International Journal of Hybrid Innovation Technologies
Vol.3, No.1 (2023), pp.1-20
http://dx.doi.org/10.21742/ijhit.2653-309X.2023.3.1.01

Synthesis of the Convolution Neural Networks Structures for Images Classification

Nataliya Boyko

Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine
Nataliya.I.Boyko@lpnu.ua

Abstract

In this work would be analyzed Generative Adversarial Networks (GANs) and how they work on the example of an artist's style translation to landscape photographs. This architecture consists of a generator and discriminator which oppose each other in the minimax game. The goal of the discriminator is to learn original data distribution. At this time, discriminator evaluates how well a generator performs in reproducing original style. Since in this paper, we will work with photos and pictures, convolutional neural networks will be used as a generator and a discriminator. Also, would be considered a modification to GAN architecture, which can notably improve result: another pair of discriminator and generator will be added, and a new cycle loss would be introduced. A comparative analysis with other methods such as Neural Style Transfer will also be performed. The methods of comparative analysis of Neural Style Transfer and Pix2Pix are presented in the work. Neural Style Transfer approach is more stable and with a well-chosen photo can give better results, but is more limited. Pix2Pix, is another modification to GAN architecture with the only difference that now our data is images and with additional loss that is calculated as some distance between generated and original pictures. The article presents the essence of the architecture of the PatchGAN generator, where the first part of the network takes the original image and encodes it, reducing the dimension, and the other takes this result and tries to reproduce the desired image. The goal of the work is to learn the artist's style, no additional processing was applied except normalization to [-1, 1] before feeding to the neural network (because as activation is used hyperbolic tangent) and resizing to (256, 256, 3). The same actions were applied to the second part of the dataset: photos of nature. Data networks with Ivan Marchuk's paintings and photos of nature were trained with the help of Google Colab. The Adam was the optimizer. The results were compared with the Style Transfer model.

Keywords: *Generative adversarial network, PatchGAN, CNN, CycleGAN, Artificial neural network, Convolution neural network Image generation, Style transfer*

1. Introduction

The importance of Generative Adversarial Networks is hard to overestimate. They can be applied nearly in every area. In medicine, they are used to denoise scans and detect anomalies. Also, 3D designers can use them to generate 3D-objects, music, pictures that can

Article Info:
Received (May, 1 2023), Review Result (June 5, 2023), Accepted (July 15, 2023)

notably facilitate the development of movies and games. Additionally, people use them to generate text descriptions to the images, process images, enhance them, and many more.

In this paper considered how Generative Adversarial Networks work on the example of generating images with a distinct style. Also, modifications to the previous architecture were introduced to adapt to the problem of Image-to-Image translation.

The paper focuses on fine-grained image-to-image transformation for visual recognition, that is, image transformation using a fine-grained category according to a certain style to synthesize new images that preserve the identity of the input image. In order to use the new samples, it is necessary to carry out a number of transformations using the proposed transformation algorithms. In the study, special attention was paid to the optimization of the generator and the discriminator using the stochastic gradient descent algorithm.

The paper presents two approaches - Neural Style Transfer [2] and direct conversion of a photo into a photo (Pix2Pix). The research itself is aimed at proving the main aspects of these approaches and the identified advantages of their application.

Research is conducted on a descriptor that is similar to the architecture of the CNN network, called PatchGAN [12].

To achieve the ultimate goal of recognition, the created images do not necessarily have a high visual quality. However, they should be correctly classified even by detailed generation scenarios. Achieving this goal is difficult because images from different fine-grained categories may have only minor differences. This research is aimed at generating images according to a certain "style".

The goal of the study was to analyze how GANs can be used in the task of Image-to-Image transformations, compare with other methods and use in practice to generate images with a particular style. The following tasks were set to achieve this.

1. Research on how GANs work and their mathematical proof.
2. Investigate what modifications to the architecture are made to train them in the images.
3. Analyze other methods that can solve the problem of generating images with style.
4. Collect data for training and form a dataset.
5. Train GAN and analyze results.

The object of the study was artificial neural networks (GAN) that are capable of generating new data with the same characteristics as in the given data. It was decided to focus on the problem of Image-to-Image translation: style transfer.

The results can be used to generate images (augmenting datasets, easing of work for graphic designers). For writing this work, was analyzed scientific literature on similar topics.

2. Literature analysis

Due to the popularity of the GAN model, there are many studies on Neural Style Transfer [2] and direct conversion of a photo into a photo (Pix2Pix) approach. First of all, they explain the basic theory and discuss the advantages and disadvantages of these algorithms.

The mentioned works describe the performance of conventional GANs [3], which are applied to image-to-image transformation tasks [18], or super-resolution [6][9] and general-purpose image-to-image translation tasks [10][19]. Most of the described models deal with scenarios where the input and output images have a pixel-to-pixel spatial correspondence and are generally not applicable to the tasks of image generation according to a certain "style", which are discussed in detail in our work.

In recent works, attempts were made to generate images based on the works of a famous author, having studied his "style". However, the existing works are aimed primarily at the synthesis of data of high visual quality [7][9][18]. In these works, there is no image development algorithm that identically reproduces the features of the original images, especially with the use of the fine-grained image transformation scenario, which is the main goal of the study.

The article [18] presents GAN-based techniques used in a specific application, i.e., the task of image-to-image transformation based on adversarial learning. GAN offers a new concept of image-to-image transformation through adversarial learning. In recent years [10], many methods based on adversarial learning have been proposed and impressive results have been achieved. Related reviews mainly focus on the basic GAN model and its general variants. Isola et al [19] aimed to provide an overview of methods based on adversarial learning, focusing on the image-to-image transformation scenario.

A brief overview of the basic GAN is given in [18]. In addition, approaches related to our research are described in [19], which are divided into competitive style transfer and competitive image restoration, such as super-resolution, image painting, and rain removal. Network architectures of generative models and loss functions are presented and discussed in detail in.

Existing approaches to image-to-image conversion mainly focus on data synthesis. Therefore, generating images with correct identification labels is still challenging and underexplored.

However, in the presented study, the author provides an introduction to the GAN architecture and provides an example of using the network to create styled photos. In the work, a modification of the CNN architecture is proposed in order to adapt it to the solution of the outlined problem.

