Scientific Research Publishing

Journal of Signal and Information Processing, 2018, 9, 217-228
http://www.scirp.org/journal/jsip
ISSN Online: 2159-4481
ISSN Print: 2159-4465

Realization of Virtual Human Face Based on Deep Convolutional Generative Adversarial Networks

Zijiang Zhu*, Xiaoguang Deng, Junshan Li, Eryou Wei

South China Business College, Guangdong University of Foreign Studies, Guangzhou, China
Email: *zzjdwh2002@163.com

How to cite this paper: Zhu, Z.J., Deng, X.G., Li, J.S. and Wei, E.Y. (2018) Realization of Virtual Human Face Based on Deep Convolutional Generative Adversarial Networks. *Journal of Signal and Information Processing*, **9**, 217-228.
https://doi.org/10.4236/jsip.2018.93013

Received: March 7, 2018
Accepted: August 19, 2018
Published: August 22, 2018

Abstract

At present, the generative adversarial networks research that generates a high confidence image for a large number of training samples has achieved some results, but the existing research only performs image generation for known training samples, but does not use the training parameters for image generation other than training samples. This paper uses the TensorFlow deep learning framework to build deep convolutional generative adversarial networks to complete the generation of virtual face images. From the experimental results, it can better generate virtual face images similar to real faces, which provides new ideas and methods for the research of generating virtual images.

Keywords

Deep Convolution, Generative Adversarial Networks, Deep Learning, Virtual Human Face

1. Introduction

In 2014, the scholar Ian Good fellow proposed the concept of generative adversarial networks [1]. Once proposed, generative adversarial networks quickly became a research hotspot in academia. The learning style of the collaborative confrontation network is unsupervised learning, but it avoids the difficulty of unsupervised learning very well, because each of the generative confrontation networks contains such a pair of models: the generation model and the discriminant model. Because of the discriminant model, the generated model can learn to approximate the real data without using the prior knowledge for complex modeling, and finally make the generated data reach a false degree.

Radford *et al.* published a paper in 2015 that presented deep convolutional

generative adversarial networks and used them for LSUN scene recognition challenges, Mnist handwritten numbers, and SVNH data sets [2]. In the LSUN dataset, the indoor scene image is successfully generated. On the Mnist and SVNH datasets, in order to verify the validity of the feature representation of the deep convolutional generative adversarial networks, the feature representation is input into the L2-SVM. Comparing the obtained classification results with other unsupervised algorithms, the results show that the highest accuracy of deep convolutional generative adversarial networks is 82.8%. At the same time, the paper points out that deep convolutional generative adversarial networks can be used to control the generation and disappearance of specific objects in a picture. The principle is that in the hidden space, if we can know which variables control an object, by blocking these variables, we can make a specific object in the picture disappear. Zhang *et al.* [3] used deep convolutional generative adversarial networks to create a text-to-image application in 2016. Generative adversarial networks generated related images based on specific sentences entered. The input of the generator is not only random noise, but also it includes specific statement information, and the discriminator identifies whether the sample is true or not and whether it matches the specific statement information. Isola *et al.* [4] applied generative adversarial networks to various existing deep neural network projects, such as segmenting images to restore original images, coloring black and white images, and coloring according to textures. Coutinho *et al.* [5] used generative adversarial networks for privacy protection, and Odena *et al.* [6] used deep convolutional generative adversarial networks for image synthesis. Gene Kogan used deep convolutional generative adversarial networks for the generation of Chinese calligraphy characters and achieved certain results. Some scholars have used deep convolutional generative adversarial networks for super-resolution image reconstruction. The basic idea is to input low-resolution images and then output high-resolution versions.

In 2015, Google Inc. opened up its internal deep learning framework TensorFlow [7], which has been applied to scenes such as image recognition, image segmentation, speech recognition, and natural language processing, and has achieved good industry results. The TensorBoard function in TensorFlow makes it easy to collect and visualize data such as loss values during training. This paper is based on the TensorFlow platform for the research and implementation of deep convolutional generative adversarial networks [2]. Adding a convolutional layer, a de-convolution layer, and using batch normalization based on the original generative adversarial networks makes the learning process faster and more stable.

