classes supported by the classification model, and with this measure, we expect that the model will produce divers images. The inception score is calculated by estimating the class conditional probabilities for each generated image. The images that have a higher ranking in one class compared to all other classes correspond to the best images. And therefore the conditional probability of all generated images must have low entropy. As it is a question here of obtaining a diversity of images, it is therefore necessary to use the marginal probability $\int_z P(y|x = G(z))dz$, that is to say the probability distribution of all the images generated. To verify this condition of obtaining a maximum of diversity of images, it is therefore necessary to maximize the entropy. The general formula of this score is a combination of these two distributions by using the Kullback-Leibler divergence (KL) between the conditional and the marginal distributions knowing that here, we are interested in the average of the KL divergence for all generated images. The inception score for a generator is given by:

$$IS(G) = exp(\mathbb{E}_{z\sim P(z)} KL(P(y|x)||P(y)) \tag{2}$$

There is another measure named Fréchet Inception Distance (FID) which was proposed by Heusel et al. [8]. It measures the similarity between two distributions (real data x and generated data g) by applying embeddings on real and fake images to extract their features. The embedding is computed

using a pre-trained convolutional network called Inception-v3 which was proposed by a team at Google. The formula of FID is given by:

$$FID(x, g) = ||\mu_x - \mu_g||_2^2 + Tr(\sum_x + \sum_g -2(\sum_x \sum_g)^{1/2}) \quad (3)$$

where: μ_x and μ_g are the feature-wise mean of real and generated images, respectively. $\sum_x$ and $\sum_g$ are the covariance matrix of real and generated images, respectively.

5.1 Our Contribution

In this section, we propose a hybrid measure based on a qualitative evaluation for each present feature and which is summarized by a quantitative score. In the following, this score will be named Measuring the Quality of the Features of an Image (MEQFI). For that, we define a list of F features that will be used to calculate the score of each generated image. For each feature i, we attribute a mark based on human feedback to calculate its probability P_i. We multiply this probability by an activation coefficient α_i. This score is given by:

$$Score(m_k) = \frac{1}{F - I} \sum_{i=1}^{F} \alpha_i * P_i \quad (4)$$

Where :

- m_k: is the k^{th} produced image to evaluate.
- F: represents the number of features that we use to evaluate the quality of the generated image.
- I : represents the number of features ignored for some images from the generated images, for example: if the generated image is not smiling, we will eliminate the feature teeth.
- α_i: represents the activation coefficient of each feature. It is equal to zero if the corresponding feature is ignored otherwise, it is set to 1.
- P_i: represents the probability that the model succeeded in creating the attribute i. This probability is calculated as follows: $P_i = \frac{mark}{scale}$.

After calculating the score of each generated image, we calculate the average score over all the generated images.

5.2 Evaluation of the images produced by DCGAN with MEQFI

Fig. 6: Eight generated images from CelebA-HQ only female.

We fix a group of features {Nose, Eyes, Eyebrows, Mouth, Ears, Skin tone} to calculate the score of each image by using MEQFI. For that, we ask a group of evaluators to attribute a mark from one to five for each feature of each image. To illustrate the result on a concrete example, we give for the images of figure 6, the results in the Table 2 in terms of the MEQFI measure for each of these images and for each attribute.

Image	Nose	Eyes	Eyebrowns	Mouth	Ears	Teeth	Skin Tone	Image Score (%)
	0,6	0,6	1	0,4	0	0,5	1	68,33
	0,5	0,6	1	0,4	0,5	0,5	0,9	62,86
	0,2	0,5	0,8	0,4	0,5	0,2	1	51,43
	0,4	0,3	0,6	0,4	0,5	0,2	0,8	45,71
	0,4	0,5	1	0,5	0,5	0	0,9	63,33
	0,4	0,6	1	0,5	0,4	0,6	1	64,23
	0,4	0,4	1	0,4	0	0,7	1	65,0
	0,4	0,5	0,7	0,6	0,4	0	0,9	58,33
Totale Score %								**59,91**

Table 2: MEQFI Score on eight generated images from CelebA-HQ-females.