3. GAN architecture

GAN consists of two connected artificial neural networks – generator and discriminator. The generator will learn distribution p_g over data x and can be represented as $G(z, \theta_g)$ where z is data from latent space $p_z(z)$ and G is differentiative function represented by some multilayer perceptron with parameters θ_g, discriminator is also represented by multilayer perceptron $D(x, \theta_x)$, where x is data from original distribution p data, that we want a generator to learn, θ_x– parameters of the model, as a result of $D(x)$ will be received a scalar value which represents the confidence of discriminator that data came from original distribution p_{data} rather than p_g. So we train D to maximize the probability of correctly predicting the distribution of data – $log(D(x)) + log(1-D(G(z)))$and G to minimize $log(1-D(G(z)))$. In other words, we can say that generator and discriminator play in the following minimax game (Formula 1) [18].

$$min_G maxDV(G, D) = Ex{\sim}p_{data}(x)[log(D(x))] + Ez{\sim}p_z(z)[log(1 - D(G(z)))] \quad (1)$$

But, when this cost function is used in practice, the generator cannot learn properly. It takes place, because, on first iterations, the discriminator will with high accuracy predict from which distribution data originally came from, because the generator did not have enough time to learn, so data is contrasting. And because of that, the generator will not receive sufficient gradient for learning. To deal with this, the generator will try to maximize $log(D(G(z)))$ instead, which will help to get larger gradients. What can be visualized as follows (Fig. 1):

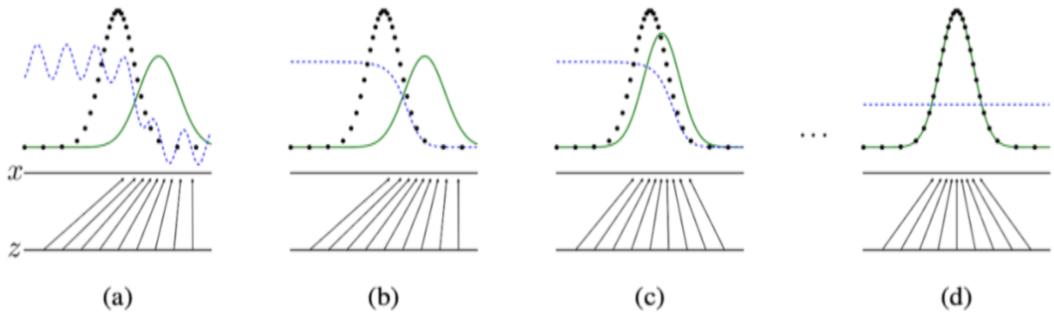

Figure 1. Visualization of the training process

Where a blue dashed line indicates *D*, green p_g, black p_{data}, and p_z, underneath them. In (a) we can see a situation where training has just begun, in (b) parameters of the discriminator are updated, on (c) generator has been updated gradually making such updates, ideally after a certain number of iterations the algorithm should enter the Nash equilibrium depicted in (d).

Generator and Discriminator can be optimized using stochastic gradient descent. Then the training process will look like this [1][2]:

for number of training iterations **do**

for *k steps* **do**

- Sample minibatch of *m* noise samples $\{z^{(1)}, \ldots , z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of *m* examples $\{x^{(1)}, \ldots , x^{(m)}\}$ from data generating distribution $p_{data}(x)$.
- Update the discriminator by ascending its stochastic gradient (Formula 2):

$$\Delta_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[logD\left(x^{(i)}\right) + \log(1 - D\left(G\left(z^{(i)}\right)\right)) \right] \quad (2)$$

end for

- Sample minibatch of *m* noise samples $\{z^{(1)}, \ldots , z^{(m)}\}$from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient (Formula 3):

$$\Delta_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \log(D\left(G\left(z^{(i)}\right)\right)) \quad (3)$$

end for

To proof, that the global optimum achieves when $p_g=p_{data}$, firstly we will consider optimal discriminator to any given generator, which will be (Formulas 4-5):

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)} \quad (4)$$

Proof:

$$min_G max_D V(G, D) = Ex{\sim}p_{data}(x)[log(D(x))] + Ez{\sim}p_z(z)[log(1 - D(G(z)))] =$$

$$= Ex{\sim}p_{data}(x)[log(D(x))] + Ex{\sim}p_g(x)[log(1 - D(x))] = p_{data}(x)log(D(x)) + p_g(x)\, log(1 - D(G(z))dx, \quad (5)$$

Then,

$$y = a \cdot \log(y) + b \cdot \log(1-y)$$
$$y' = \frac{a}{y} - \frac{b}{1-y}$$
$$\frac{a}{y^*} = \frac{b}{1-y^*}$$
$$\frac{1-y^*}{y^*} = \frac{b}{a}$$
$$\frac{1}{y^*} = \frac{a+b}{a}$$
$$y^* = \frac{a}{a+b}$$

Substituting the result, we obtain (Formula 6)

$$max_D V(G,D) = E_{x\sim p_{data(x)}}\left[\log\left(D_G^*(x)\right)\right] + E_{z\sim p_z}\left[\log\left(1 - D_G^*\left(G(z)\right)\right)\right] =$$
$$E_{x\sim p_{data(x)}}\left[\log\left(D_G^*(x)\right)\right] + E_{x\sim p_g}\left[\log\left(1 - D_G^*\left(G(x)\right)\right)\right] =$$
$$E_{x\sim p_{data(x)}}\left[\log\frac{\mathrm{p_{data}(x)}}{\mathrm{p_{data}(x)+p_g(x)}}\right] + E_{x\sim p_g}\left[\log\frac{\mathrm{p_g(x)}}{\mathrm{p_{data}(x)+p_g(x)}}\right], \tag{6}$$

Now, we can demonstrate that *C(G)* reaches a global minimum at p_g=p_{data}. From 2 it is obvious that at $p_g = p_{data} D_G^*(x) = 0{,}5$ than, substituting this value into the 3, and considering p_g=p_{data}:

$$C(G) = \int_x (p_{data} log\frac{1}{2} + p_{g(x)} log\frac{1}{2})dx = -log2\int_x (p_{data}(x)dx - log2\int_x p_g(x)dx = -2 \cdot log2$$
$$= -log4$$

Proof:

$$C(G) = \int_x (log2 - log2)p_{data}(x) + p_{data}(x)\log(\frac{p_{data}(x)}{p_g(x) + p_{data}(x)}) + (log2 - log2)p_{g(x)}$$
$$+ p_g(x)\log(\frac{p_g(x)}{p_g(x) + p_{data}(x)})$$

$$C(G) = -log2\int_x p_g(x) + p_{data}(x)dx + \int_x p_{data}(x)\,(\log 2 + \log(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)}) + p_g(x)$$
$$+ \left(\log 2 + \log\left(\frac{p_g(x)}{p_{data}(x) + p_g(x)}\right)\right) - log4 + \int_x p_{data}(x)\log(\frac{2 * p_{data}(x)}{p_{data}(x) + p_g(x)})$$
$$+ p_g(x)\log\left(\frac{2 * p_g(x)}{p_{data}(x) + p_g(x)}\right) =$$
$$-log4 + \int_x p_{data}(x)\log(\frac{2 * p_{data}(x)}{p_{data}(x) + p_g(x)}) + p_g(x)\log\left(\frac{2 * p_g(x)}{p_{data}(x) + p_g(x)}\right) =$$
$$-log4 + \int_x p_{data}(x)\log(\frac{p_{data}(x)}{\frac{\left(p_{data}(x) + p_g(x)\right)}{2}}) + p_g(x)\log\left(\frac{p_g(x)}{(p_{data}(x) + p_g(x))/2}\right) =$$
$$-log4 + KL\left(p_{data}\left\|\frac{p_{data} + p_g}{2}\right.\right) + KL\left(p_g\left\|\frac{p_{data} + p_g}{2}\right.\right) =$$
$$-log4 + 2 \cdot JSD(p_{data}|p_g)$$

And from the definition of the Jensen-Shannon divergence we know, that between two identical distributions it is equal to 0.