2. Theoretical Basis for Deep Convolutional Generative Adversarial Networks (DCGAN)

2.1. Generative Adversarial Networks

The generative adversarial networks are composed of a generation model and a

discriminant model and are also known as generators and discriminators. The generator and discriminator compete with each other with the common goal of generating data points that are very similar to the data points in the training. The training process of generative adversarial networks is: the generator generates samples similar to real data from random noise or latent variables, and the discriminator judges the generated data and the real data, and feeds the result back to the generator. The generator and the discriminator are trained at the same time until a Nash equilibrium is reached, that is, the generated data generated by the generator is almost the same as the real sample, and the discriminator cannot accurately distinguish the generated data from the real data [8]. The structure of generative adversarial networks is shown in **Figure 1**.

In the process of training, the goal of the generator is to generate real data as much as possible to deceive the discriminator, and the target of the discriminator is to distinguish between the generated data and the real data. If it is true output 1, if it is false output 0, the two constitute a process of mutual game. The final result of the game is that the generator can generate false data, and the discriminator cannot distinguish between the real data and the generated data, that is, the discriminator output is 0.5, which is equivalent to guessing for distinguishing between true and false data.

The above is the core principle of generative adversarial networks, and the mathematical expression [8] is shown in Formula (1) below:

$$\min_G \max_D V(G,D) = \int_x P_{data}(x)\log\left(D(x)\right)\mathrm{d}x + \int_z P_z(z)\log\left(1-D\left(G(z)\right)\right)\mathrm{d}z \quad (1)$$

In Formula (1), x represents the actual data of the input, z represents the noise added to the generator, and $G(z)$ represents the data generated by the generator. $D(x)$ indicates the probability that the discriminator judges whether the real data is true, and the closer $D(x)$ is to 1, the better. $D(G(z))$ is the probability that the discriminator judges whether the data generated by the generator is true. Since the goal of the generator is to generate data that is comparable to the real data, the generator expects the value to be as large as possible, that is, expectation $V(G,D)$ becomes smaller, so there is $\min_G$ in front of the expression. For the discriminator, the stronger the discriminator's ability to judge, the larger $D(x)$, the smaller $D(G(z))$, at which time $V(D,G)$ becomes larger, so $\max_D$ is recorded. In the training process of generative adversarial

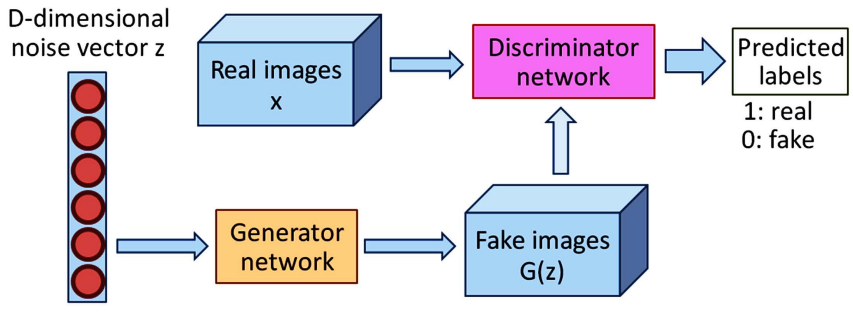

Figure 1. Basic structure of generative adversarial networks.

networks, using the random gradient descent method [9] training, the first training generator, using the gradient method, the larger the expected $V(G,D)$, the better, the second step is to train the discriminator, use subtracting the gradient method, it is expected that the smaller $V(G,D)$ is the better. When the most desirable result is reached, $D(G(z))=0.5$.

