ISSN 1870-4069

Image Creation Using Generative Adversarial NetWorks (GAN) Applied to the Fashion Industry

Alejandro Acosta, Alberto Ochoa Zezzatti, Gustavo Delgado, José Mejía

Universidad Cuauhtemoc,
Maestria en Big Data
Mexico

alejandro.acosta@uabc.edu.mx

Abstract. The object manipulation of an image has become very popular lately. It is common to see applications about facial recognition or any other type of objects [2, 4, 19]. Technology allows people to achieve these tasks fast and in a really powerful way. In the fashion industry, there exists a lot of images due to photographers that can be taken and be analyzed to detect features and metadata. Deep learning has proven to be an excellent approach to the image recognition field by using convolutional neural networks (CNN). In this work, will be presented some image recognition and manipulation applied to the fashion industry by using CNN in order to create new objects from existing ones. Giving objects a certain level of realism on contours, boundaries, textures, colors, and shapes from the original "clothing" object.

Keywords: Deep learning, convolutional neural networks, fashion, clothing, image detection, image manipulation, machine learning, computer vision, generative adversarial networks.

1 Introduction

In the fashion industry, clothing is one of the most important subjects around it. The parades, photography, designers, stylists and much more, work to show to the world how amazing the new clothes are so people may look for that new skirt or new blouse in the stores the very next day to purchase. Technology makes everything better, and this includes the fashion industry. Image manipulation has become popular these days that is applied for practically every aspect of life.

Fashion designing is a particular area in the industry. It is known that this industry relies on designers and their work when designing the next big thing on clothes to present. In addition, it is known that graphic designers help, creating clothes on the computer from what the fashion designers pass to them.

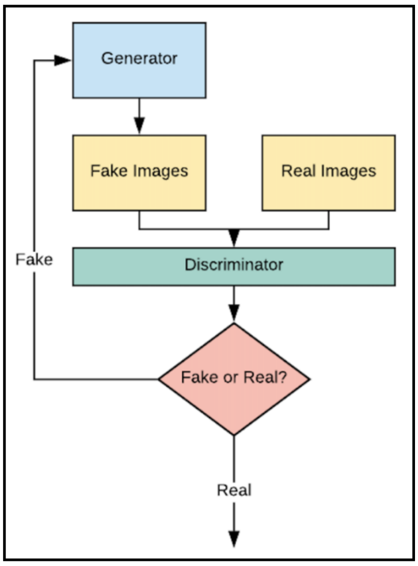


Fig. 1. Flowchart about GANs process.

An outstanding way to improve this process is to create new fashion designs based on a series of previous works created by fashion designers.

This will allow them to create new clothes that they could not even think about it or never occurs to them before [1].

Deep learning algorithms that learn from previous works and create new things are becoming powerful around the internet that demonstrate that it is possible for computers to create new things. Recent research shows that there is an approach that allows deep learning to create new things. The subject is about Generative Adversarial Networks (GAN). This kind of deep learning gives a new generative approach to image processing tasks in computer vision [5, 6].

[7] GANs are a kind of convolutional neural networks (CNN) presented to the world around 2014. These networks combine concepts about machine learning and Game Theory.

GAN model is also commonly known by DCGAN (Deep Convolutional Generative Adversarial Networks) could be understood a game between two agents: a Generator and a Discriminator. El objective of the Generator (G agent) is to generate the best information that it can from the real dataset (training step). On the other hand, the Discriminator (D agent) is in charge of classifying the real data and the fake data generated by the G. At the same time, the G is getting better on data creation since it is learning from the output that the D is discarding (classifying). The goal of this is that G can cheat D by generating realistic information that D can’t distinguish from fake or real information. Both algorithms help each other to get better and improve by themselves [9].