4. Optimization of architecture

However, to effectively use GANs for Image-to-Image translation to the previously shown architecture [Figure 2]:

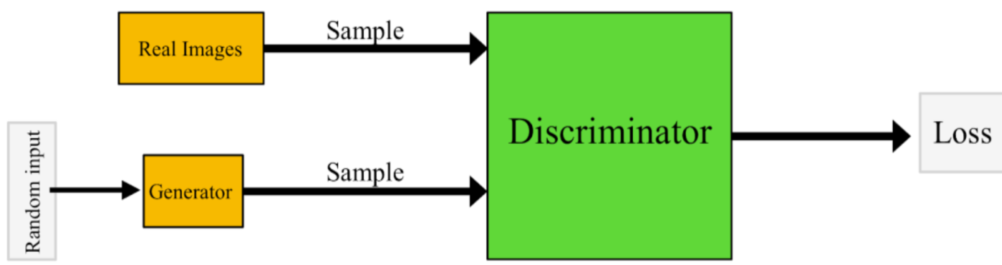

Figure 2. Schematic representation of GAN architecture

Some changes need to be made [Figure 3] [1]:

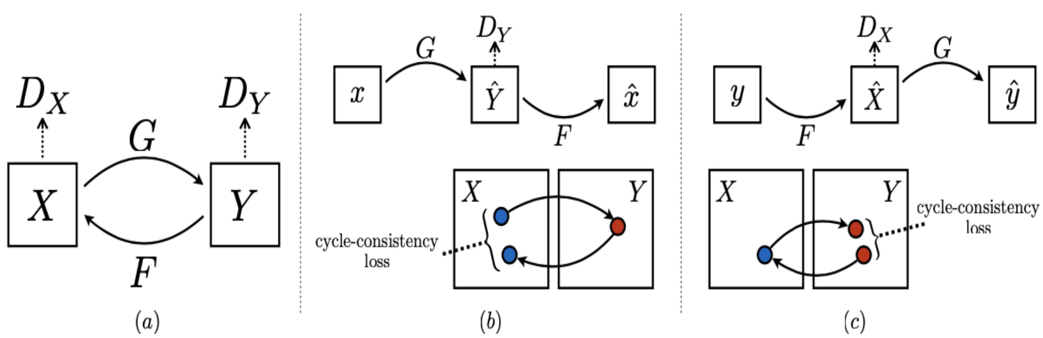

Figure 3. Schematic representation of CycleGAN architecture

As shown in (a), now model consists of pair of generators and discriminators, generator G objective is to transform images from area X to Y: $G(X)\rightarrow Y$, and discriminator D_Y evaluates it, the same works in adverse – $F(Y)\rightarrow X$, and D_x evaluates it. The corresponding loss functions will look like this (Formulas 7-8) [1]:

$$min_G max_{D_y} L_{GAN}(G, D_y, X, Y) = Ey{\sim}pdata(y)[logDy(y)] + Ex{\sim}pdata(x)[log(1 - Dy(G(x)))] \quad (7)$$

$$min_F max_{D_x} L_{GAN}(F, D_x, X, Y) = Ex{\sim}pdata(x)[logDx(x)] + Ey{\sim}pdata(y)[log(1 - Dx(F(y)))] \quad (8)$$

In theory, the model could learn only using these two functions but, to improve the result we can make this process cyclic, this means that by feeding to G some image y and transferring the result to F, we should get the original image y.

To achieve that, another loss function was constructed (Formulas 9-11):

$$Lcyc(G, F) = Ex{\sim}pdata(x)[||F(G(x))-x||1] + Ey{\sim}pdata(y)[||G(F(y))-y||1] \quad (9)$$

The result will look like:

$$L(G, F, Dx,Dy) = L_{GAN}(G, Dy,X,Y) + L_{GAN}(G, Dx,X,Y) + Lcyc(G, F), \tag{10}$$

where – hyperparameter that allows controlling importance of the newly added loss.

And we will be solving

$$G^*, F^* = arg\ min_{G, F}\ maxDy, Dx\ L(G, F, Dx, Dy) \tag{11}$$

5. Comparison with other methods

Several other methods deal with this problem: Neural Style Transfer [2], and Pix2Pix [3].

In the first method – Neural Style Transfer, we take a deep pre-trained neural network, keeping only convolutional layers and pass two images: “style” image and image we want to be stylized, which parameters will be optimized.

The first loss is represented as a sum of two different losses: first measures as a squared sum of the difference between activations from the last layer of “style” and original images. The second loss is measured as the difference of Gram matrices, which is calculated for both images as a correlation between activations on different channels of layers.

This method is more stable and with a well-chosen “style” image can give better results, but on the other hand is limited, because we can gather knowledge about style from only one image. Another advantage of this method is that the image can have any shape with the only constraint that the shape of the “style” image should be equal to the shape of the original image. It is possible because we only use convolution layers.

The second method, Pix2Pix, is another modification to GAN architecture with the only difference that now our data is images and with additional loss that is calculated as some distance between generated and original pictures.

But this method requires a large paired dataset, which in case of style transferring from the artist’s pictures cannot be assembled [4][5][6].

6. Implementation

As a discriminator is used CNN with the following architecture [Figure 4]:

