

CycleGAN and SRGAN to Enrich the Dataset

Budi Priswanto^{1)*}, Handri Santoso²⁾

¹⁾²⁾ Universitas Pradita

¹⁾budi.priswanto@student.pradita.ac.id, ²⁾handri.santoso@pradita.ac.id

Submitted : Apr 17, 2022 | **Accepted** : Apr 18, 2022 | **Published** : Apr 19, 2022

Abstract: When developments in the field of computer science are growing rapidly. For example, the development of image or video predictions for various fields has been widely applied to assist further processes. The field of computer vision has created many ideas about processing using deep learning algorithms. Sometimes the problem with using deep learning or machine learning is in the availability of the dataset or the unavailability of the dataset. Various methods are used to add to or enrich the dataset. One way is to add an image dataset by creating a synthetic image. One of the well-known algorithms is Generative Adversarial Networks as an algorithm for generating synthetic images. Currently, there are many variations of the GAN to around 500 variants. This research is to utilize the Cycle GAN architecture in order to enrich the dataset. By doing GAN as a synthetic image generator. This is very important in procuring image datasets, for training and testing models of Deep Learning algorithms such as Convolutional Neural Networks. In addition, the use of synthetic images produces a deep learning model to avoid overfitting. One of the causes of the overfitting problem is the lack of datasets. There are many ways to add image datasets, by cropping, continuously rotating 90 degrees, 180 degrees. The reason for using Cycle Generative Adversarial Networks is because this method is not as complicated as other GANs, but also not as simple. Cycle GAN synthetic images are processed with Super Resolution GAN, which aims to clarify image quality. So that it produces a different image and good image quality.

Keywords: Cycle GAN; Super Resolution GAN; Deep Learning; Data acquisition; Varian GAN; Deep Learning

INTRODUCTION

Synthetic images have many uses, which at first glance, synthetic images have a lot of negative meanings. One of the uses of synthetic images is to increase the number of image datasets to be more. Several methods are widely used in adding datasets is to scrape images on google. In addition, by doing Automatic image cropping and resizing for image datasets. There are still a lot of additions by changing the rotation of the image by 90 degrees to 180 degrees, adding noise to the image, and flipping the image so that it produces a different image with the same image. The process of adding an image dataset is the image augmented method.

Datasets are very important in the deep learning process, with the more datasets, the more models will produce with better accuracy. While in getting the dataset sometimes it becomes difficult to get. This is the problem with doing training or building a model from scratch. At the stage of collecting a large number of image datasets, it becomes a challenge or the most difficult part if you get an inappropriate dataset. Another way is to search for datasets through public image datasets, search datasets through kaggle and through google image.

The creator of the Generative Adversarial Network (GAN) is Ian Goodfellow, who accidentally generates an image and there are discriminators as a classification, namely the original image and the fake image. GAN research began in 2014, from year to year there have been many people conducting research on the topic of Generative Adversarial Network.

Generative Adversarial Networks (GAN) is an unsupervised learning algorithm that was developed in 2014 (Goodfellow et al., 2014). Two neural networks used by this deep learning algorithm consist of a generator and a discriminator. With a structure like this, it can produce a synthetic image that is similar to the original image. The use of this algorithm is widely used in the manufacture of images, videos, and sounds. Cycle Generative Adversarial Networks is a GAN algorithm and is capable of translating Images which is also known as Image Translation. Cycle architecture GANs owned data must have an image that is divided into two paired domains

*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

and generates a synthesized image based on the results of the paired domains. The application of Cycle GAN is used for season translation, object transfiguration, generating photos from paintings, and style transfers.

One of the studies introduces specific working principles and the history of GAN development and various applications of GANs in digital image processing. Cycle-GAN is utilized in the fields of medical imaging analysis and bioinformatics (Lan et al., 2020). From the application of GAN to medical images, the GAN-based model provides a better solution to provide dataset deficiencies in performing medical image analysis. It can be considered one of the important additions to the manual labeling of the radiologist. Models based on one GAN are mostly used as a data augmentation method to increase image variety and quantity.

Super Resolution Generative Adversarial Network (Maqsood et al., 2021), (Lee et al., 2018) is an algorithm that is able to repair images with better image resolution. The results of generating the GAN Cycle image are then repaired using the Super Resolution Generative Adversarial Network (Wenlong et al., 2021). The purpose of this research is to produce a new image with good image quality in terms of pixel resolution. The Cycle GAN and Super Resolution GAN processes are intended to transfer images and at the same time improve images, which will enrich the image dataset.