The information that the GAN algorithms will process and will be talked about in this paper are going to be images. Apparel images in specific. There is another characteristic around GANs and it is that are designed to reach a Nash equilibrium at which each player cannot reduce their cost without changing the other players parameters. This makes the algorithms to be prepared so they can create realistic images from any random noise data [10, 17]. The Ds are based on a particular type of algorithm named as Backpropagation. That is what allows to D to be recursive and iterative so it can learn from previous trainings and testing phases. This series of steps also will let the Discriminator to be refined and it is going to be better at distinguishing between fake and real images [11].

2 Related Work

As mentioned already. GAN algorithms are used around the internet for generation data and it is important to highlight those other related works and papers that have been a game-changer on the application that offers. Jun-Yan Zhu, and other colleagues of him, in their paper from 2017 titled “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks” [8], introduced their very famous CycleGan that provides an interesting way to reconstruct images from pair of data that is an input and an output. Quoting the paper mentioned above: “Image-to-image translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs.”

Also, the paper mentions how this algorithm works in order to accomplish the goal they are proposing. Quoting the paper again, “We present an approach for learning to translate an image from a source domain X to a target domain Y in the absence of paired examples. Our goal is to learn a mapping:

$$G: X \rightarrow Y, \tag{1}$$

such that the distribution of images from G(X) is indistinguishable from the distribution Y using an adversarial loss. Because this mapping is highly under-constrained, we couple it with an inverse mapping:

$$F: Y \rightarrow X, \tag{2}$$

and introduce a cycle consistency loss to enforce $F(G(X)) \approx X$ (and vice versa). Qualitative results are presented on several tasks where paired training data does not exist, including collection style transfer, object transfiguration, season transfer, photo enhancement, etc. Quantitative comparisons against several prior methods demonstrate the superiority of our approach”. This is one of the many works published that are so impressive in terms of impact and the application that can be applied.

The very first time that the GAN name appears, was because of Ian Godfellow [13] in 2014. He is the creator of this type of networks and every work related to GANs are thanks to Godfellow.

There are GANs that are based on Bayesian multivariable probabilistic models, where uses a set of random variables provided a graph. This way a generator can create images from not exactly a vector but from a cloud of data (distribution) [15].

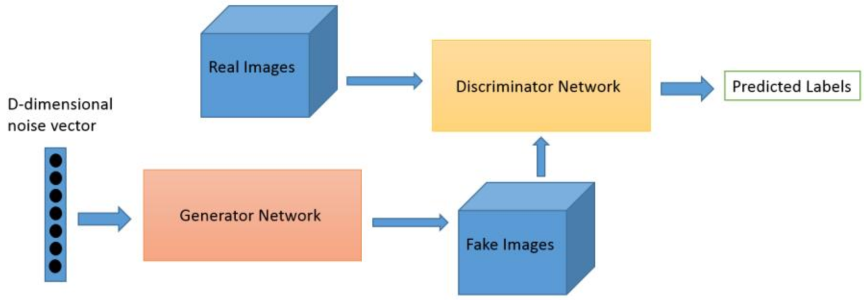


Fig. 2. Data Flow (source: https://www.oreilly.com/ideas/deep-convolutional-generative-adversarial-networks-with-tensorflow).

3 Proposed Approach

Creates a deep learning algorithm that will learn from fashion images and is going to be able to analyze and create new objects by recognizing patterns on the base images using Generative Adversarial Networks.

To address the approach to generate new objects given a dataset that will help to train GAN algorithms, GAN must be defined and describe how these networks work.

[6] GAN generates data distributions through the adversarial process. GAN gradually improves the quality of generations by the adversarial training process. GAN exhibits the following advantages: 1) GAN belongs to the type of non-parametric production-based modeling methods, which does not require prior approximate distributions of training data. 2) GAN works on the whole image and takes less time to generate samples by directly using global information.

What makes GAN so outstanding is its special structures. GAN is a deep adversarial framework consisting of a generative network named generator and a discriminatory network called discriminator.

The generator captures the data distributions, which wish to pass through the test of the discriminator, and the discriminator estimates the probability whether the sample is from true distributions.