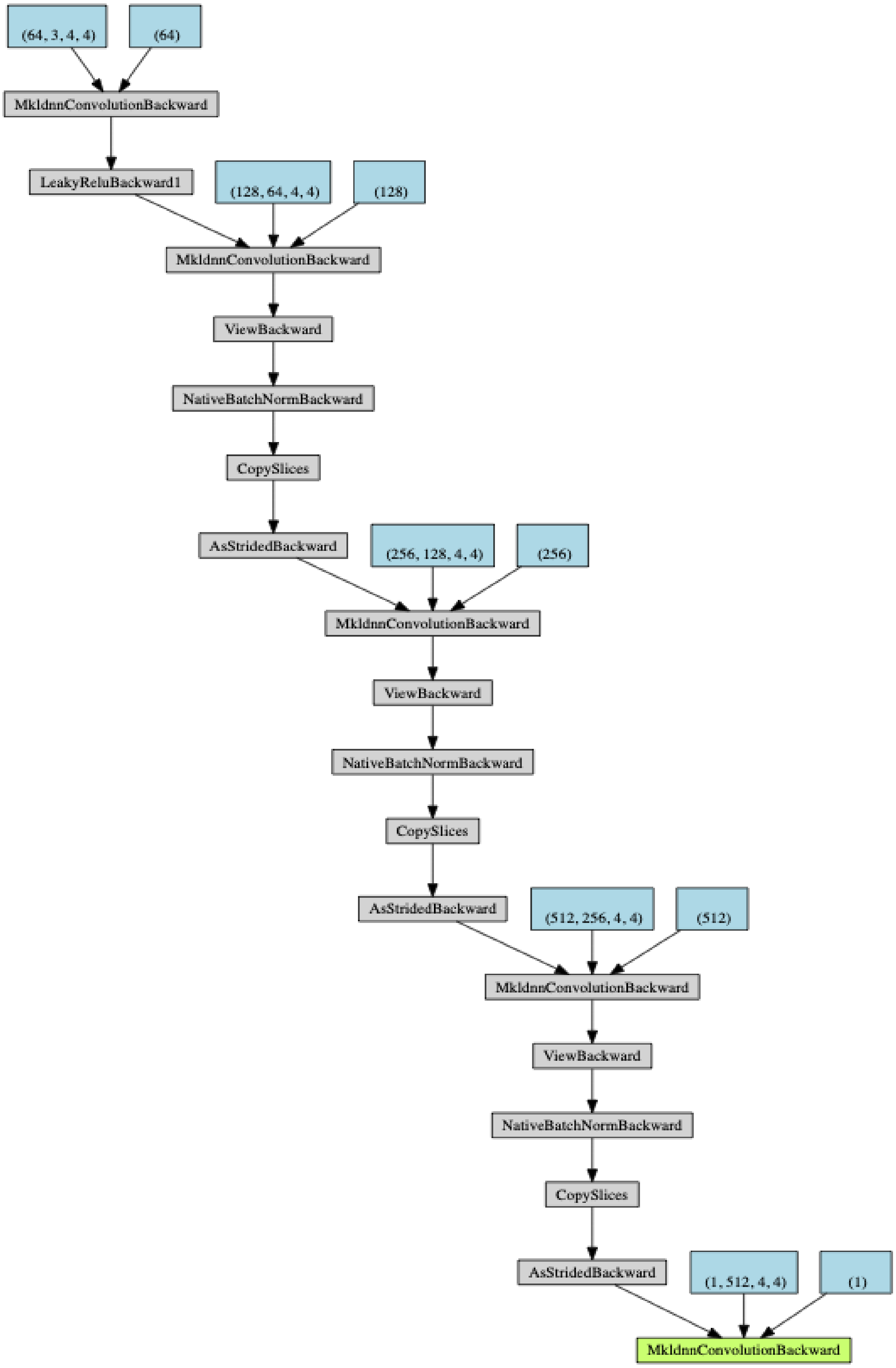

Figure 4. Discriminator architecture

Such architecture of discriminator is called PatchGAN [7]. It is an ordinary convolution neural network classifier with the only difference that it outputs not one scalar value for the whole image but many for different parts of the image. It can be done by removing all fully connected layers and dividing output from convolution layers into equal batches and calculate scores for each of them. This modification will help to handle some problem areas modifying only them and not affecting the whole image. Also on each layer is used Instance Normalization, which makes the model more stable.

The core of the generator is similar to autoencoders [8], the first part of the network takes an image and thus encodes it reducing the dimension, the second part takes the result and tries to reproduce the original image or in our case, an image with a new style [Figure 5].

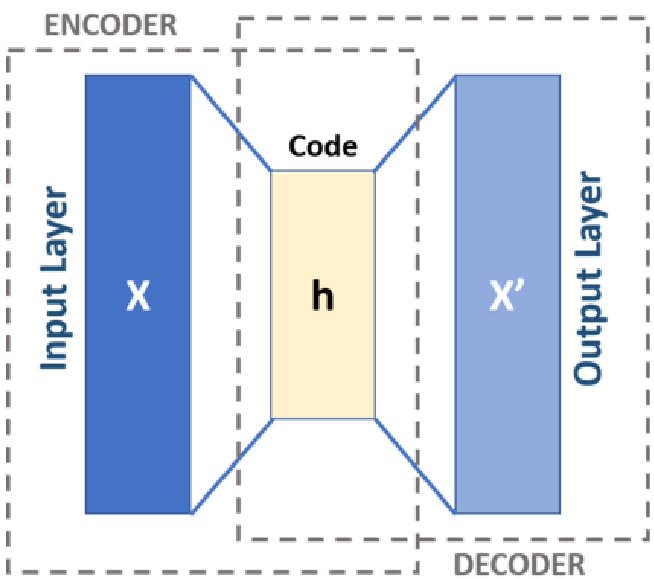

Figure 5. Autoencoder architecture

As a basis of the generator was chosen ResNet [9] modified to the current task [10]. The total number of parameters for the models turned out as follows: 11.378 million in generator and 2.765 million in discriminator.

7. Experiments

The next step was to choose what author's style would be transferred, for this was considered couple Ukrainian artists: Wassily Kandinsky [11], Anatolii Kryvolap [12], Vasyl Lopata [13], Kazemyr Malevych [14], Ivan Marchuk [15] and Mykola Pymonenko [16]. Eventually, it was decided to train on Marchuk's pictures because he had more paintings that allowed us to collect a relatively large dataset (552 pictures). Because the goal of the work is to learn the artist's style, no additional processing was applied except normalization to [-1, 1] before feeding to the neural network (because as activation is used hyperbolic tangent) and resizing to (256, 256, 3). The same actions were applied to the second part of the dataset: photos of nature.

Class A – Marchuk's pictures [Figure 6]:

Figure 6. Examples of Marchuk`s pictures

Class B – Photos [Figure 7]:

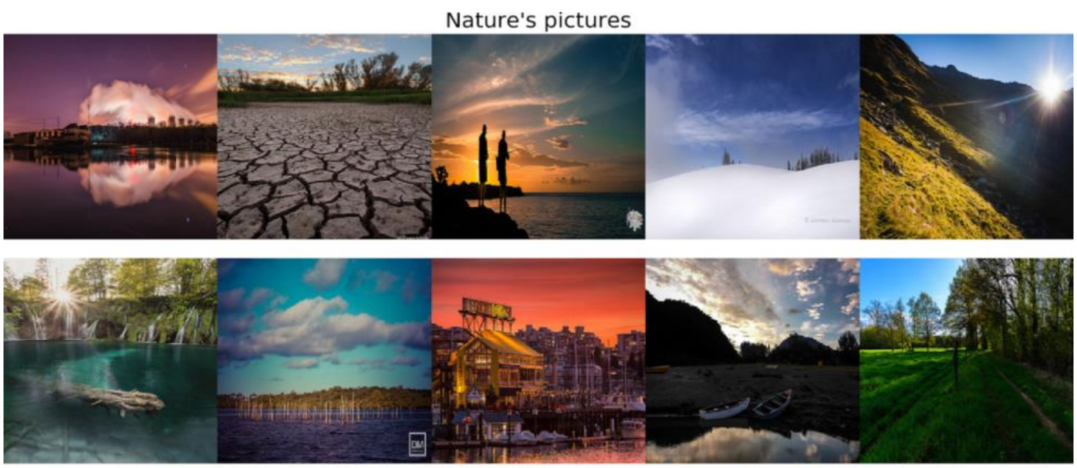

Figure 7. Photos example

8. Results and discussion

The training was performed on Google Colab, it lasted 14 epochs and took about six hours. Adam [17] was used as an optimizer with learning rate 2e-4, linear decaying, momentum 0.5, the size of minibatch was 4.

Below is depicted how errors were changing by iterations [Figure 8] [Figure 9].