2.2. Deep Convolutional Generative Adversarial Networks

Although generative adversarial networks solve the problem of unsupervised learning that needs to consider prior knowledge, distribution, etc., there are some problems with generative adversarial networks. First of all, the generative adversarial networks have no convergence problem. When the generator and the discriminator use the neural network representation, when there is no equilibrium, the two are always in the process of adjusting the parameters. Secondly, generative adversarial networks are difficult to train. Because there is no loss function, it is difficult to judge whether progress has been made during training, and the training process may have a collapse problem, causing the generator to degenerate, continuously generating the same sample points and unable to continue learning, and then affect the training of the discriminator. Finally, because generative adversarial networks do not require prior knowledge to model, the process of training the model is too free and uncontrollable.

In order to solve the problem of unstable construction training against network training, Yang Yu *et al.* attempted to extend the generative adversarial networks using a supervised learning convolutional neural network architecture [2]. Convolutional neural networks have good performance in deep learning tasks with supervised learning, but are less used in unsupervised learning. After the use of generative adversarial networks to convolutional neural network architecture, the construction of generative adversarial networks can be achieved by using existing supervised learning tools to build deep convolutional generative adversarial networks, which greatly shortens the time of network construction of generative adversarial networks. And it improves the model construction and operational efficiency. Eventually they found an architecture that was able to train steadily on multiple datasets and generate higher resolution images, which were eventually named deep convolutional generative adversarial networks.

Deep convolutional generative adversarial networks have the following changes compared to generative adversarial networks: 1) It removes the fully connected layer, adopts a full convolutional network structure, replaces the space pooling with a step size convolution on the discriminator, allows the identification of network learning space down-sampling, and uses micro-step convolution on the generator, allowing learning to generate spatial up-sampling of the network; 2) It introduces the batch normalization, batch standardization in the generator and discriminator to make the data more centralized, do not worry about the data being too large or too small, improve the learning efficiency, solve the training problem of poor model initialization, and make the gradient spread deeper and prevent the builder crashes at a certain point during training; 3) In

terms of the activation function, the generator output layer is activated using the Tanh function, the other layers are activated using the ReLU function, and all layers of the discriminator are activated using the Leaky ReLU function.

Deep convolutional generative adversarial networks are initialized with random weights, so random input will produce a completely random image, but the network has a lot of parameters that can be adjusted. Therefore, our goal is to set the parameters so that the images generated according to the random input can be very similar to the real training data, that is, the purpose is to match the generated data with the distribution of real training data in the image space.

3. Realization of Virtual Human Face Image

3.1. Training Model

The TensorFlow deep learning framework provides a variety of APIs for convolution, de-convolution, and activation functions for neural network construction. Using the neural network library provided by TensorFlow, we can quickly and easily construct the neural network we need.

The virtual face generation application training process based on deep convolutional generative adversarial networks consists of two main parts. One is data preprocessing, which converts the CelebA face data set into 12 TF Records format files. The second is to define the relevant network layer, build the generator and discriminator, and write the training code to start the training of deep convolutional generative adversarial networks. Figure 2 is a flow chart of the experiment in this paper.

After getting our training model, you can use Tensor Board to view our model training loss value, neural network structure and other visual content, and you can also call the trained model to generate virtual face and view the model training effect.

