Densely Connected Convolution Network using Generative Adversarial Nets (GANs)

Ahmed M.Y.Albhnasaswy
College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport
Cairo, Egypt
ahmedbhna@gmail.com

Yasser.M.K.Omar
College Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport
Cairo, Egypt
Dr_yasser_omar@yahoo.com

Essam Fakharany
College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport
Cairo, Egypt
essam.elfakharany@aast.edu

Abstract

Synthesizing realistic images has been a challenge in machine learning, due to the images that are complex and highly dimensional, thus making them hard to model well. This paper proposes an extension to the Generative Adversarial Networks (GAN) namely DenseGAN to synthetically generate more challenging and more complex images such as the faces of people or cats which contain large amounts of features. The key innovation of this piece of work is to replace the traditional Convolutional Neural Network (CNN) in the discriminator with the newly Densely Connected Convolution Network (DenseNet); this aims at ensuring the maximum information flow of the discriminator layer and enhancing the gradient outpouring to the generator. Empirically, this proposal aims to show that the proposed DenseGAN is capable of creating realistic facial images. Furthermore, a detailed comparison between images generated by standard GANs, DCGAN and the new proposed model (DenseGAN) was conducted .The new model using CAT and CelebFaces Attributes (CelebA) dataset is executed and the results obtained from the generating task are demonstrated instead.

Keywords: Generative Adversarial Network, Convolution Neural Network, Densely Connected Convolution Network, Celeb Faces Attributes.

1. Introduction

Deep learning is both a popular and a controversial research topic nowadays. More concretely, supervised learning has shown even human- competitive performance for discriminative tasks [1]. This Impressive performance achieved by Artificial Neural Network (ANN) and the use of huge amount of data, logged into the network and let the network learn by applying a series of non-linear operations to draw a decision boundary which would, for example, let a network classify an image among 1000 classes [2]. As a result, the controversy of whether the results of unsupervised learning are still by far less than the results obtained by supervised learning or not has become a question that inspired the researchers to aim through their thorough research to help bridge the gap between the success of CNNs for supervised and unsupervised learning while focusing on an important subfield of unsupervised learning which is a generative modeling. Moreover, the applications of intelligent machines in mimicking human-like synthesis present a more complex and visionary objectives that any researcher would have ever dreamt of. Eventhough the goal seemed too optimistic to begin with, GAN has advances towards achieving a state of the art of synthesis. GAN has shown promising results as a generative model [3]. It uses an adversarial training based on minimax algorithm for training generative model. After the impressive result achieved by GAN in generation tasks, the generative models were positioned at the top of research areas especially in image and video generation. Significant improvements in image generation [4], [5] and video prediction[6] were demonstrated. The research with its generic framework is divided into two phases; one that attempts to stabilize the training procedure and the other that applies GAN on the problems of interest, for example, filling in any missing data [7]. It remains unclear why there is less focus on building new architucture in GAN to improve these results.

In addition to making use of GANs in achieving state of art results, GANs have provided the researchers with an out-of-the-box tool to generate artifacts from noise with an unknown underlying distribution [3].

In this paper, a straight forward method constraints on the architectural topology of Densely GAN that make them stable to train and generate 64 by 64 images are to be proposed and evaluated. This class of architectures is named Densely Generative Adversarial Network (DenseGAN). The less of variations in architectures used in the generative model prompted the researchers to introduce a Dense-GAN framework. This research introduces a class of DenseCNN to show how to scale up GAN using DenseNet[8] from difficult adversary to a forgiving instructor. This approach is believed to be a novel contribution to the research community. The optimality of such a system when deployed to generate realistic image is put into questioning; though many researches have been conducted on the basic direct sequential connections in the model but few experiments have been carried on to the full extend connections of the network.

Also, the rest of the paper is structured as follows: The basics of the GAN framework are explained in **Section 2.1**, GAN and the architecture of the DenseNet will be presented in **Section 2.2**. **Section 3** provides an overview of the existing methods for image generation task. The presented model, DenseGAN, is described in **Section 4**. **Section 5**provides an evaluation and comparison of the presented DenseGAN to different image generative models. Finally, the conclusion and future work will be presented in **Section 6.**

2. Background

In this section, the existing work which this proposed model is built upon will be described . Section 2.1 describes GAN framework and section 2.2 describes the DenseNet framework.