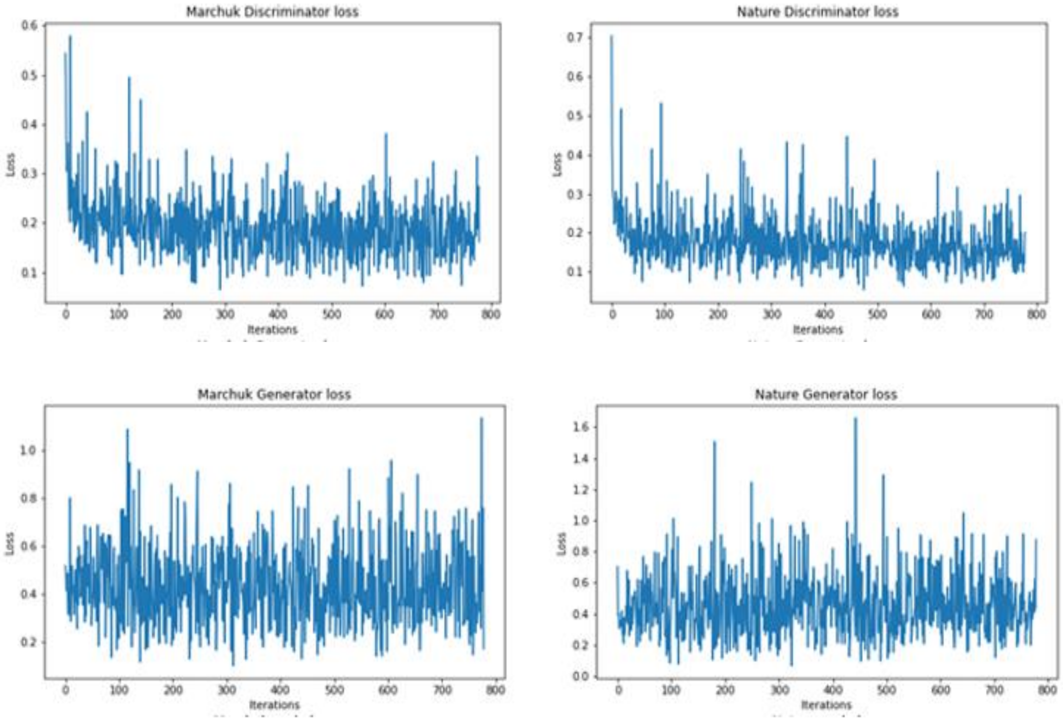

Figure 8. Loss changes by epoch

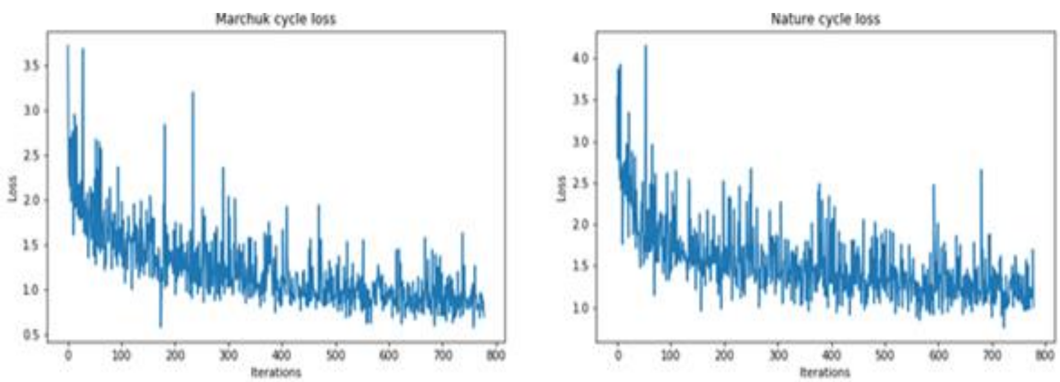

Figure 9. Loss changes by epoch

As can be seen on [Figure 8] and [Figure 9], discriminators quickly learned to recognize generated and original images restraining the generator in reducing the loss. It can be solved by taking a weaker model as a discriminator, for example, reducing the number of its parameters. But even this loss was enough to learn some most important patterns, as can be seen from the progress by the epochs [Figure 10] [18].

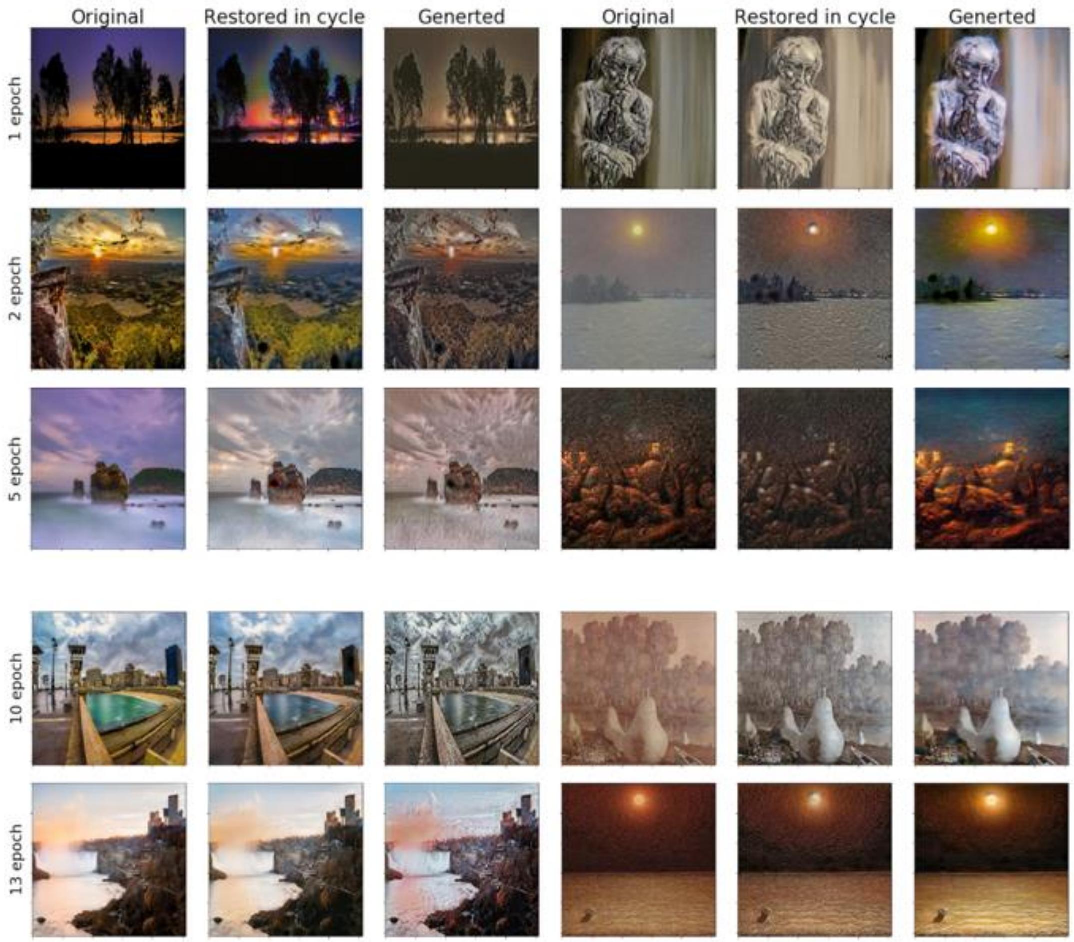

Figure 10. Progress by epochs

In [Figure 10], we can see that on the first epochs generators suffered from chessboard artifacts what is a result of deconvolution operation. Over time the problem ceased to be noticeable only in several separate photos. Instead, there appeared another problem, on the images with sea and sky, the generator merged them into one object, making the boundaries between them obscure, or added some artifacts, which can be seen on the example of 13 epoch. It may happen because there are too many abstract pictures in the training part. Such as the following [Figure 11]:

Figure 11. Example of abstract picture

And the discriminator has likely overfitted on such abstract paintings, causing the generator to add similar noise, paying less attention to the context of the [Figure 12].