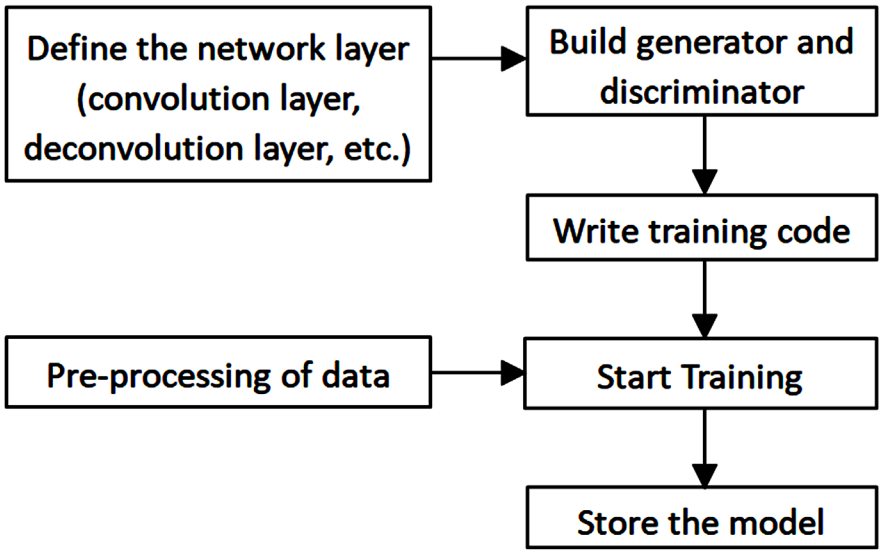

Figure 2. Training flow chart.

3.2. Preparation of TF Records Format of Data

Data preprocessing is the first step in this experiment. The link provided by the official website downloads the CelebA data set. The size of each face image is 178 × 218 × 3, and the input of the discriminator neural network is 64 × 64 × 3. Therefore, the face image needs to be cropped and then converted into a dataset in the TF Records format. In addition, since the number of face images in the CelebA dataset is as high as 200,000, the dataset needs to be divided into multiple TF Records format files. In this paper, it is divided into 12 TF Records files. The data preprocessing flowchart is shown in Figure 3.

3.3. Definition of Network Layer

In deep convolutional generative adversarial networks, the generator uses a deconvolutional neural network as the network layer, and the discriminator uses a convolutional neural network as the network layer.

The implementation of the deconvolution neural network in the generator is achieved by calling the conv2d_transpose method in the TensorFlow neural network library to achieve weight multiplication, and using bias_add to implement offset addition.

The implementation of the convolutional neural network in the discriminator is to achieve weight multiplication by calling the conv2d method in the TensorFlow neural network library, and bias addition is implemented using bias_add.

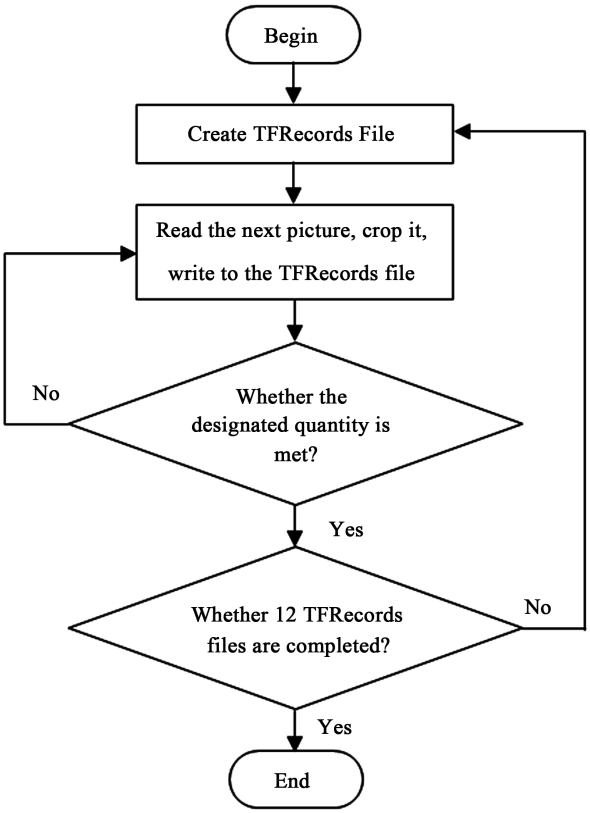

Figure 3. Data preprocessing flowchart.

3.4. Definition of Generator

Use the deconvolution layer defined above, you can build the generators needed for deep convolutional generative adversarial networks. The final output is 64 × 64, so set OUTPUT_SIZE to 64. The recursive moving step size 2, the output of each layer is expanded by four times than the input, so that the output size of each layer can be 32 × 32, 16 × 16.8 × 8, 4 × 4, respectively. The value of BATCH_SIZE is set to 64; the GF is set to 64, and the number of feature maps is 512, 256, 128, 64, respectively. Finally, the structure of the generator is shown in Figure 4.