LITERATURE REVIEW

Many previous studies on Generative Adversarial Networks (GAN). These studies discuss the synthetic image dataset from generated GAN. The discoveries of the new Generative Adversarial Network methodologies vary widely. So there are many variants of the Generative Adversarial Network.

Table 1. Previous research on Cycle GAN

Author	Topic	Advantage	Disadvantage
(Asaf et al., 2021)	Improved CycleGAN with application to COVID-19 classification	The discussion about making synthetic images using the GAN cycle is followed by the Grad-CAM process, where the image is given a different color from the original image, thus changing the image that is very different from the original image.	The result of making a synthetic image using Cycle GAN and Grad Cam aims to change the image that is different from the original. But what if the image generator or synthetic image is not clear or the resulting resolution is low. The disadvantage is that it doesn't use Super Resolution, which is used to add image resolution, so the image is clearer.
(Suarez et al., 2019)	Image vegetation index through a cycle generative adversarial network	The discussion in research discusses CycleGAN to obtain synthetic NIR images, using some loss of functionality, residual network architecture (RESNET) is used to go deeper without decreasing accuracy and error rate.	The drawback of this research is that it does not discuss synthetic images that are blurry and unclear, low resolution. Because the image generated from Cycle GAN does not necessarily get a good image. What if the synthetic image generated from the GAN cycle is low resolution?
(Won, 2022)	An Experiment on Image Restoration Applying the Cycle Generative Adversarial Network to Partial Occlusion Kompsat-3A Image	Discussion about GAN by improving the image using the GAN cycle. The recovered image data is KOMPSAT3A satellite.	Weaknesses in this study, do not use methods such as the Super Resolution Generative Adversarial Network to overcome the problem of image blur.

*name of corresponding author



(Tang et al., 2019)	Unpaired Low-Dose CT Denoising Network Based on Cycle-Consistent Generative Adversarial Network with Prior Image Information.	The discussion about Cycle GAN is used to improve image data so that the image quality is better.	The disadvantage is that low resolution images require image processing such as the Super Resolution Generative Adversarial Network.
(Park et al., 2019)	Adaptive weighted multi-discriminator CycleGAN for underwater image enhancement.	Discussion Cycle GAN is used to improve the quality of the image. The process by adding adaptive weighting to limit the disadvantages of the two types of discriminators to balance their effects and stabilize the training procedure.	The disadvantage is that low resolution images require image processing such as the Super Resolution Generative Adversarial Network.

In the previous study, which has been discussed in table 1, it was stated that the study used GAN to produce new synthetic images. Several studies to produce synthetic images using Cycle Generative Adversarial Network. The question that arises in this research is, what about the synthetic image that is produced but the resolution is low? **The state of the art** in this research is to create a synthetic image using the Cycle Generative Adversarial Network by adding a synthetic image process from the GAN cycle to the Super-Resolution Generative Adversarial Network process to produce a synthetic image with good image resolution quality. **Contribution** of Cycle GAN and SR GAN in addition to the number of image datasets. The limitation of this research only discusses the Cycle GAN process and the Super-Resolution Generative Adversarial Network.

METHOD

This study proposes a method for the deep learning process to generate synthetic images that are used to enrich the dataset, so that there is no problem of dataset shortages for the training process and the testing process. In figure 1, it is explained how to start the modeling process.