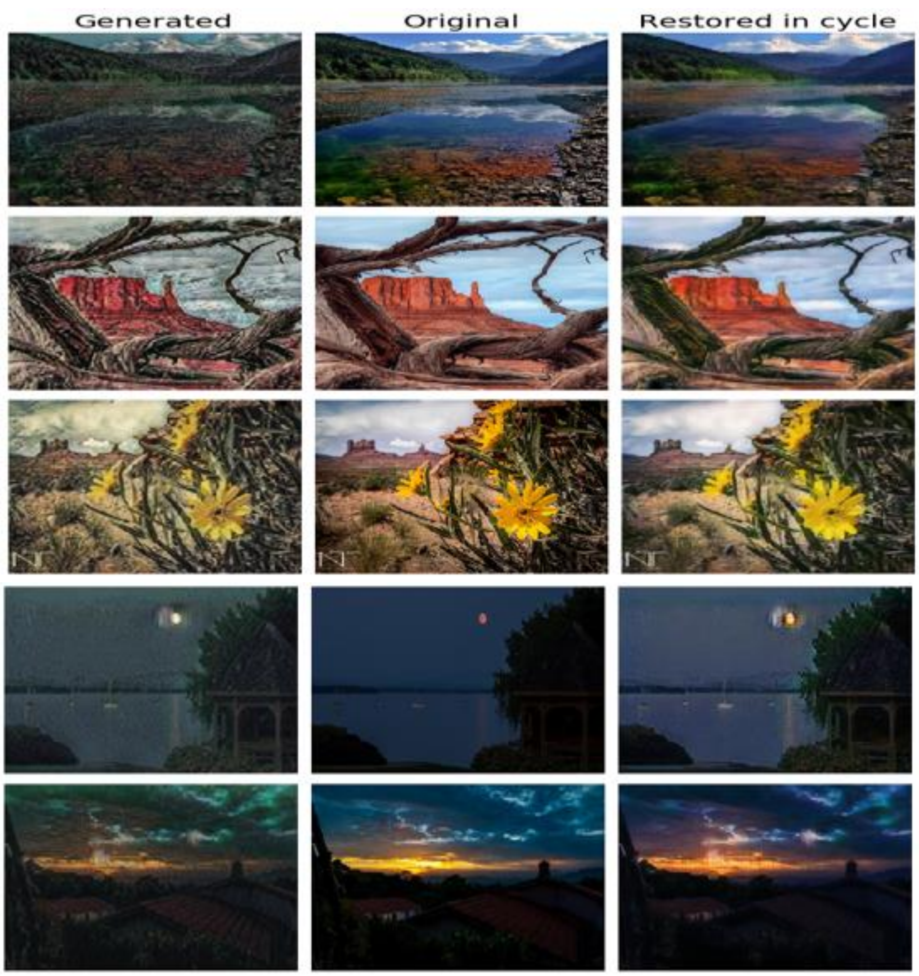

Figure 12. Results on 13 epoch

From these examples [Figure 13], we can see noticeable progress in the generated images. There are graininess and colors which are inherent to Marchuk's pictures. But still, some problem areas appear, like the moon in the fourth image.

Furthermore, the second generator, which is responsible for generating photos from pictures, was improving with each step, learning how to remove grain from the paintings and make more natural colors. Also, an interesting effect was observed, when the generator was processing some abstract paintings, it always painted it in blue and green colors.

Original Generated

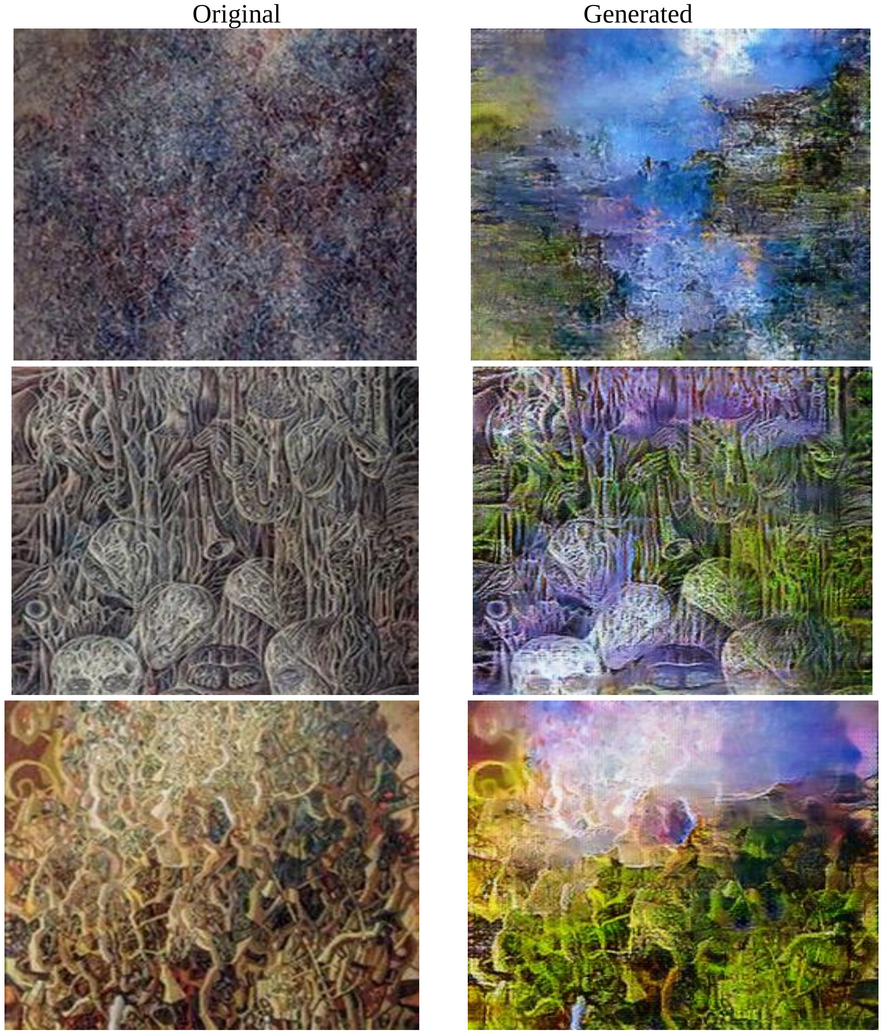

Figure 13. Example of transformations of abstract pictures

This effect can be explained by the fact that in the training part many photos were showing the sky and grass, and discriminator learned this, which forced generator to search zones where possibly could be sky and grass and paint them in the appropriate colors.