2.1. Generative Adversarial Network

The basic idea of GAN is to train two models, the first model is called the discriminator D(x). It takes an input (an image) and output a probability scalar that announces whether the input is real image or fake one. D(x) is similar to energy function, where x is from training samples which means when it's real it takes low values, otherwise it takes high value. The second model is called the generator G(z). The generator accepts input from random source of noise z $\sim$ P(z) in which uniform or gaussian can be distributed , and is trained to generate fake samples that the other model cannot distinguish from real data .The generator model aims at confusing the discriminator by generating samples similar to real image. Thus the main objective function of GAN to train G and D together can be expressed as follows:

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

Where $pData(x)$ is the true data distribution and $pz(z)$ is the noise distribution. Generator and discriminator both are differentiable functions. In the following approach, both G and D are assumed to be Deep Neural Networks (DNN), where θ_G and θ_D are parameters of G and D, respectively. Stochastic Gradient Descent (SGD) updates the parameters simultaneously. Each player is designed to minimize the cost function competing with the other player in the objective function of the Equation (1).Later, both models would then be updated. The stopping criteria of the GAN model is to reach the Nash Equilibriums (NE).

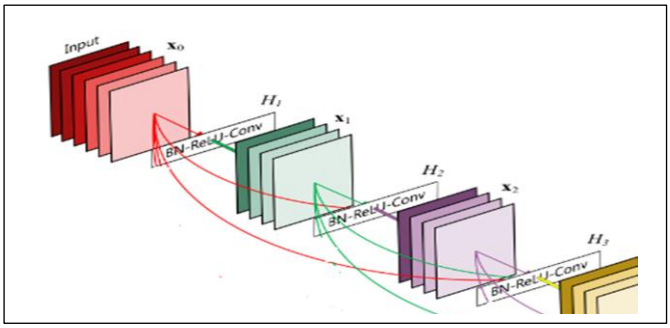

Figure 1: DenseNet architecture with 4-layers dense block. Each layer takes all preceding feature-map as input.

2.2. Dense connected convolutional networks

To further improve the information flow between layers, Gao Huang et.al [8] proposed a different connectivity pattern which introduced direct connections from any layer to all subsequent layers. Figure 1 illustrates the DenseNet architecture. The idea of DenseNet is to connect every other layer in a feed-forward fashion. For each layer, the feature-maps of all previous layers are used as inputs, and its own feature-maps are used as input into all the next layers ; consequently, the $\mathscr{H}$ layer receives the feature-maps of all preceding layers as input, which can be expressed as:

$$Xl = Hl([x0, x1, \ldots., xl-1]), \qquad (2)$$

Where [x0,x1,…xl-1] are the concatenated feature-maps produced from the previous layer, an advantage of DenseNet is that the gradient flows directly through the identity function from the last layers to the above layers

and prevents any information impediments to reach all the subsequent layers. DenseNet proves that concatenating feature-maps are learned by the increase of the variations of different layers in the posterior layers input improving efficiency.

3. Related Work

Current state of the art models moved away from Restricted Boltzmann machine (RBMs)[10], because RBM was primarily used in pre-training a classification network, and it fell out of favor, due to difficulties in training and likelihood estimation .

Another work focuses on using another generative model that is called Auto encoders .Vincent et al. [11] established Denoising Autoencoder, which reconstructs the original artificat from a corrupted version of it. A recent relevant ide is Variation Auto encoders [12] that provides probabilistic interpretation and uses the direct means of squared error instead of adversarial network, so it tends to produce blurry images. From different angel, recent researchers have proposed a good generative model for drawing chairs with different shape by training convolutional neural nets [13].

Superior to all the previous Generative models is, L. Goodfellow who introduces a different generative model that is known as the Generative Adversarial Networks (GAN)[3]. In his GAN framework, he uses an adversarial training based on minimax algorithm for training generative model and generated images suffering from being noisy. Positive results were shown in reference [14] in his work Conditional GAN for face generation. His work extended the CGAN [7] to propose a conditional version of GAN by feeding a label Y to both the generator and the discriminator agents in a GAN. CGANs were deployed to generate MNIST digits and were applied in image tagging [6]. Denton et al. [5], proposed Laplacian Pyramid of Generative Adversarial Networks (LAPGAN) that combined CGANs into the framework of Laplacian pyramids. LAPGAN demonstrated its ability to generate high- quality images, but still suffered from the objects tending to wobble. Historical attempts to scale up GANs using CNNs to model images have been unsuccessful. This motivated the authors of DCGAN[3] to develop an impressive architecture and they have shown a promising direction for image generation task based on his deep architecture.