3.5. Definition of Discriminator

Using the convolutional layer defined above, you can construct the discriminator required by deep convolutional generative adversarial networks, the picture input convolutional layer, the number of convolutional layer output feature maps, and the convolution kernel moving step size is 2. After the convolution process, the output is reduced to the original quarter, so the output sizes of the convolutional layers are 32 × 32, 16 × 16, 8 × 8, 4 × 4, respectively, and the number of feature maps is 64, 128, 256, and 512, respectively. The resulting discriminator structure is shown in Figure 5.

3.6. Training

The training process code is written primarily for the definition of the training process and the data acquisition process. The training process definition includes calling the network model to generate data, defining activation functions, and defining optimization algorithms. The data acquisition process definition includes data graph type selection, definition of acquisition time, and the like.

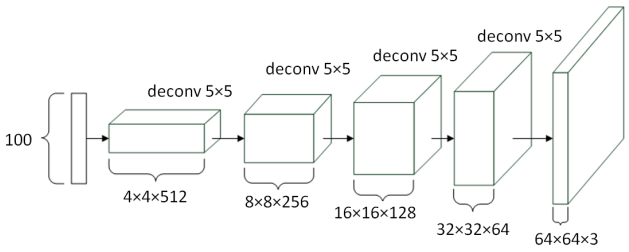

Figure 4. Generator.

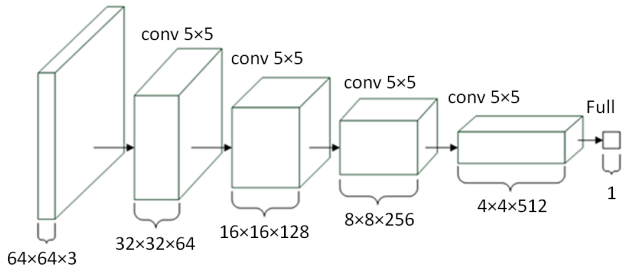

Figure 5. Structure of discriminator.

In this paper, the experiment activation function uses the sigmoid function to calculate the loss value generated during the training process by calling the sigmoid_cross_entropy_with_logits method in the Neural Network of TensorFlow. For the discriminator, the prediction from the real input should be 1 and the input generated from the generator is 0; for the generator, the discriminator should be predicted to generate data for it, and d_loss_real is the real data input to the discriminator. The cross entropy between the result and the expected result of 1; d_loss_fake is the cross entropy between the result of the generator generating data input to the discriminator and the result expected to be 0; d_loss is the sum of d_loss_real and d_loss_fake; g_loss is the result of the generator generating data input to the discriminator and expected to be 1. The result is the cross entropy between the two. The optimization algorithm selects AdamOptimizer [10], and its adaptive non-convex optimization feature is suitable for modern deep learning, without the need to manually adjust the learning parameters and other hyper-parameters.

Data collection includes Loss values, discriminator histograms, generator histograms, etc., mainly using TensorFlow's summary operations, including scalar_summary, histogram_summary, etc. Once you have completed these definitions, you can write data traversal code. Each epoch is sampled according to the specified batch, and the optimization algorithm is run to update the network until the end of the process. In this paper, set the epoch value to 5 and the batch value to 3072. The current Loss value is output every 20 steps. The current generator is tested every 100 steps to generate 64 virtual faces and save the generated virtual face image. The model is saved once every 500 steps using TensorFlow's Saver() method.

4. Experiment Results

4.1. Virtual Human Face Generation Effect

The data used in this experiment is the CelebA dataset. The CelebA dataset is a large face attribute dataset developed by the Chinese University of Hong Kong. It contains more than 200,000 celebrity face images, each with 40 attribute annotations. Training a total of 5 epoch, each epoch has 3072 training pictures, each 100-step sampling generator, and finally can get a total of 155 generator generated test charts, a partial screenshot of the test chart shown in **Figure 6**. Each test chart contains 64 virtual faces.