By using GAN, it is required to ensure that the two agents are interacting at the same time, Generators and Discriminators. That is what is very interesting about this matter. A Generator is “The Artist”, a neural network that is trying to create pictures of clothes that look real. Discrimination is “The Critic”, a network that is examining clothing pictures to determine which ones look real or fake.

Showing a different diagram but this time as GAN data flow, it can be represented as the following fig. 2.

The discriminator it is consider a classification model or networks that uses unsupervised classification to determine whether the input image is real or not [16, 18, 19].

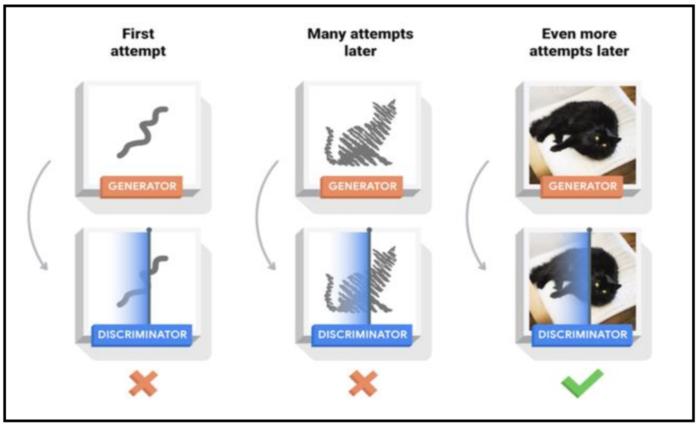


Fig. 3. Illustrative Generator and Discriminator process.

4 Materialization

The first part of the materialization of the work was to find a language and framework that allows handling GANs. Tensorflow and Python do the job excellently. There is an existing repository that is already built to generate images with the GAN approach. The repository belongs to Greg Surma, Github username is "gsurma" and the repository name is "image generator". This repository includes a full implementation of a GAN algorithm with Tensorflow and Python already. This is an existing work that was taken to be modified as needed for this project since the repository is open for exploring. Secondly, the dataset was replaced. Instead of using the dataset on the code repository, it was changed to use a collection of 500 t-shirts images. The project was adjusted for slowdown the epochs because the bigger the epochs amount are indicated, the bigger the computer power and resources will be needed. It is a good idea to run not too many epochs if you are on a computer without the necessary resources. The estimation is that 10 epochs can take up to 4 hours to complete. However, if there are just a few epochs, the results are not going to be as expected. Next, the training begins by using the t-shirt dataset. It is necessary and very important to crop all the images at the same size. For our purposes, all the images were resized to 200 x 200px.

The proposal is to use GAN algorithm based on Tensorflow framework that provides tools to accomplish these tasks. The main framework is based on the Python language that is also very much used for the community as mentioned above.

Looking at the big picture, the process is basically generating an image by the Generator and being analyzed by the Discriminator (Fig. 3). While the discriminator rejects the image, the generator will create a new one based on the rejection made. This is how the generator will learn. Once the discriminator validates and tags the generated image as valid, the image will be considered as realistic.

Each time that the discriminator rejects, the generator will start over again in a next iteration generating a new one to be analyzed. Each iteration, loop or repetition in this

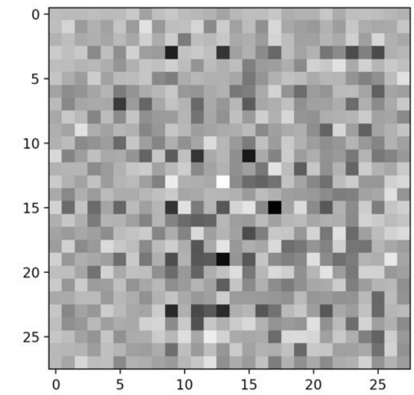


Fig. 4. Generated image by the Generator algorithm.