T. Salimans et al. [15] demonstrated various techniques and tricks for stabilizing the training procedure of GAN and forced the discriminator to output the class label for the training data using auxiliary decoder network .GAN has been used in tasks such as image-to-image translation [16] and semantic image in-painting [17]. In Auxiliary Classifier GAN(AC-GAN)[18], the author used the discriminator in extra tasks as a classifier and proved that it would lead to improving the original task which is synthetic cohesive 128X128 images. Springenberg et al.[19] proposed CAT-GAN which can be used for semi-supervised learning and unsupervised learning by modifying the objective function to output a distribution over classes.
All this success has built up the Radford et.al [3] architecture DCGAN which was focused to build a strong generator that can generate samples in different settings and can converge faster, which has less remains and focuses to progress the architecture of discriminator and lessen the DCGAN problematic failure mode which is a mode-collapse and vanishing gradient problem.

4. Proposed model

The proposed approach introduces a variant architecture of GAN, that is called a DenseGAN to synthesize images of size 64-by-64 that depend on the power of the discriminator model rather than the generator. In this paper, the aim is to let the generator become obedient to the discriminator instructions which would take the role of the discriminator from difficult adversary to forgiving instructor. Obedience would come from increasing the gradient following the generator from the discriminator that lead the generator to enhance the generated samples. This straightforward contribution is by replacing the traditional discriminator in DCGAN with the DenseNet objective. Motivated by these considerations in section 2.1 and 2.2, a class of DensCNN to scale up GAN using denseNet to a densely version of GAN, that allows the discriminator layer introduced a direct connections from any layer to all subsequent layers to let the generator regulate training and reach higher performance in a fraction of the training time required for the standard GAN model is introduced. Adding more connection to adversarial nets doesn't aim to accelerate the convergence but to ensure the maximum information and gradient flow between discriminator layers.

Our discriminator is considered one dense block that contains different composite functions of three consecutive operations: batch normalization [21], followed by rectified liner unit (ReLU) and a 4x4 convolution (conv).

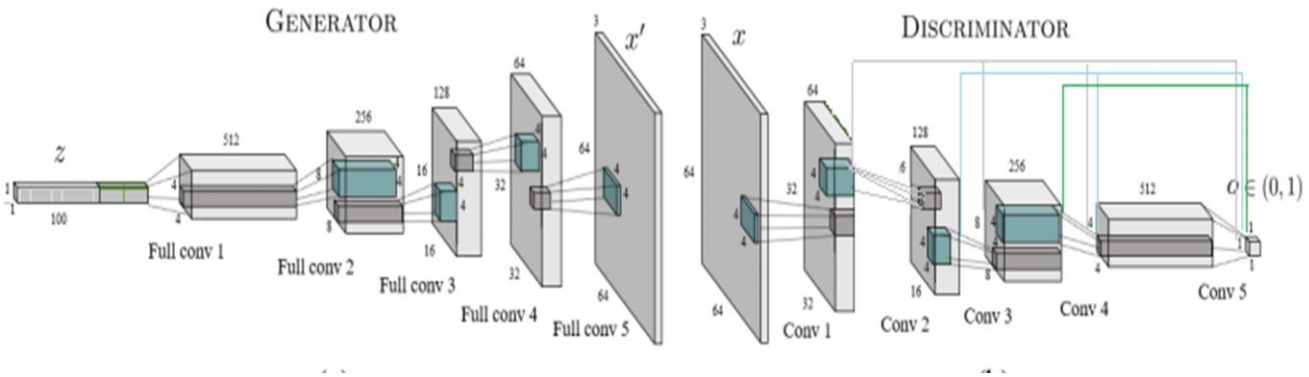

Figure 2: Architecture of the generator (a) and discriminator (b) of DenseGAN model.

The goal of applying more connections and gradients to be followed in the discriminator layers happens when the generator generates samples far away from being real, and in this case, it will get a gradient indicating how to enhance its output to reach the real data. This aims to control the generational procedure and improves the quality of generated images by increasing the information flowing between layers. The framework is illustrated in Figure 2.

To keep the feed-forward quality, each layer in the network of the discriminator obtains additional input from all the previous layers. In this, the researchers concatenate the feature-map instead of combing them as similar to DenseNet. Additionally, the pooling layer in the discriminator layers are to be removed as they are considered responses for changing the size of feature map to not miss the matching features while aiming at concatenating them to modify the concatenation method. The cost function of the DenseGAN generator can be expressed as:

$$\mathrm{CG} = \frac{1}{M}\sum_{I}^{N} Ex \sim pG(x)\big[\log\big(1 - Di(x)\big)\big) \qquad (3)$$

where $pG(x)$ is the distribution induced by the generator $G(z)$ and M represents the size of mini-batch. $D* = arg\,maxD\,V\,(D,G)$, gradient descent on $pG(x)$ will recover the globally designed optimal solution, $pG(x) = pdata(x)$, so that the generator's distribution exactly matches the data distribution.