You can see that the generator test chart with epoch is 0 and the number of steps is also 0. As shown in **Figure 7**, you can see the generation of 64 100-dimensional random data after processing by the generator. It failed to generate a face image.

Looking at the test chart with epoch 0 and 3000 steps, as shown in **Figure 8**, it can be seen that after a round of training, the generator has been able to generate the basic face of humans. However, it is obvious that the generated face is relatively deformed, and the "eyes" have not yet been correctly generated. Most of the faces of the faces are replaced with black shadows.

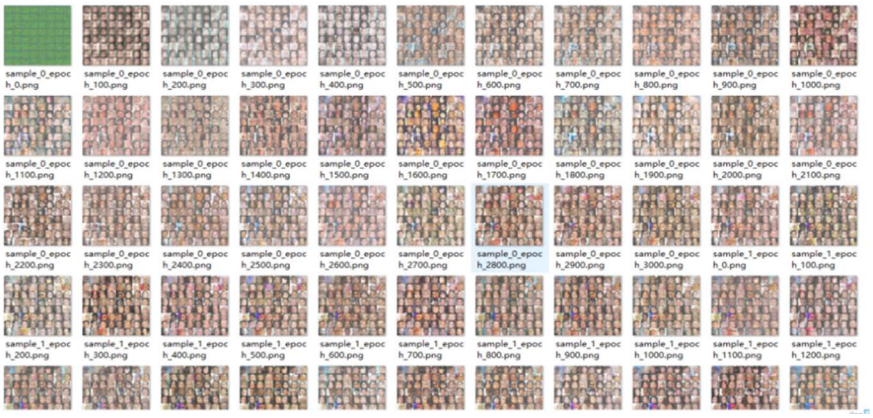

Figure 6. Screenshot of generator generation diagrams.

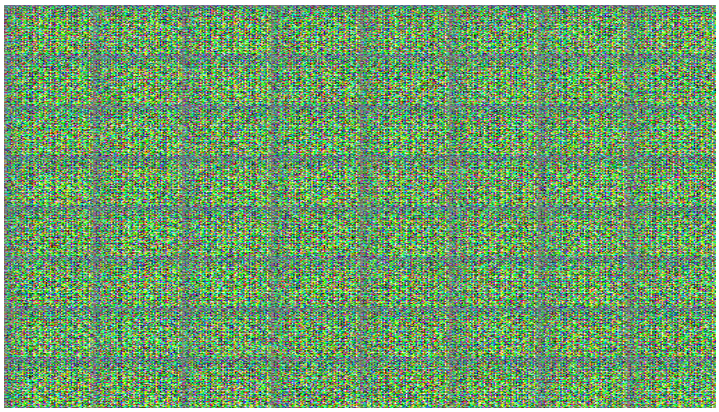

Figure 7. Test Chart with 0 epoch steps.

Figure 8. Test chart with 3000-0 epoch steps.

Finally, look at the last round of a test chart with epoch at 4, as shown in **Figure 9**. It can be seen that the generator has been able to correctly generate normal faces, and the eyes can be generated correctly, but there are some strange virtual human faces. For example, in the first column of the eighth line, it is not a complete face, only half a sheet. There is still room for improvement in the model.

Figure 9. Test chart with 400-4 epoch steps.

4.2. Analysis of the Application Prospect of Virtual Human Face

In this paper, the model obtained by deep convolutional generative adversarial networks training shows that the virtual face similar to the real face can be generated better, and the most direct application in the future is face modeling. The model data generated by the distribution of the real data of the face is generated by the model, and the face model database is constructed to solve the problem of insufficient annotation data. Secondly, thanks to the confrontation training mechanism of generative adversarial networks, virtual face images generated by deep convolutional generative adversarial networks can solve the problem of insufficient data sources faced by traditional machine learning. Therefore, virtual face images can be used in the field of semi-supervised learning in the future. For example, Chinese researcher Gou Chao proposed using simulated images and real images as sample data for neural network training, which is used to implement human eye detection [10].