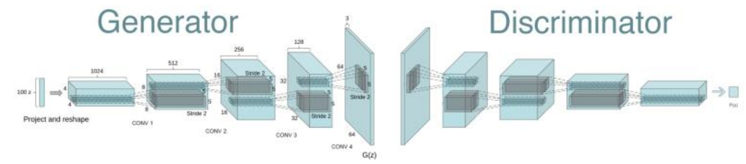


Fig. 5. An example architecture for generator and discriminator networks that uses convolutional layers to process visual information.

paper will be called as an "Epoch". The first part is to create the generator using python code and Tensorflow as primary deep learning framework. The generator uses tf.keras.layers.Conv2DTranspose (up sampling) layers function to produce an image from a seed. It starts with a dense layer that takes the input, then up samples several times until you reach the desired image size of whatever is specified. (Fig. 4).

The following part is for the discriminator to take the image generated from the random noise and determine if is real or fake. The discriminator is a Convolutional Neural Network (CNN-based) image classifier. The discriminator model is the one that will be trained in order to distinguish between real or fake images. To train the model, it was taken a specific in DataWorldTeam a Web site with many adequate datasets with several fashion images. Analyzing and explaining further the Generator and the Discriminator algorithms at the working phase, both can be explained with more details.

The Generator, as mentioned, it takes random noise as input and samples the output in order to fool the Discriminator that it's the real image. Once the Generator's output goes through the Discriminator, it is known that the Discriminator verdict whether it thinks that it was a real image or a fake one. This information has to be used to feed the Generator and perform backpropagation here. If the Discriminator identifies the Generator's output as real, it means that the Generator did a good job and it should be given a good feedback. However, if the Discriminator recognized that it was given a fake image, it means that the Generator failed and it should be given a negative feedback.

The following diagram shows how the generator and discriminator are built from the inside Fig 5.

In Fig. 5, it is explained how DCGAN are used for simple modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64-pixel image. Notably, no fully connected or pooling layers are used [12]. For the Discriminator, it gets both real images and fake ones and tries to tell whether they are legit or not. The system designers or people implementing the algorithm know whether they came from a dataset (reals) or from a generator (fakes). This information can be used to label them accordingly and perform a classic backpropagation allowing the Discriminator to learn over time and get better in distinguishing images.

When the Discriminator correctly classifies fake and real images, then it can be considered that it has a positive feedback on loss gradient. If it fails at classifying them, it can be said that has negative feedback on it. This process allows the algorithm to learn and get better at every turn. Quoting a paragraph of [12] for visualizing of the Discriminator features, “Previous work has demonstrated that supervised training of CNNs on large image datasets results in very powerful learned features (Zeiler & Fergus, 2014). Additionally, supervised CNNs trained on scene classification learn object detectors (Oquab et al., 2014).

We demonstrate that an unsupervised DCGAN trained on a large image dataset can also learn a hierarchy of features that are interesting. Using guided backpropagation as proposed by (Springenberg et al., 2014), we show in Fig. 5 (figure on that paper) that the features learnt by the discriminator activate on typical parts of a bedroom, like beds and windows. For comparison, in the same figure, we give a baseline for randomly initialized features that are not activated on anything that is semantically relevant or interesting.” This is very important to understand the work that the Discriminator does, because is not just detect if an image is real or not, but to learn that there are some common features in the dataset that needs to be considered as a MUST when the generator presents a generated image and it tries to pass on.

To make these previous paragraphs better understandable too, here is a definition of Backpropagation that has become a very important part of GANs. It is an algorithm used to efficiently train Artificial Neural Networks (ANN). Its main feature is that it is a Recursive algorithm that go back and learn to get better down the road until it reaches the goal that it was implemented for. In order for the Generator and Discriminator to learn, it is necessary to implement Loss Functions to introduce the backpropagation idea into this. A Loss Function is used for measuring the discrepancy between the two outputs. For the developed work on this paper, it is also important to consider a very important thing before doing the training of the algorithms. It is the Optimizers. The idea of this is to adopt optimizers in order to balance the Generator and Discriminator learning rates. Optimizers allow the algorithms to balance its rate for learning. This is

Table 1. Orthogonal array.