To enhance the gradient at the start of the training , the log(1-D) term should be replaced with –log (D) that is similar to [4], so the equation can be rewritten as :

$$\mathrm{CG} = -\frac{1}{M}\sum_{I}^{N} Ex \sim pG(x)\big[\log\big(Di(x)\big)\big) \qquad (4)$$

The traditional discriminator layers skip the redundant feature maps by passing them on from layer to layer. The feature of each layer will be concatenated in discriminators D_i. When the dimension of the layer in the discriminators is $n \times n \times d$, where d is the depth size, the researchers would start to replicate the feature vector spatially to match the size $n \times n$ of the feature map and perform a depth concatenation. This aims at making the discriminator's layer saturation from learned features and support the generator by knowing what it's going to generate well and to ensure maximum information flow between layers in the network. The feature concatenation that is proposed by [8] in the classification task is used in the presented DenseGAN to improve the quality of synthesized images. On the other hand, the generator G will accept latent distribution $P(z)$ which is assumed to be the uniform distribution $P(z)=\mathcal{U}(-1,1)$.

4.1. Model Architecture

In this approach, a variant of GAN,is introduced and is called DenseGAN to synthesize images. For implementing DenseGAN, Tensorflow[22] will be used. All the models were examined using identical architectures and hyper parameters (Deep Nets).

Preliminaries. Let X = {X_i | i = 1,...,n} be a dataset, where Xi represents data instances and n represents instances number in the dataset (|X| = n). Every data instance is a pair $X_i = (I_i, C_i)$, where x_i is an image, and c_i is the class label to which the image corresponds. For training the DenseGAN we let the generator G accept the noise vector and generate an image $x_{fake} \sim p_g$. on the other hand, the discriminator D receives as input as image an we train the discriminators by accepting mini-batch of x_{fake} or x_{real} to as an input and) and output a probability scalar that announces whether the input is real image or fake.

Table 1: Detailed generator and discriminator architecture

Generator						Discriminator					
Operation	**Kernel**	**Stride**	**Filters**	**BN**	**Activation**	**Operation**	**Kernel**	**Stride**	**Filters**	**BN**	**Activation**
						Convolution	4 × 4	2 × 2	64	No	ReLU
Full convolution	4 × 4	2 × 2	512	Yes	ReLU						
Full convolution	4 × 4	2 × 2	256	Yes	ReLU	Convolution	4 × 4	2 × 2	128	Yes	ReLU
Full convolution	4 × 4	2 × 2	128	Yes	ReLU	Convolution	4 × 4	2 × 2	256	Yes	ReLU
Full convolution	4 × 4	2 × 2	64	Yes	ReLU	Convolution	4 × 4	2 × 2	512	Yes	ReLU
Full convolution	4 × 4	2 × 2	3	No	Tanh	Convolution	4 × 4	1 × 1	1	No	Sigmoid

Generator. The architecture of generator G is similar to DCGAN generator which is formed from a sequence of layers. It starts with fully convolution layer with a number of hidden units =512 followed by transpose convolution layers (de-convolution). Table 1 above describes the generator and discriminator architecture. The generator G ended with hyperbolic tangent activation (non-linearity). The generator accepts noise distribution vector of size 100 $z \in Z_i$ and output generated images (X fake) $Xfake = G(z)$. Thus, Batch normalization [20] was done after each layer.

Additionally, after trying different configurations, the Batch Normalization and non-linear activation functions from the last layer were removed to guarantee that the output distribution is similar to P(z). The Generator was trained with Adam optimizer [21] with learning rate =0.0002. The output image size used as a bas Our precise task is to parameterize G so that it replicates the empirical density model $Pdata(x, y)$,and the generated x' is 64x64x3.

Discriminator. We build an adversarial discriminator that takes image vectors and insert it to the model, to decide whether they are real samples or generated. This adversarial model is using convolutional layers followed by ReLU activation (non-linearity) and dropout layer to prevent over fitting and regulate the model. Dropout is applied at all layers except the output layer of discriminator. The discriminator activation output is sigmoid function.. The auxiliary information is added to discriminators input by concatenation method; it contains information about the class of the training example X. The discriminators are trained with Adam optimizer [22] with learning rate=0.002 then it was decreased to 0.0002. Global parameters in the experiments shown in table 1.
The discriminator tasked to distinguish samples from training data $Pdata(x)$ and samples from the model while the generator tasked to fool the discriminator.