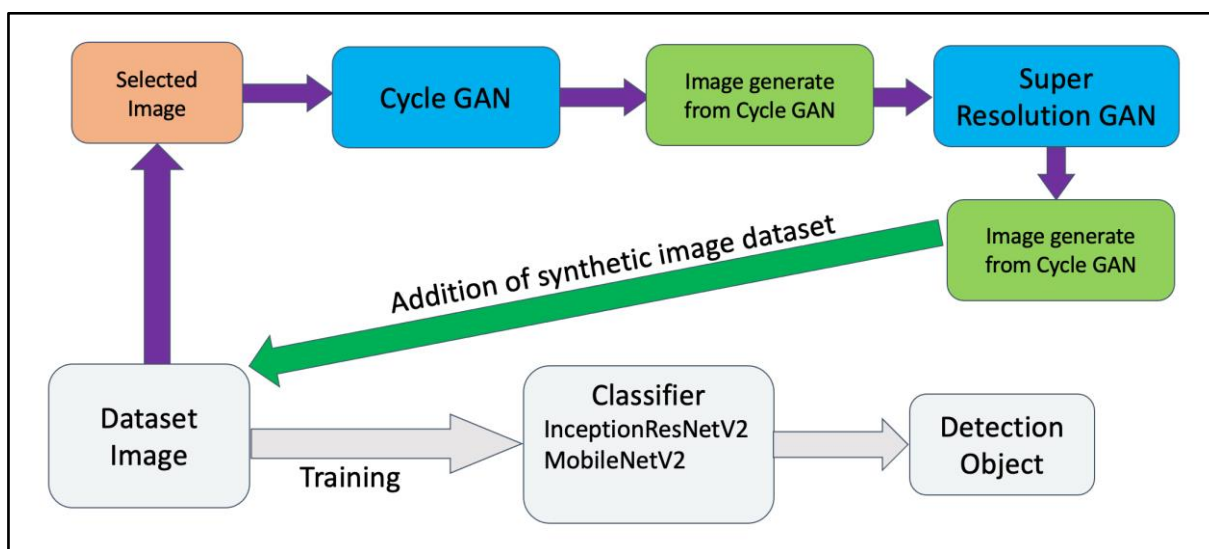


Fig. 1 Proposed method for research
Source : researcher property

Figure 1 shows the proposed research by processing with a combination of Cycle-GAN and Super-Resolution-GAN. The image dataset is selected for further Cycle-GAN process, from that process produces a synthetic image. Furthermore, the GAN Super Resolution process is carried out with the aim of increasing the image resolution so that the image resolution becomes higher. The resulting synthetic image is much better than the

*name of corresponding author



low-resolution image. Synthetic images resulting from the Super-Resolution Generative Adversarial Network process will be collected as an additional image dataset for further deep learning processes.

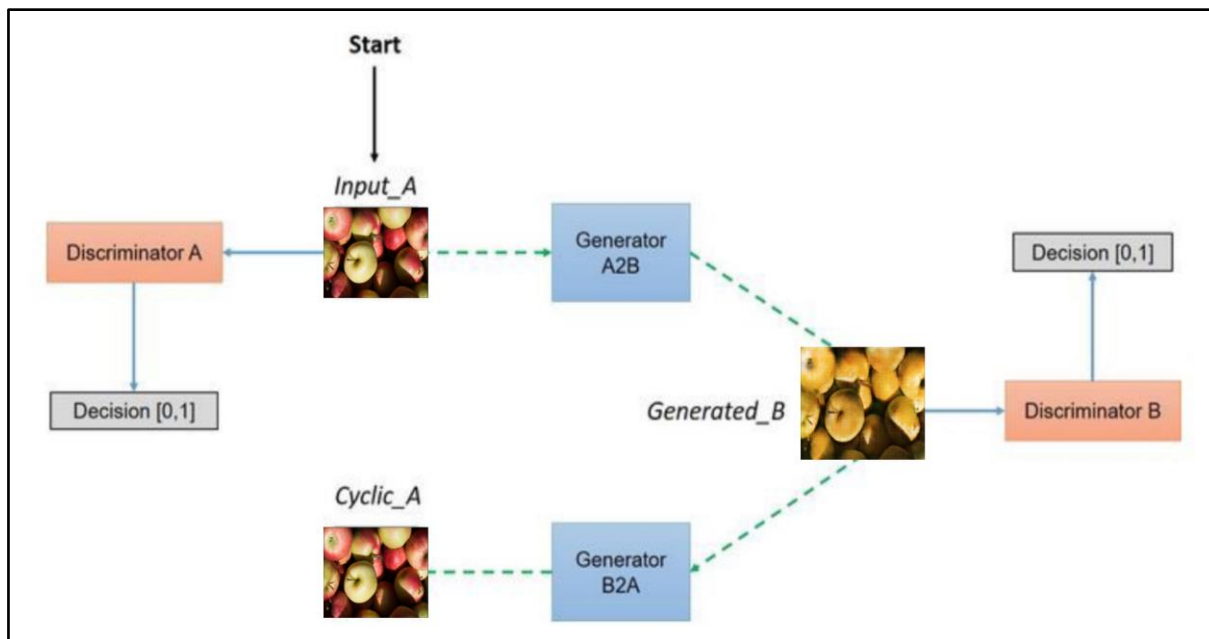


Fig. 2. Cycle Generative Adversarial Network

Source : <https://towardsdatascience.com/>

The Cycle-GAN method, used in this study, extends the GAN method to the domain of semantic segmentation, by performing unpaired image-to-image translation while reusing synthetic image annotations. However, this method is also limited in the transfer of semantic information and has limited domain shifting capabilities, when the classes between sets are different. Recent results in parallel with this study have demonstrated progress to enable unattended domain adaptation for pixel-level semantic segmentation and increase the realism of synthetic image datasets. Currently, a new method is proposed that works both at the feature level and at the pixel level (Barth et al., 2020).