After getting and analyzing the results, it was decided to make a comparison with a different method: Style Transfer. For this purpose, four paintings by the artist were randomly selected as a style image and used VGG [19] as a pre-trained network. Below are shown results of the training:

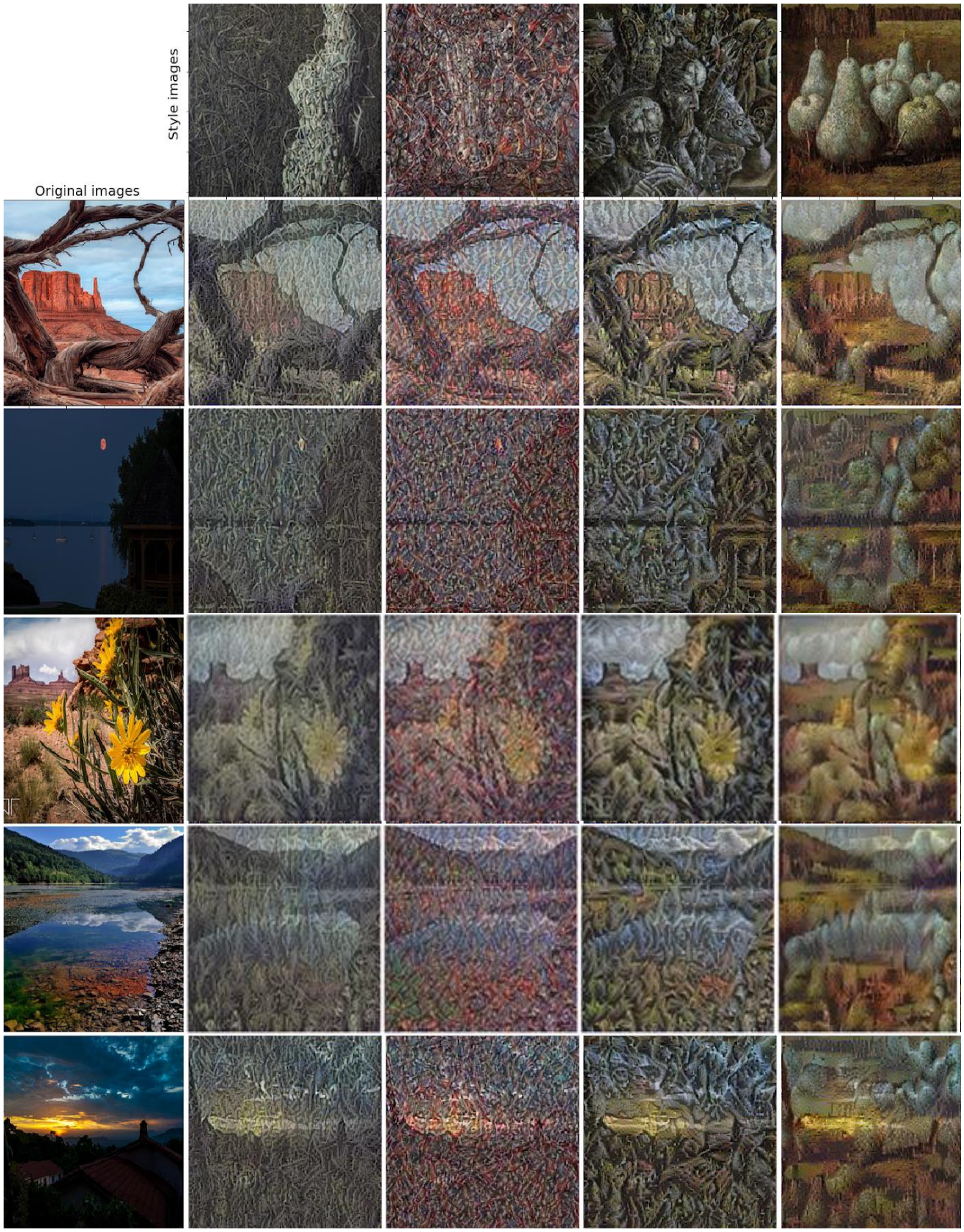

Figure 14. Results Style Transfer

In the [Figure 14], we can see confirmation of the previous words that in Style Transfer, choosing a style image has an enormous impact on the result. It has both pros and cons. Among the pros is the fast training and quality of the generated images, with a well-chosen style image. Among the disadvantages - the selection of the right style image can take a long time, but even with such an image, this method cannot provide such diversity as CycleGAN.

Summing up, although in some cases the network produced satisfactory results, it also did not go without a problem: generator performed poorly on images where both sea and sky (second image), and sometimes artifacts could appear on top of the image (first image).

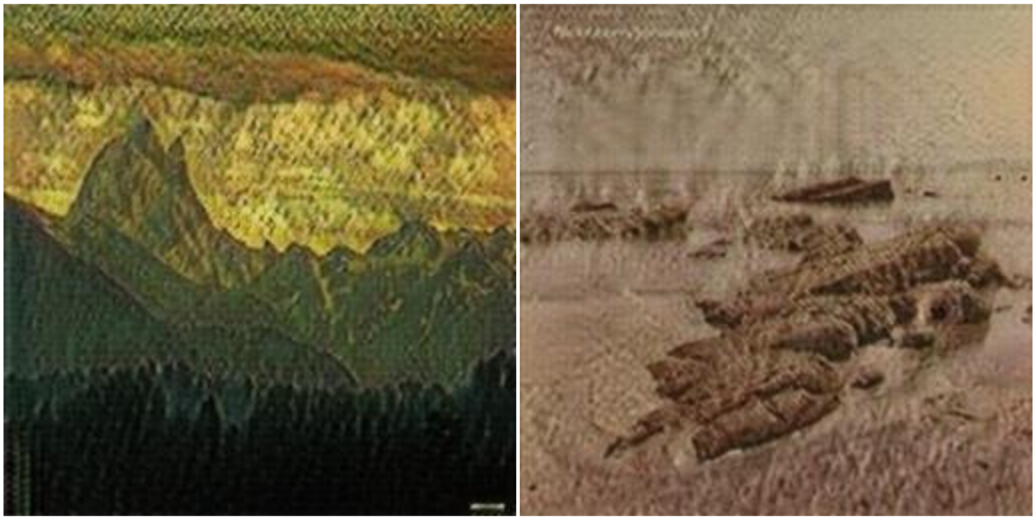

Figure 15. Couple bad examples

It may be the result of the overfitting of the neural network because in the first epochs following effects were not observed.

Figure 16. Results from the first epoch

The study analyzes the competing networks to use them to generate photos. During the work, a comparison of Neural Style Transfer methods and direct conversion of photos into photos (Pix2Pix) was performed. The use of these methods in practice is used to generate images with a certain "style". To achieve this goal in the study were implemented GAN and introduced their mathematical processing. Modifications to the GAN architecture are made to reproduce their images.