Variable Value								
A	B	C	D	E	F	G	H	Color
H	H	H	H	H	H	H	L	1
H	H	H	H	H	H	L	H	2
H	H	H	H	H	L	H	H	3
H	H	H	H	L	H	H	H	3

important since this approach is working with both at the same time and will not be a good idea if only one algorithm is actually learning and the other not.

4.1 Training

Now, for the training of the algorithms it is going to be set with 1 Epoch to test the functionality of the Generator and Discriminator. Results may vary based on the epochs that are specified. For the first epoch, as can be expected, the generated image will be discarded for it is a blurry image of pure noise data. As aside note, it is important to continually verify the optimizer and loss functions at each Epoch so it can balance both algorithms the better possible.

In order to be able similar, the most efficient arrangement of clothes in a dresser, we developed an atmosphere able to store the data of each one of the representing these clothes, this with the purpose of distributing of an optimal form to each one of the evaluated clothes. One of the most interesting characteristics observed in this experiment was the diversity of the cultural patterns established by each clothe with respect to their symbolic capital. The scenes structured associated with the agents cannot be reproduced in general, since they only represent a little while dice in the space and time of the different clothes.

These represent a unique form and innovating of adaptive behavior which solves a computational problem that it does not try to clustering the clothes only with a factor associated with his external appearance (attributes of each clothe), trying to solve a computational problem that involves a complex change between the existing representations.

The generated configurations can be metaphorically related to the knowledge of the behavior of a potential customer with respect to an optimization problem (to select culturally specify similar clothes, without being of the same kind [3]). The main experiment consisted of detailing each one of the clothes on a collection, with 500 agents, and one condition of unemployment of 50 époques, this allowed us to generate the best selection of each kind of clothes and their possible location in a Dresser, which was obtained after comparing the different cultural and social similarities from each clothe, and to evaluate with Multiple Matching Model each one of them [10].

The developed tool classified each one of the clothes pertaining to each kind, with different wardrobe for clothes that included identity and for clothes only with cultural identity, this permit identifies changes in the time respect at other clothes. The design of the experiment consists in an orthogonal array test, with the interactions between the

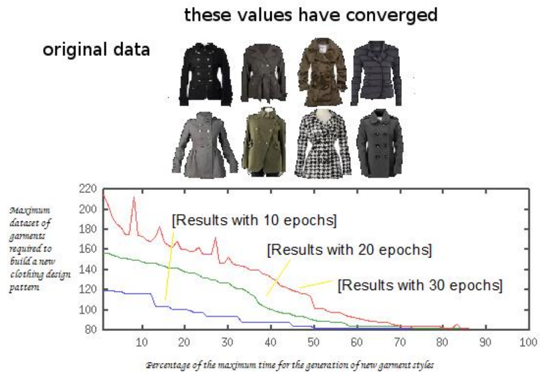


Fig. 6. Representation of our GAN to design new models of clothes.

Table 2. Comparison between epochs.

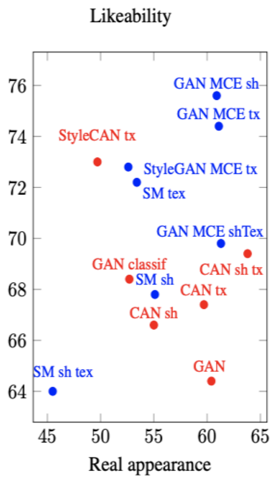
	1 Epochs	10 Epochs	50 Epochs	100 Epochs
Image Size	128	128	128	128
Noise Size	100	100	100	100
Batch Size	64	64	64	64

variables: emotional control, ability to fight, intelligence, agility, force, resistance, social leadership, and speed. These variables are studied in a range of color (1 to 64). The orthogonal array is L-N (2**8), in other words, 8 factors in N executions, N is defined by the combination of possible values of the 8 variables the possible range of color (see Table 1).