5. Experiment Scenario and Evaluation

Many experiments are applied on the existence of two different datasets within Tensorflow[22] library. The proposed model objectives are to be evaluated by improving the sample quality on variant datasets.

5.1. Dataset

Two-image datasets of different complexities and variations were used, Cat [23] and CelebFaces Attributes (CelebA) [24]. CAT dataset [23] for cat-face images have a diverse range of backgrounds, which can range from severely harmful distraction to the presented model of learning how to generate cat faces. This noisy background data was disregarded by centering the images on the cat faces and removing the outliers.
CelebA is a dataset composed of 202,599 face colored images and 40 attribute binary vectors. The aligned and cropped version was used , and the images were scaled down to 64x64.
Furthermore , the images for both datasets were preprocessed by scaling them to a [-1, 1] to match the distribution given by the hyperbolic tangent used in the last activation function of the generator, using the normalization to pixel values X=[-1,1] X[-1,1]X channel number .

5.2. Details of training

The generator and discriminators are trained from scratch using SGD. We have initialized the weights for the network and have not loaded other network weights. Weights were initialized from normal initializer with standard deviation=0.02 and bias initialization of constant zero. A weight decay of 10^{-4} and a momentum of 0.9 was used.
(Samples were drawn independently at each updated step) during 200 epochs, and the generated samples after every single epoch were saved. To prevent overfitting in the discriminator network a dropout layer was added after each convolution layer, except the first layer and the drop rate was set to 0.4.

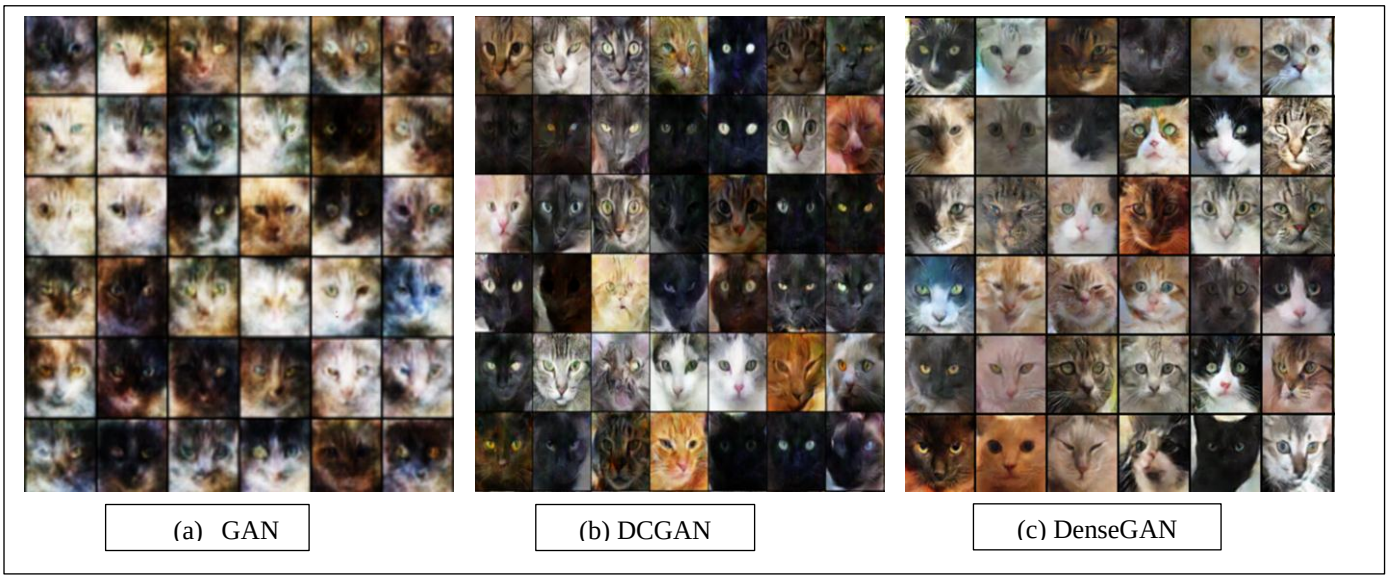

Figure 3: The results of the comparisons are shown in different generative models. Considering the Generation from Cat dataset, each cat image is 64x64x3. DenseGAN shows an increase in the quality of the generated samples. Samples of GAN are more noisy in comparison to the presented model and DCGAN.