Figure 2 is an image for a diagram of the entire GAN Cycle process. The Cycle GAN algorithm has two unpaired domains. Then refer to one of the goals. Cycle GAN can pair both images into one new image without having to have a reference first. Based on the figure, it consists of two discriminators, two image inputs, two generators, and one image output. The first discriminator serves to check the validation of the real image which is then compared with the second discriminator. In the second discriminator, validation checks are taken based on the images produced by the two generators. The first generator is for the first original image that is in the image data, while the second generator is for the second original image that is in the second image data. In the second generator the image is usually often replaced with a style image model, so it is called Style Transfer. Then the two generators produce one synthesis data in the form of an image that has the features of both (Akbar et al., 2021).

In figure 2, there are two different sets of images, apples, and oranges, one generator converts the image of apples into images of oranges, and the other converts oranges into apples. Like flipping the first image to the second image. During the training phase, the discriminator is here to check whether the image calculated by the generator looks real or fake. Through this process, the generator can get better feedback from each discriminator. CycleGAN method, the generator will get additional feedback from other generators. This feedback ensures that the image produced by the generator is cycle consistent, meaning that applying both generators successively to the image will produce a similar image (Efros et al., 2017).

This GAN cycle is widely used for various purposes, for example, to produce images by transferring the first image to the second image, actually, this process is similar to style transfer, where the style transfer process uses paired images, but this GAN Cycle does not have image pairs like style transfer. Cycle GAN uses only two image inputs and two discriminators.

*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

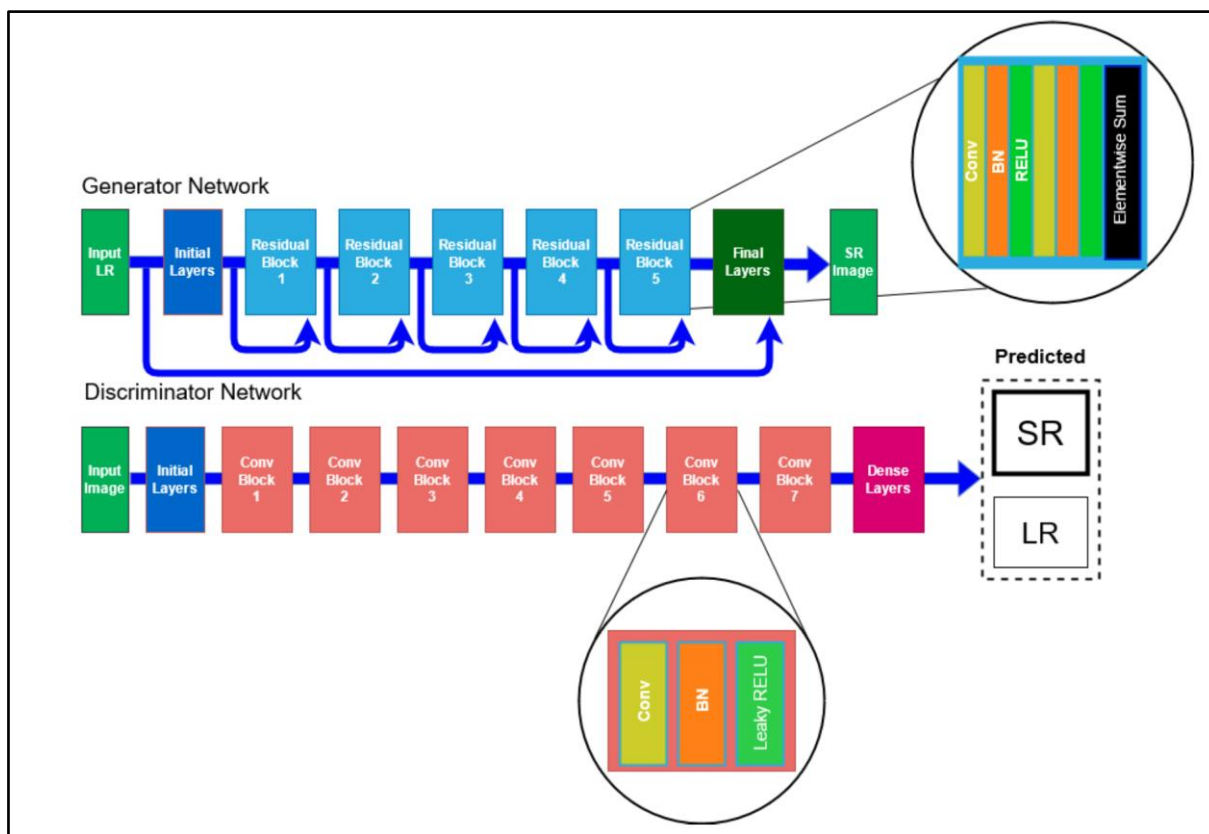


Fig. 3 Super-Resolution Generative Adversarial Network
Source : (Ledig et al., 2017)