According to the results obtained for the time of accommodation in the transport must be performed to accommodate the time of unloading in the following design. On the basis of the results obtained in the experiment, the similarity average between clothes, it was found that the average is 4.0625, this means that you have to improve in the management of the use of the combination colors and standardize according to the weather.

Other factors that affect are the turn as well as the use of accessories, such as: bracelets, necklaces, pins and charms, an implementation in order to resolve this problem is the use of fashion visualization.

As can be seen to hold this type of materials is necessary to use merchandise direct and indirect. The merchandise indirect secure ground, it gives resistance to lateral and longitudinal movement of the clothes. While direct ties anchored to manikins in the dresser. To perform this process, it is necessary to count with the correct dimensions and the distance from the street and the height of these, cables or chains, taking into account the material of these materials.



Method/Human Method	over-all	shape nov.	shape comp.	tex. nov.	tex. comp.	real fake
DCGAN MCE shape	**3.78**	**3.58**	**3.57**	**3.64**	**3.57**	**60.9**
DCGAN MCE tex	**3.72**	**3.57**	3.52	**3.61**	**3.58**	**61.1**
StyleCAN tex	3.65	3.37	3.31	3.44	3.21	49.7
StackGAN	3.62	3.45	3.38	3.43	3.33	51.9
StyleGAN MCE tex	3.61	3.38	3.29	3.50	3.37	53.4
DCGAN SM tx	3.60	3.43	3.42	3.46	3.41	52.6
StackGAN MCE tex	3.59	3.36	3.31	3.44	3.28	55.9
StyleGAN	3.59	3.28	3.21	3.27	3.15	47.2
StackGAN CAN sh	3.51	3.56	**3.56**	3.58	3.40	50.7
DCGAN MCE shTex	3.49	3.40	3.24	3.40	3.31	**61.3**
StackGAN CAN tex	3.48	**3.57**	3.54	3.55	3.50	48.4
DCGAN CAN shTex	3.47	3.28	3.18	3.33	3.16	**63.8**
StackGAN MCE sh	3.45	3.27	3.16	3.28	3.12	60.4
StackGAN CAN shTex	3.42	3.37	3.32	3.44	3.32	49.5
DCGAN classif	3.42	3.32	3.32	3.37	3.29	52.7
DCGAN SM sh	3.39	3.27	3.12	3.30	3.23	55.1
DCGAN CAN tex	3.37	3.23	3.12	3.35	3.09	59.7
DCGAN CAN sh	3.33	3.28	3.16	3.27	3.12	55.0
StackGAN MCE shtex	3.30	3.25	3.28	3.31	3.27	41.5
DCGAN	3.22	2.95	2.78	3.24	2.83	60.4
DCGAN SM shTex	3.20	3.10	3.00	3.13	3.1	45.5

Fig. 7. Analysis of various studies around different types of GAN.

Fig. 8. Representation of Tall Models built with a GAN using 1000 diverse faces on a dataset of DataWorldTeam.

4.2 Results

All the very first Epoch results are expected to be images with no sense at all since the generator is basically constructing "something" that it doesn't have any reference on either what it is building is right or not. Until the discriminator tell, it is obviously in the next steps, see Figure 6.

4.3 Comparison between Epochs

Here it can be visualized a comparison rates between the different Epochs and the adjustments on every specs around the optimizer and loss functions. We show in Table 2 these features.

4.4 Analysis of GANs

In the paper "Design Inspiration from Generative Networks" [14], the authors conducted a study where were mixed 2 different generations. 500 total images picked randomly from 5 best models and 300 real down-sampled images from a RTW dataset. They asked if the images were real or generated to about 45 participants who rated 20 images each in average. They obtain 20% of the generations thought to be real, and 21.5% of the original dataset images were considered to be generated.

In this study, it is very notable that DCGAN networks are very reliable to deliver good results than the rest of GANs.

5 Conclusion

The generative process offered by GANs method has the advantage that uses a simple approximation of sampling from an unknown distribution, such as the current presented by the Generator from any noise seed data. Unfortunately, GAN requires a lot of computing power to run a good number of epochs to ensure generating very good output images. That means the algorithm will not reach its optimal level because it requires many epochs that used in this work.