DenseGAN system was tested with different numbers of mini-batches. Cat dataset was trained with a variant of mini-batches of size {100, 256, 128} but the researchers found that the best performing model was the one with mini-batch =128. Figure 3 above shows the generated images from DenseGAN generator on Cat dataset corresponding to the input distribution.

Evaluation of Generative model.

Evaluating generative model is known to be a complicated and challenging problem [25]. While no standard evaluation metric exists, recent works have introduced many new metrics.

T. Salimans et al. [15] proposed the inception score that can be considered as a good evaluation metric, for GANs to show the discriminability of the generated images, but it assumes that the labels exist for the dataset. DenseGAN model is evaluated using two methods: first by showing samples from the proposed model and visually examining the quality of the generated samples and second by monitoring the loss function of DenseGAN generator during the training. Figure 7: below shows the cost function of the generator and discriminator during the training and shows the confusion of both cost functions on Cat dataset.

Results analysis.

In Figures 3 and 5, the images shown are drawn from the DenseGAN generator net after training. Thus, the images are by far better than the images generated by the existing methods; as a result, these images are believed to be at least competitive with the best generative models in the literature as well as these images are believed by the researchers to highlight the potential of the adversarial framework while the DCGAN suffers from mode collapse.

The good generated images are the only ones shown ; however, there are many other noisy images in the all-tested model. DCGAN framework is applied using the Cat dataset; the Cats are looking good using DCGAN and there is a noticeable lack of variety (lots of black cats) still some images are pretty good looking and they are in a higher resolution. Despite the difficulty of generating images of high-resolution level, the researchers observe that DenseGAN are able to learn about the objects that appear in the foreground.

One eye closed and one eye open or a weird noise are all samples as well as other samples ,like that of some cats that have heterochromia which is a phoniemeonia in some cats where each of the cats' eyes has a different color, and all appeared in the standard GAN framework that is based on a traditional Multi-layer perceptron.

The improvement of the generated images is logical and is not miraculous because of the DenseNet architecture which achieved a state of the art in classification tasks. The researchers note that the DenseGAN generator converge to those generated images takes about 6 to 8 hours with 200 epochs while DCGAN show better convergences in about 2 to 4 hours. The tardiness in DenseGAN converges is the architecture of DenseGAN that suffers from a large number of connections between layers and a concentration on operations before layers.