Some studies that are very challenging to estimate high resolution (HR) images from low resolution (LR) images are called super-resolution (SR). Super Resolution has received great attention from participants in the computer vision research community and has developed various applications (Nasrollahi & Moeslund, 2014), (Yang et al., n.d.). In principle, for Super-Resolution Generative Adversarial Network, there are two parts, namely Generator Network and Discriminator Network. The generator performs the process, from Input Low Resolution (LR) then proceeds with Initial Layer to Residual Block 1, Residual Block 2, Residual Block 3, Residual Block 4, Residual Block 5. Then proceed with Final Layer to get a Super Resolution image. Each residual block has a Convolution process, Batch Normalization, ReLU, to Elementwise Sum.

For the Discriminator Network process, the Input Image process is continued to Initial Layers then Convolution Block 1, Convolution Block 2, Convolution Block 3, Convolution Block 4, Convolution Block 5, Convolution Block 7 up to Dense Layers process and become Super Resolution and Low-Resolution predictions. The Convolution Block process consists of Convolution, Batch Normalization, and Leaky RELU. This discriminator does not produce an image but a classifier that can determine Super Resolution or Low Resolution. This is what makes the Generator Adversarial Network method used as a method for changing images from low resolution to Super Resolution or High Resolution. Does not require additional devices to repair pixels in the image. The concept is only to do the Super-Resolution Generative Adversarial Network method.

To classify orange images into two classes, namely super-resolution and low resolution, a three-layer CNN model is used. The network consists of two convolution layers with 20x3x3 kernels each. The max-pooling layer is then used before the classification layer is used to identify the class each image belongs to. The same model architecture is used to train both versions of the data, namely low resolution, and high resolution. The model architecture is shown in Figure 4. Discussion Model training is carried out using training sets and validation. The validation set was created by setting aside 15% of the training data. The model was trained to study the effectiveness of the images sampled using SRGAN for classification. The optimization of the model uses Adam, and the learning rate = 0.001 to ensure there are no glitches (Maqsood et al., 2021).

The Super-Resolution Generative Adversarial Network is used for upsampling the image to increase its resolution which helps the convolutional neural network (CNN) learn high-quality features during training. This research empirically shows that the Super-Resolution Generative Adversarial Network can be used effectively to improve image quality and produce much better results when compared to models trained using low-resolution

*name of corresponding author



images (Maqsood et al., 2021). This is evident from the results obtained on the upsampled image, which is 83% of the overall test accuracy, which is substantially better than the overall test accuracy achieved for low-resolution images, which is 75%

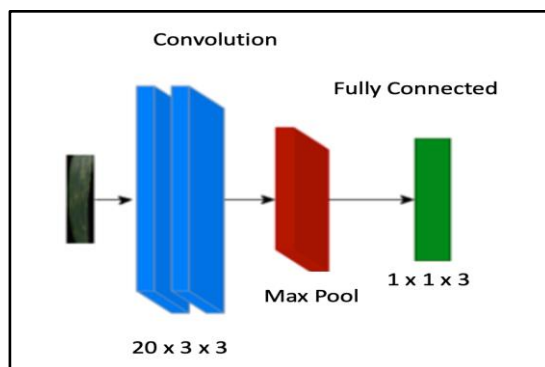


Fig. 4 CNN model architecture
Source : (Maqsood et al., 2021)

RESULT

From the experiment using the image dataset to convert the apple image into an orange image, the results obtained using Cycle GAN as shown in Figure 5.