Comparing the Neural Style Transfer and Pix2Pix methods requires a large amount of data, which consists of paired photos, through the original photos in the desired style. Thanks to the use of GAN, you can compile data from images and real photos to which the selected style can be transferred.

The PatchGAN discriminator architecture evaluates each layer of the convolutional network independently and returns the pixel error that was involved in calculating significant batch to improve the evaluation of individual plot images. And in the architecture of the generator of the first part of the network there is an original image and when encoding reduces its dimension, and the other - gets the results of processing the first part and creates.

The paper performed actions to repeat the artist's style for this reduction of the image size to (256, 256, 3) and normalization within [-1, 1] for submission to the neural network input.

The training of the data transmission network with Ivan Marchuk's paintings and photos of nature was carried out with the help of Google Colab, Adam was used for the optimizer. On the example of GAN discriminators, it is faster to learn to develop the generated picture generators and not to allow to reduce an error.

Therefore, the selected problem can be solved with a weak model of the discriminator to reduce the number of its parameters. This approach allows the photo generator with each step to better remove the grain from the picture inherent in the author and give natural colors.

After obtaining and analyzing the results, the results were compared with the Style Transfer model, where VGG was taken as a trained neural network. As a result, the Transfer Transfer model is limited, desirable and fast to process.

Comparing with the method of transferring styles can give good results, but this requires the successful selection of a picture with style, and the fact that such an approach will not be able to provide such a variety of results.

As a result, the neural network has been trained, however, not without problems. One of the problems was the processing of images with the sea and the sky, often the generator made the boundaries between them too blurred.

As a result, the neural network has been trained, but the result cannot be called perfect. One of the problems was the treatment of the sky and water, the generator often made it too noisy, so the boundaries disappeared. One possible solution is to try other hyper-parameters, expand the data set by finding more pictures or by using the transfer of styles in random landscape photos. You can also use previously trained other data models. One solution is to balance the generator and the discriminator. This can be done by taking a weak model as a discriminator. In addition, the data set can be manually divided into train and test sets, taking all abstract images for testing.

9. Conclusion

The article reveals the importance of Generative Adversarial Networks, as they can be used in many areas. They are used to generate various media content, which allows you to generate a text description for the photo, process them, increase the size and convert images. In operation, the generator and discriminator are optimized by means of a stochastic gradient descent algorithm.

This study presents an overview of recent image-to-image transformation algorithms based on adversarial learning. The work presents algorithms for solving two tasks: style transfer and image restoration. The first task is image restoration, including ultra-high resolution, colorization, de-blurring, de-noising, and de-hazing. During the conducted experiments, the important role of competitiveness was proven. The second task is to choose a style for the selected set of images. It has been experimentally proven that the use of Style Transfer has a huge impact on the result and the choice of image style. Providing a variety of styles is only possible with CycleGAN.

The conducted research made it possible to evaluate the quality of the CycleGAN network, which eventually gave satisfactory results. However, image-to-image transformation to create images that maintain identity and have the same style can improve the effectiveness of visual recognition and its rapid learning. However, by adopting a GAN-based model, it is possible to observe the process of encoding the image by various parameters, to preserve the fine details and identity to identify the style of the image. For this purpose, the work of discriminators, who quickly learned to distinguish the pictures generated by the generators and not to allow stable reduction, is cited. This problem was also solved by taking a weaker model as the discriminator and changing the number of its parameters. In the process of experiments, the generator learned the main patterns and showed progress over epochs, which can be seen in the presented results. The presented model showed better results and allows improved known methods of image transformation to preserve identity and match style for small training samples.

10. Acknowledgements

The study was created within the framework of the project financed by the National Research Fund of Ukraine, registered No. 2021.01/0103, "Methods and means of researching markers of ageing and their influence on post-ageing effects for prolonging the working period", which is carried out at the Department of Artificial Intelligence Systems of the Institute of Computer Sciences and Information of technologies of the National University "Lviv Polytechnic".

References

[1] C. Li, M. Wand, “Precomputed real-time texture synthesis with markovian generative adversarial networks,” European conference on computer vision ECCV 2016, (2016) DOI: 10.1007/978-3-319-46487-9_43

[2] L. A. Gatys, A. S. Ecker, and M. Bethg, “A Neural Algorithm of Artistic Style”, Journal of Vision Sciences Society Annual Meeting Abstract , vol.16, p.326, (2016) DOI: 10.1167/16.12.326

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image to-image translation with conditional adversarial networks” In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017) DOI: 10.1109/CVPR.2017.632

[4] N. Boyko, A. Bronetskyi, N. Shakhovska, “Application of artificial intelligence algorithms for image processing”, Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”, MoMLeT&DS-2019, vol.2386, pp.194-211, **(2019)**

[5] N. Boyko and B. Mandych, “technologies of object recognition in space for visually impaired people”, The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19-21, CEUR, pp.338-347, **(2020)**

[6] N. Boyko and N. Tkachuk, “processing of medical different types of data using Hadoop and java MapReduce”, The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), pp.405-414, **(2020)**

[7] S. Ioffe and Ch. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift”, Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP, vol.37, pp.448-456, (2015) URL: arXiv: 1502.03167

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770-778, (2016) URL: arXiv: 1512.03385
[9] J. Johnson, A. Alahi, and F.F. Li, “Perceptual losses for real-time style transfer and super resolution”, European Conference on Computer Vision ECCV 2016, pp.694-711, (2016) DOI: 10.1007/978-3-319-46475-6_43
[10] D. P. Kingma and J. Ba, “Adam. A method for stochastic optimization”, 3rd International Conference for Learning Representations, San Diego, (2015) URL: https://arxiv.org/abs/1412.6980
[11] Open wikiart Vasil Kandinskiy URL: https://www.wikiart.org/uk/vasil-kandinskiy
[12] Open wikiart Anatoly Kkryvolap URL: https://anatolykryvolap.com
[13] Open wikiart Vasil Lopata URL: https://sverediuk.com.ua/vasil-lopata-hudozhnik-grafik/
[14] Open wikiart Kazimir Malevich URL: https://www.wikiart.org/uk/kazimir-malevich
[15] Open wikiart Mikola PimonenkoURL: https://www.wikiart.org/uk/mikola-pimonenko
[16] Open Landscape pictures dataset: URL:https://www.kaggle.com/arnaud58/landscape-pictures/data#
[17] Dataset Stock dynamics https://www.kaggle.com/econdata/stock-dynamics
[18] I. J.Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets”, NeurIPS Proceedings Advances in Neural Information Processing Systems 27, (2014) URL: arXiv: 1406.2661
[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image to-image translation with conditional adversarial networks” In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017) DOI: 10.1109/CVPR.2017.632

Author

Nataliya Boyko
Associated Professor at the Department of Artificial Intelligence, Lviv Polytechnic National University.
Scientific interests: Machine learning, data visualization, intellectual data analysis, system analysis

This page is empty by intention.