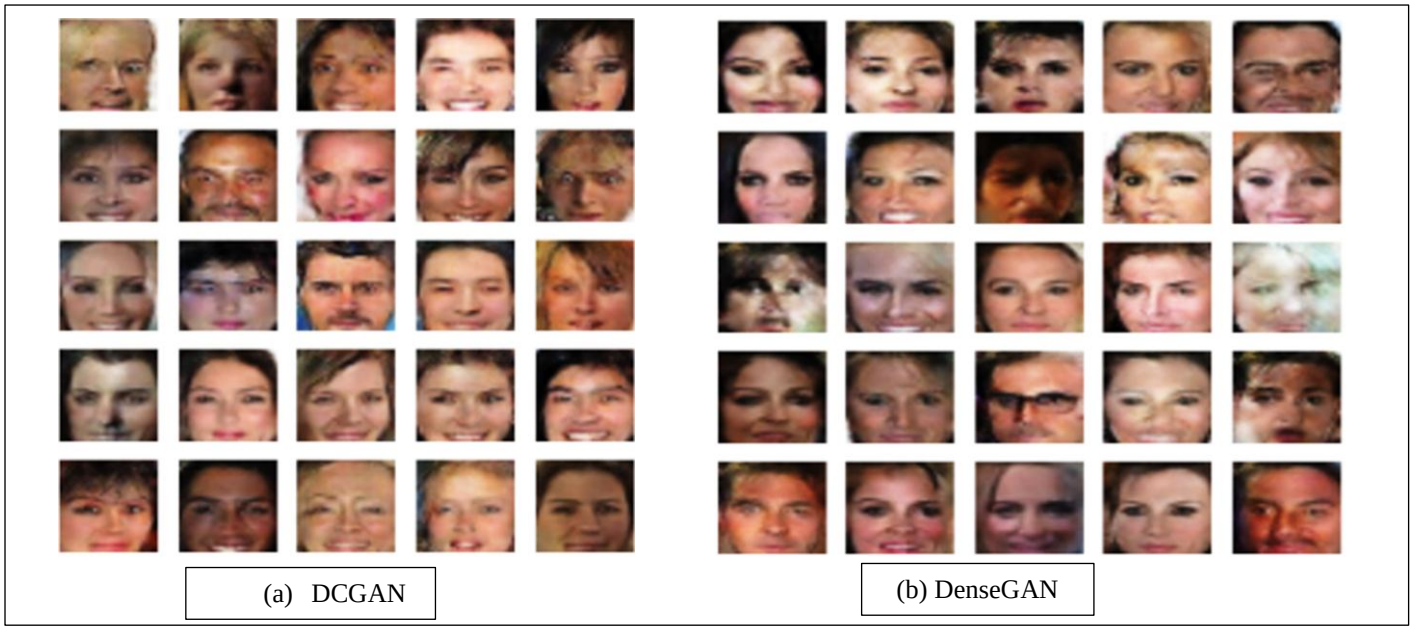

Figure 4: Generation from CelebA face images. Left (a):DCGAN. Right (b): DenseGAN. Although both models are far from being realistic, they show knowledge of the appearances of face such as their eye and mouth.

Finally, The researchers trained DenseGAN to generate high-reolution images on celebA . Compared with the datasets that were experimented so far, CelebA presented an extensively larger model, so modeling the real data distribution in a generative model would become very challenging and difficult. DenseGAN framework was trained to generate 64x64 RGB images.
The second approach was to compare it against DCGAN generated images.
Figure 4 above displays the generated samples using the DCGAN and the presented model DenseGAN. The DenseGAN model performance is the best comparable to the best existing algorithms for DGAN.
DCGAN is still better than DenseGAN in CelebA dataset that depends on its deep architecture where large CelebA dataset requires deeper layers. In addition,the 64x64 face images skipped many meaningful features but the DeseGAN did not respond to bigger image sizes. Although the generated images are far from being realistic, one advantage of the generator is that it captions few facial expressions in the generated face images.

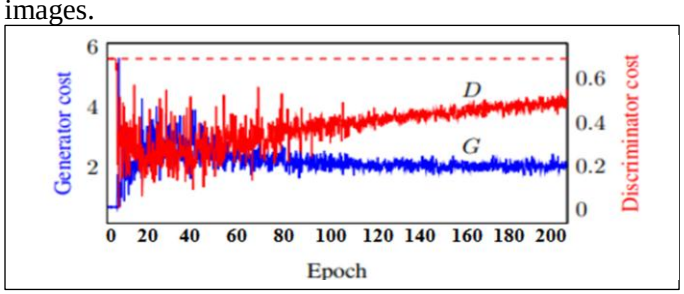

Figure 5:DenseGAN cost curves on Cat dataset. D trends toward maximumal confusion while G cost continually improves.

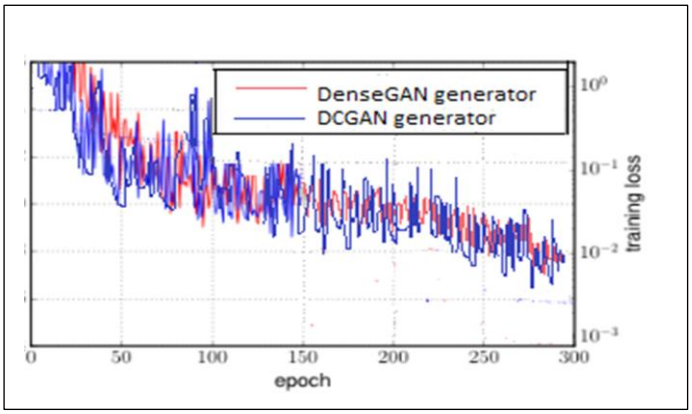

Figure 6: Generator Loss function compression. The DenseGAN generator showing less confu-sion verses the DCGAN generator.

6. Conclusion and Future Work

The paper presented an architectural system in adversarial nets, which is the framework for robust unsupervised learning. DenseGAN is based on Densely connected convolution network and generative adversarial network while keeping the main objective of standard generative adversarial network; it aims to let the discriminator assist the generator in his task by increasing the information flowing to it. The model is trained to generate images of static size 64x64x3and is

to yield an impressive result in generation task at Cat dataset, but is still far from achieving that in CelebA dataset. Figure 3 corroborates this conclusion with recognizable generated images. The presented work exhibits more stable behavior than standard GANs during training and is shown as at least competitive with DCGAN. DenseGAN shows an increase in image quality but it suffers from an increase in the network parameters.