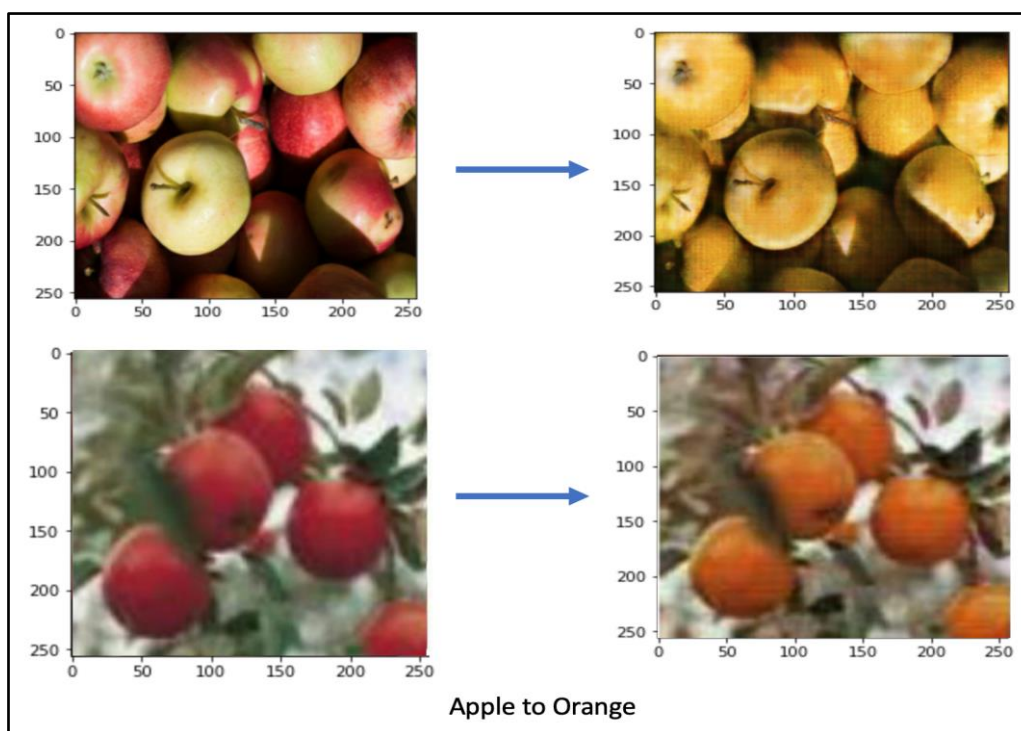


Fig. 5 Hasil dari Cycle GAN dan Super Resolution Generative Adversarial Network.

Source : researcher property

The loss of discriminator x is 0,5237 and the loss discriminator on y is 0,5880, the use of epoch 40, with a training time of 72 hours, still produces an image that is not bright. This process runs slowly because it uses the CPU device, not the GPU. This is especially effective with computers that use GPUs. The use of hardware that does not support causes the resulting model to not have high accuracy, so the model made is not optimal.

DISCUSSIONS

Here's a script that can be displayed in an experiment using Cycle GAN, where this script snippet intends to generate a synthetic image.

*name of corresponding author



Script Generator

```
def generate_images(model, test_input, epoch):  
    prediction = model(test_input)  
    display = [test_input[0], prediction[0]]  
    fig, axes = plt.subplots(1,2, figsize = (9,9))  
    title = ["Input image", "Generated image"]  
    for i in range(2):  
        axes[i].imshow(display[i] * 0.5 + 0.5)  
        axes[i].set_title(title[i])  
    plt.axis("off")  
    plt.savefig("output/output_{ }.jpg".format(epoch))  
    plt.close()
```

Script Training model

```
def train_step(real_x, real_y):  
    with tf.GradientTape(persistent=True) as tape:  
        fake_y = generator_g(real_x, training = True)  
        cycle_x = generator_f(fake_y, training = True)  
        fake_x = generator_f(real_y, training = True)  
        cycle_y = generator_g(fake_x, training = True)  
        same_x = generator_f(real_x, training = True)  
        same_y = generator_g(real_y, training = True)  
  
        disc_real_x = discriminator_x(real_x, training = True)  
        disc_real_y = discriminator_y(real_y, training = True)  
        disc_fake_x = discriminator_x(fake_x, training = True)  
        disc_fake_y = discriminator_y(fake_y, training = True)  
  
        #calculate loss  
        gen_g_loss = generator_loss(disc_fake_y)  
        gen_f_loss = generator_loss(disc_fake_x)  
  
        total_cycle_loss = calc_cycle_loss(real_x, cycle_y) + calc_cycle_loss(real_y, cycle_y)  
        total_gen_g_loss = gen_g_loss + total_cycle_loss + identify_loss(real_y, same_y)  
        total_gen_f_loss = gen_f_loss + total_cycle_loss + identify_loss(real_x, same_x)  
        disc_x_loss = discriminator_loss(disc_real_x, disc_fake_x)  
        disc_y_loss = discriminator_loss(disc_real_y, disc_fake_y)  
  
        generator_g_grad = tape.gradient(total_gen_g_loss, generator_g.trainable_variables)  
        generator_f_grad = tape.gradient(total_gen_f_loss, generator_f.trainable_variables)  
  
        discriminator_x_grad = tape.gradient(disc_x_loss, discriminator_x.trainable_variables)  
        discriminator_y_grad = tape.gradient(disc_y_loss, discriminator_y.trainable_variables)  
  
        # optimizer  
        generator_g_opti.apply_gradients(zip(generator_g_grad, generator_g.trainable_variables))  
        generator_f_opti.apply_gradients(zip(generator_f_grad, generator_f.trainable_variables))  
        discriminator_x_opti.apply_gradients(zip(discriminator_x_grad, discriminator_x.trainable_variables))  
        discriminator_y_opti.apply_gradients(zip(discriminator_y_grad, discriminator_y.trainable_variables))  
    return total_gen_g_loss, total_gen_f_loss, disc_x_loss, disc_y_loss
```