Each line of the Table 2 represents the application of MEQFI on the concerned image.
After having collected the human feedbacks of the 64 images for each of the three datasets, we calculate the score of each image with MEQFI, and we plot the results in the Figure 7 :

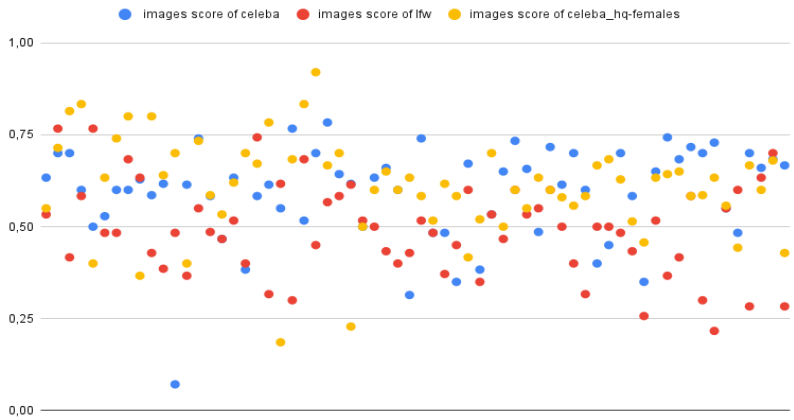


Fig. 7: MEQFI score for 64 images generated by DCGAN from the datasets (CelebA, CelebA-HQ-females, LFW).

Where the blue, yellow and red dots correspond to the results on the CelebA, CelebA-HQ-females and LFW corpora respectively. According to the distribution of dots, we can remark that the yellow ones have higher scores (CelebA-HQ-females) than the blue dots (CelebA) and the red dots (LFW) at the end. This confirms what we discussed in Section 4.3. The total score on the 64 generated images of each dataset is given in the Table 3.

Dataset	**Total Score (%)**
CelebA	59,18
CelebA-HQ-female	60,74
LFW	49,12

Table 3: MEQFI score of 64 DCGAN generated images from different datasets (CelebA, CelebA-HQ-females, LFW).

6 Conclusion

In this work, we discuss the use of generative adversarial networks in human face imaging. We focus on using deep convolutional GANs to produce new data and to achieve data augmentation of human face images. Based on our experiments, the DCGAN architecture generates good images with similar properties to the training distribution, especially when the training dataset contains high quality images. The training process of DCGAN models is still unstable, even with the present techniques that help to improve the stability of the model. Indeed, we will address this issue in our further works. Observing the generated images, we remarked that the generated images from CelebA-HQ (only female faces) dataset are much better than the two other datasets because it contains high quality images, and also because we used only female faces only which reduces the noise in the generation process. In this article, we developed also a new measure to evaluate the quality of the produced images, this is the measure MEQFI that allows to take into account the evaluation of each annotator and combines them into a single score. The scores achieved by this measure are correlated to what we observed with the loss function during the learning step.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large scale GAN training for high fidelity natural image synthesis". In: *arXiv preprint arXiv:1809.11096* (2018).

[3] Emily L Denton, Soumith Chintala, Rob Fergus, et al. "Deep generative image models using a Laplacian pyramid of adversarial networks". In: *Advances in neural information processing systems* 28 (2015).

[4] Hao Dong et al. "Semantic image synthesis via adversarial learning". In: (2017), pp. 5706–5714.

[5] Ricard Durall et al. "Combating Mode Collapse in GAN Training: An Empirical Analysis using Hessian Eigenvalues." In: (2021), pp. 211–218.

[6] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[7] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017).

[8] Martin Heusel et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (2017).

[9] GB Huang, M Ramesh, and T Berg. "Learned-Miller". In: *Labeled faces in the wild: A database for studying face recognition in unconstrained environments* (2007).

[10] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: (2015), pp. 448–456.

[11] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: (2017), pp. 1125–1134.

[12] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: (2019), pp. 4401–4410.

[13] Tero Karras et al. "Progressive growing of gans for improved quality, stability, and variation". In: *arXiv preprint arXiv:1710.10196* (2017).

[14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[15] Christian Ledig et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: (2017), pp. 4681–4690.

[16] Christian Lopez, Scarlett R Miller, and Conrad S Tucker. "Human validation of computer vs human generated design sketches". In: 51845 (2018), V007T06A015.

[17] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: 30.1 (2013), p. 3.

[18] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[19] Augustus Odena. "Semi-supervised learning with generative adversarial networks". In: *arXiv preprint arXiv:1606.01583* (2016).

[20] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[21] Scott Reed et al. "Generative adversarial text to image synthesis". In: (2016), pp. 1060–1069.

[22] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016).

[23] Jiirgen Schmidhuber. "Making the World Differentiable: On Using Self-Supervised Fully Recurrent N eu al Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environm nts". In: (1990).

[24] Han Wang et al. "Image super-resolution using a improved generative adversarial network". In: (2019), pp. 312–315.

[25] Bing Xu et al. "Empirical evaluation of rectified activations in convolutional network". In: *arXiv preprint arXiv:1505.00853* (2015).

[26] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: (2017), pp. 2223–2232.