Another disadvantage is that both algorithms, Generative and Discrimination are not synchronized due to the initial conditions. Discriminator has a better start since it has a standpoint because it can compare with the actual dataset. Therefore, thinking in game theory, discrimination is wining.

Following this logic, the Generator requires more time to create better copies of the fake images to reach the point when it can beat the Discriminator [7]. In future research it is possible to make diverse tall models with different aspects to modelling clothes in a specific society as Guam, Nepal or Timor-Leste with average size of 1.57 to female or 1.64 to male, as is shown in figure 8.

References

1. Liu, Z., Yan, S., Luo, P., Wang, X., Tang, X.: Fashion landmark detection in the wild. Lecture Notes in Computer Science, pp. 229–245 (2016)
2. Lao, B., Jagadeesh, K.: Convolutional neural networks for fashion classification and object detection. Semantic Scholar (2015)
3. Yu, W., Liang, X., Gong, K., Jiang, C., Xiao, N., Lin, L., Yat-Sen, S.: University, DarkMatter AI research. Layout-Graph Reasoning for Fashion Landmark Detection (2019)

4. Cetinic, E., Lipic, T., Grgic, S.: A deep learning perspective on beauty, sentiment and remembrance of art. In: Institute of Electrical and Electronics Engineers (IEEE), pp. 1–1 (2019)
5. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Visual Geometry Group, Department of Engineering Science, University of Oxford (2015)
6. Yangjie Cao, Li-Li Jia, Yong-Xia Chen, Nan Lin, Cong Yang, Bo Zhang, Zhi Liu, Xue-Xiang Li, Honghua Dai: Recent advances of generative adversarial networks in computer vision. In: Institute of Electrical and Electronics Engineers (IEEE), 7, pp. 14985–15006 (2019)
7. Atenas, F., Sanhueza, F., Valenzuela, C.: Redes neuronales adversarias convolucionales para generación de imágenes (2017)
8. Jun-Yan, Z., Park, T., Isola, P.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks efros. Berkeley AI Research (BAIR) laboratory, UC Berkeley (2015)
9. Li, F.F., Johnson, J., Yeung, S.: CS231n: Convolutional neural networks for visual recognition (2017)
10. Chernov, A.V.: On some approaches to find Nash equilibrium in concave games. Automation and Remote Control, 80(5), pp. 964–988 (2019)
11. Kanezaki, A.: Unsupervised image segmentation by backpropagation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 760–769 (2018)
12. Radford, A., Metz, L.: Unsupervised representation learning with deep convolutional generative adversarial networks (2016)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Annual Conference on Neural Information Processing Systems (NIPS) (2014)
14. Sbai, O., Elhoseiny, M., Bordes, A., LeCun, Y., Couprie, C.: Design inspiration from generative networks. Cornell University (2018)
15. Reyes, M.A., Sánchez, L.A., Mote, R.E.: Una propuesta genética y bayesiana para resolver problemas de clasificación en aplicaciones médicas. Research in Computing Science (2016)
16. Figueredo-Avila, G.A.: Clasificación de la manzana royal gala usando visión artificial y redes neuronales artificiales. Research in Computing Science 114(1), pp. 23−32 (2016)
17. Ortiz-Rangel, E., Mejía-Lavalle, M., Sossa-Azuela, H.: Using pulsed neural networks to improve filtering of images contaminated with gaussian noise. Computacion y Sistemas, 21(2), pp. 381−395 (2016)
18. Zecua, E., Caballero, I., Martínez-Carranza, J., Reyes, C.A.: Clasificación de estímulos visuales para control de drones. Research in Computing Science 114(1), pp. 201−212 (2016)
19. Mejía-Lavalle, M., Lux, M., Pérez, C., Martínez, A.: Digital images text detection as strategy to improve content-based image retrieval. Research in Computing Science (2016)