In training the model using the CPU instead of the GPU, there are differences in results, because in training the model it is necessary to limit the time. The time required has reached 72 hours or 3 days, so the computer becomes hot and very slow, so the training process is limited to 40 epochs. To get maximum results, it is necessary to do training for a long time or epochs of up to 200 epochs, so that the resulting synthetic image will be maximal and good in terms of image. From low resolution to super-resolution, the image looks clear and nice.

*name of corresponding author



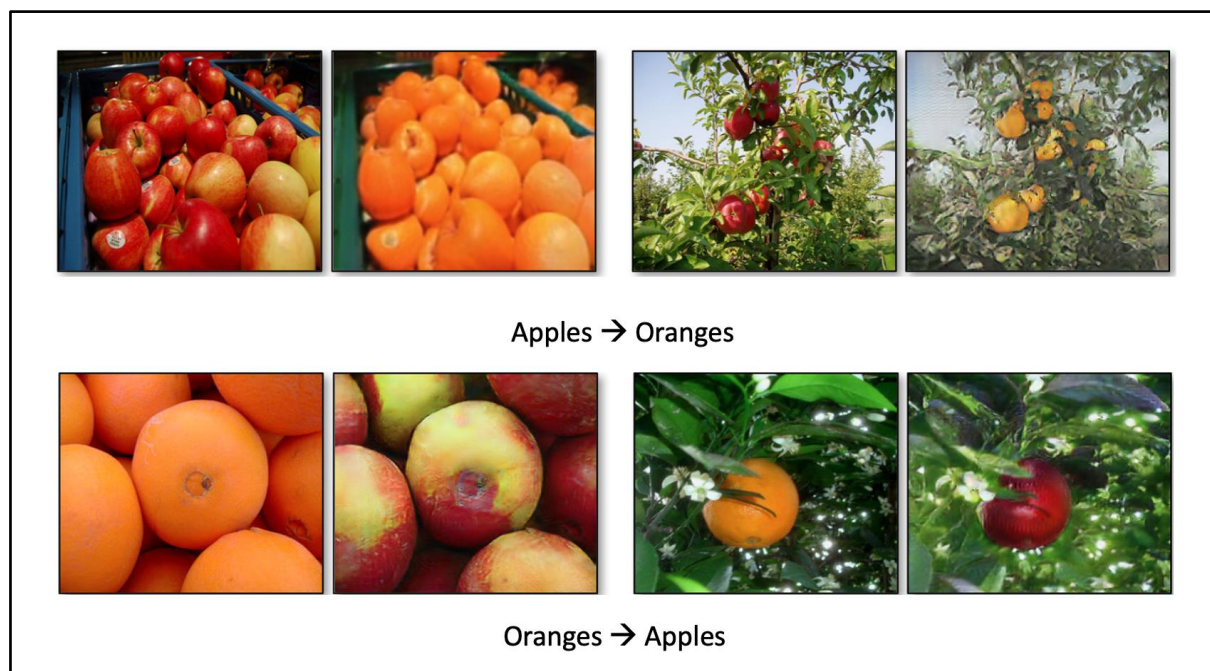


Fig. 6 Cycle GAN and Super Resolution GAN

Source : (Efros et al., 2017)

CONCLUSION

The conclusion of this study combines two Generative Adversarial Networks (GAN), namely Cycle GAN with Super Resolution GAN. This method gets the maximum, due to computer limitations with hardware specs Core I 5, DDR3 12GB, SSD 250 GB, without GPU. Training takes a very long time, which is 72 hours, with 40 epoch conditions. The results of the training as shown in Figure 5.