To wrap up, three ideas for future work are presented which are:

- Using the DenseNet in the generator instead of the discriminator.
- Generating images of different sizes instead of making the generator static flexible
- Using the learnable features of the discriminator for supervised classification tasks.

Acknowledgment

Huge thanks go to Nile research center specially Mahmoud Serag for his valuable technical help and support. I would like to also thank Zhuang Liu et.al [8] for publishing the source code of his experiment and the DenseNet architecture.

References

[1] K. Simonyan and A. Zisserman.: *Very deep convolutional networks for large-scale image recognition*. CoRR, vol. abs/1409.1556, 2014.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton.: *Imagenet classification with deep convolutional neural networks*. In Advances in Neural Information Processing Systems 25, pp. 1106–1114, 2012.

[3] A. Radford, L. Metz, and S. Chintala.: *Unsupervised representation learning with deep convolutional generative adversarial networks*. In CoRR, abs/1511.06434v2, 2016.

[4] L. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio.: *Generative Adversarial Nets*. In Advances in Neural Information Processing Systems 27, pages 2672–2680. Curran Associates, Inc., 2014.

[5] E. L. Denton, S. Chintala, R. Fergus, et al.: *Deep generative image models using a laplacian pyramid of adversarial networks*. In Advances in neural information processing systems, pages 1486–1494, 2015.

[6] M.Mathieu,C.Couprie,Y.leCu.: *Deep multi-scale video prediction beyond mean square error*.arXiv preprint arXiv :1511.05440,2015.

[7] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do.: Semantic image inpainting with perceptual and contextual losses.: arXiv preprint arXiv:1607.07539, 2016.

[8] G.Huang, Z.Liu, L.Van der Maaten.: *Densely connected convolutional networks*. In CVPR,2017.

[9] M. Mirza and S.: Osindero.: *Conditional Generative Adversarial Nets*. arXiv:1411.1784 [cs, stat]. 2014.

[10] G. Hinton, S. Osindero, and Y.-W. Teh.: *A fast learning algorithm for deep belief nets*. Neural computation, 18(7):1527– 1554, 2006.

[11] Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol.: *Extracting and Composing Robust Features with Denoising Autoencoders*. In Proceedings of the 25th International Conference on Machine Learning, ICML '08, pages 1096–1103, New York, NY, USA, 2008.

[12] D. P. Kingma and M. Welling.: *Auto-Encoding Variational Bayes*. arXiv:1312.6114 [cs, stat],. arXiv: 1312.6114. Dec. 2013.

[13] A.Dosovitskiy, J.T.Springenberg, and T.Brox.: *Learning to generate chairs with convolutional neural networks*. In CVPR, 2015.

[14] J. Gauthier .: *Conditional generative adversarial nets for convolutional face generation* .Cs231n.stanford.edu Nov 2016.

[15] T. Salimans.: *Improved techniques for training gans*. arXiv preprint arXiv:1606.03498,2016.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros.*: Image to-image translation with conditional adversarial networks*. arXiv preprint arXiv:1611.07004, 2016.

[17] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do.: *Semantic image inpainting with perceptual and contextual losses*. arXiv preprint arXiv:1607.07539, 2016.

[18] A. Odena , C. Olah, and J. Shlens.: *Conditional image synthesis with auxiliary classifier gans*. arXiv preprint arXiv:1610.09585,2016.

[19] Jost Tobias Springenberg.: *Unsupervised and semi-supervised learning with categorical generative adversarial networks*. ArXiv preprint arXiv:1511.06390,2015.

[20] S.loffe ,C.Szegedy .*Batch normalization :. Accelerating deep network training by reducing internal covariate shift*. arXiv preprint arXiv:1502.03167v3,2015.

[21] D. Kingma and J. Ba. Adam.: *A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al.: *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*. arXivpreprintarXiv:1603.04467,2016.

[23] w.zhang, j.sun, and X.tang.*: Cat head Detection –How to Effectively Exploit Shape and Texture Features*. Proc. Of European Conf. computer vision, vol. 4, pp.802-816, 2008

[24] Z. Liu, P. Luo, X.Wang, and X. Tang.: Deep learning face attributes in the wild," in Proceedings of International Conference on Computer Vision (ICCV), December 2015.

[25]L.Theis,A .v.d.Oord, and M.Bethge.: *A note on the evaluation of generative models*.arXivpreprintarXiv:1511.01844,2015.