5. Conclusion

Based on the TensorFlow deep learning framework, this paper studies the deep convolutional generative adversarial networks and realizes the generation of virtual face images to obtain better results. In addition to image generation, the application of deep convolutional generative adversarial networks in the field of image super-resolution can be considered in the future. The mechanism of image super-resolution is that high-resolution images with clear and rich detail are output by transforming low-resolution images. The problem with current image super-resolution is that high-frequency details are lost during the conversion process, which is contrary to its purpose. However, generative adversarial net-

works can solve this problem well, because the generative adversarial networks can learn the high-resolution distribution of the original image during the training process, and then output a good high-resolution version of the original image.

Acknowledgements

This work was supported in part by a grant from the characteristics innovation project of colleges and universities of Guangdong Province (Natural Science, No. 2016KTSCX182, 2016), a grant from the Youth Innovation Talent Project of colleges and universities of Guangdong Province (No. 2016KQNCX230, 2016).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde, D. and Ozair, S. (2014) Generative Adversarial Networks. *Advances in Neural Information Processing Systems*, **3**, 2672-2680. https://arxiv.org/abs/1406.2661

[2] Yu, Y., Gong, Z.Q., Zhong, P. and Shan, J.X. (2017) Unsupervised Representation Learning with Deep Convolutional Neural Network for Remote Sensing Images. 9*th International Conference on Image and Graphics*, *ICIG* 2017: *Image and Graphics*, Shanghai, 13-15 September 2017, 97-108. https://link.springer.com/chapter/10.1007%2F978-3-319-71589-6_9 https://doi.org/10.1007/978-3-319-71589-6_9

[3] Zhang, H., Xu, T. and Li, H. (2017) StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 5908-5916. https://doi.org/10.1109/ICCV.2017.629

[4] Isola, P., Zhu, J.Y., Zhou, T., Alexei A. and Efros, A.A. (2017) Image-to-Image Translation with Conditional Adversarial Networks. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 5967-5976. https://dx.doi.org/10.1109/CVPR.2017.632

[5] Coutinho, M., Albuquerque, R.O., Borges, F., Villalba, L.J.C. and Kim, T.H. (2018) Learning Perfectly Secure Cryptography to Protect Communications with Adversarial Neural Cryptography. *Sensors*, **5**, 1306-1321.

[6] Odena, A., Olah, C. and Shlens, J. (2016) Conditional Image Synthesis with Auxiliary Classifier GANs. arXiv:1610.09585. https://arxiv.org/pdf/1610.09585.pdf

[7] Abadi, M., Barham, P., Chen, J., Chen, Z.F., Davis, A., Dean, J., *et al.* (2016) TensorFlow: A System for Large-Scale Machine Learning. 12*th USENIX Conference on Operating Systems Design and Implementation*, Savannah, 2-4 November 2016, 265-283.

[8] Mao, X.D., Li, Q., Xie, H.R., Raymond, Y.K. and Wang, Z. (2017) Least Squares Generative Adversarial Networks. 2017 *IEEE International Conference on Computer Vision* (*ICCV*), Venice, 22-29 October 2017, 2813-2821. https://doi.org/10.1109/ICCV.2017.304

[9] Camps-Valls, G., Tuia, D., Bruzzone, L. and Benediktsson, J.A. (2014) Advances in Hyperspectral Image Classification: Earth Monitoring with Statistical Learning Methods. *IEEE Signal Processing Magazine*, **31**, 45-54. https://doi.org/10.1109/MSP.2013.2279179

[10] Gou, C., Wu, Y., Wang, K., Wang, K.F., Wang, F.Y. and Ji, Q. (2017) A Joint Cascaded Framework for Simultaneous Eye Detection and Eye State Estimation. *Pattern Recognition*, **67**, 23-31. https://doi.org/10.1016/j.patcog.2017.01.023