SUGGESTION

This research needs to be done by adding weight factors that aim to improve the image and add training time and 200 epochs. The goal is to get a better image enhancement and high resolution.

REFERENCES

- Akbar, A., Pramadhan, G., & Eka, S. (2021). Cycle Generative Adversarial Networks Algorithm With Style Transfer For Image Generation. *Computer Vision and Pattern Recognition*. <https://doi.org/https://doi.org/10.48550/arXiv.2101.03921>
- Asaf, B.-E., Dana, C., Noa, C., & Hayit, G. (2021). Improved CycleGAN with application to COVID-19 classification. *Medical Imaging*. <https://doi.org/https://doi.org/10.1117/12.2582162>
- Barth, R., Hemming, J., & Van Henten, E. J. (2020). Optimising realism of synthetic images using cycle generative adversarial networks for improved part segmentation. *Computers and Electronics in Agriculture*, 173(March), 105378. <https://doi.org/10.1016/j.compag.2020.105378>
- Efros, J.-Y., Zhu, Taesung, P., Phillip, I., & A, A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *Computer Vision and Pattern Recognition*. <https://doi.org/https://doi.org/10.48550/arXiv.1703.10593>
- Goodfellow, I., Mehdi, M., Bing, X., & David, W. (2014). *Generative Adversarial Nets*.
- Lan, L., You, L., Zhang, Z., Fan, Z., Zhao, W., Zeng, N., Chen, Y., & Zhou, X. (2020). Generative Adversarial Networks and Its Applications in Biomedical Informatics. *Frontiers in Public Health*, 8(May), 1–14. <https://doi.org/10.3389/fpubh.2020.00164>
- Ledig, C., Lucas, T., Ferenc, H. ar., Jose, C., Andrew, C., Alejandro, A., Andrew, A., Alykhan, T., Johannes, T., Zehan, W., & Wenzhe, S. (2017). *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*.
- Lee, Y., Yun, J., Hong, Y., Lee, J., & Jeon, M. (2018). Accurate License Plate Recognition and Super-Resolution Using a Generative Adversarial Networks on Traffic Surveillance Video. *2018 IEEE International Conference on Consumer Electronics - Asia, ICCE-Asia 2018, June 2018*, 1–4. <https://doi.org/10.1109/ICCE-ASIA.2018.8552121>

*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

- Maqsood, M. H., Mumtaz, R., Haq, I. U., Shafi, U., Zaidi, S. M. H., & Hafeez, M. (2021). Super resolution generative adversarial network (Srgans) for wheat stripe rust classification. *Sensors*, 21(23), 1–12. <https://doi.org/10.3390/s21237903>
- Nasrollahi, K., & Moeslund, T. B. (2014). Super-resolution: A comprehensive survey. In *Machine Vision and Applications* (Vol. 25, Issue 6). <https://doi.org/10.1007/s00138-014-0623-4>
- Park, J., Han, D. K., & Ko, H. (2019). Adaptive weighted multi-discriminator CycleGAN for underwater image enhancement. In *Journal of Marine Science and Engineering* (Vol. 7, Issue 7). <https://doi.org/10.3390/jmse7070200>
- Suarez, P. L., Sappa, A. D., Vintimilla, B. X., & Hammoud, R. I. (2019). Image vegetation index through a cycle generative adversarial network. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2019-June*(1), 1014–1021. <https://doi.org/10.1109/CVPRW.2019.00133>
- Tang, C., Li, J., Wang, L., Li, Z., Jiang, L., Cai, A., Zhang, W., Liang, N., Li, L., & Yan, B. (2019). Unpaired Low-Dose CT Denoising Network Based on Cycle-Consistent Generative Adversarial Network with Prior Image Information. *Computational and Mathematical Methods in Medicine, 2019*. <https://doi.org/10.1155/2019/8639825>
- Wenlong, Z., Yihao, L., Chao, D., & Yu, Q. (2021). RankSRGAN- Super Resolution GenerativeAdversarial Networks with Learning to Rank. *JOURNAL OF LATEX CLASS FILES*.
- Won, T. (2022). *An Experiment on Image Restoration Applying the Cycle Generative Adversarial Network to Partial Occlusion Kompsat-3A Image*. 38(1), 33–43.
- Yang, Q., Yang, R., Davis, J., & Nist, D. (n.d.). *Spatial-Depth Super Resolution for Range Images*.

*name of corresponding author



This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.