Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 2, April 2022
DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Style Transfer Generator for Dataset Testing Classification

Bayu Yasa Wedha[1)*], Daniel Avian Karjadi[2)], Alessandro Enriqco Putra Bayu Wedha[3)], Handri Santoso[4)]
[1)2)3)4)] Universitas Pradita
[1)]bayu.wedha@student.pradita.ac.id, [2)]daniel.avian@student.pradita.ac.id, [3)] alessandro@ichthus.xyz,
[4)]handri.santoso@pradita.ac.id

Submitted : April 10, 2022 | **Accepted** : Apr 10, 2022 | **Published** : Apr 11, 2022

Abstract: The development of the Generative Adversarial Network is currently very fast. First introduced by Ian Goodfellow in 2014, its development has accelerated since 2018. Currently, the need for datasets is sometimes still lacking, while public datasets are sometimes still lacking in number. This study tries to add an image dataset for supervised learning purposes. However, the dataset that will be studied is a unique dataset, not a dataset from the camera. But the image dataset by doing the augmented process by generating from the existing image. By adding a few changes to the augmentation process. So that the image datasets become diverse, not only datasets from camera photos but datasets that are carried out with an augmented process. Camera photos added with painting images will become still images with a newer style. There are many studies on Style transfer to produce images in drawing art, but it is possible to generate images for the needs of image datasets. The resulting force transfer image data set was used as the test data set for the Convolutional Neural Network classification. Classification can also be used to detect specific objects or images. The image dataset resulting from the style transfer is used for the classification of goods transporting vehicles or trucks. Detection trucks are very useful in the transportation system, where currently many trucks are modified to avoid road fees.

Keywords: Generative Adversarial Networks, Convolutional Neural Network, Style Transfer, Image Dataset, Art Image

INTRODUCTION

The discovery of deep learning algorithms led to many scientific developments. In the field of computer vision, with the presence of deep learning, there are very varied discoveries in computer vision science. Still in research in the field of deep learning, until 2014 the idea of a Generative Adversarial Network emerged which was researched by Ian Goofellow. He accidentally generated a real dataset and tried to regenerate it until it resembled the original image. Of course with the help of the Discriminator in determining fake and real an image. Since then, research on the Generative Adversarial Network has continued to this day. There has been massive research since 2018, where this Generative Adversarial Network (GAN) has produced various variants. There are more than 500 variants of GAN that already exist in studies. For example, the Super Resolution Generative Adversarial Networks (SRGAN) researched that images with small pixels and blurring were converted into images with nice and clear pixels with larger image sizes. From Low Resolution to High Resolution (Guo et al., 2021), (Jiang et al., 2021), (Shim et al., 2022). CycleGAN, image translation technique using deep learning (Teramoto et al., 2021), (Modanwal et al., 2021). Make changes from Original object image to Synthetic object image (Barth et al., 2020). This image exchange is like changing an object and placing it in a certain place, the meaning of the place is the same but the object or image is moved.

Transfer Style is the process of rendering one image with several other image styles. Currently, there has been a significant improvement in image rendering quality. Transfer styles match traditional methods with regularizers, forcing the depth of content on the image. However, this traditional method is computationally inefficient and requires separate Neural Network training to derive new drawing styles.

Neural Transfer Style has different from GAN (Generative Adversarial Networks). Where the Generative Adversarial Network generates images from training data content using the random method. However, the Neural Transfer Style transfers the image style to the content image. The results generated from the Neural

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 2, April 2022
DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Transfer Style can set the desired output. However, the Generative Adversarial Network method cannot have the desired content image, because the GAN output is random.

Neural Transfer Style transfers the style on the style image to the content image. The Neural Transfer Style will transfer the color and pattern of the styled image to the content image. The Neural Transfer Style moves the detailed texture in the style image but retains the content of the content image. It's just that the resulting color is not solid (no gradation) like the image style. The result of pattern generation also produces a gradation pattern.

This algorithm in this study uses the VGG algorithm for feature extraction and training has been carried out for object recognition and localization. The CNN algorithm has the following three components: content, style, and style transfer. The content representation section is calculated using the mean and variance of content and style. CNN can transfer styles from one image to another image content. This algorithm has a cost for style and a fee for separate content. After that, it can change the composition of the weights between the cost of style and the cost of content, to get a new perceptual image. The stronger the force transferred to the image, the content in the image will be lost or vice versa.

The motivation of this research is to discuss Instance Normalization with the meaning of adaptive instance normalization (AdaIN). Content and style as input to be processed, AdaIN processes from the content average and content variance as input to match the style input (Huang, 2017). Through experimentation, AdaIN implementations can combine content and style and perform feature transfers statistically. Network decoder to generate image content with image styles. After that do the reversal of the output from AdaIN to the image space. The advantage of using the AdaIN Method is that the process is almost twice as fast. Furthermore, it provides abundant user control at runtime, without any modification to the training process. The purpose of this study focuses on style transfer. which can later be used as a testing dataset for the classification of Convolutional Neural Networks. This study does not discuss the classification problem but focuses on producing an image as a testing dataset.

1. LITERATURE REVIEW

This section attempts to review several studies related to the Style Transfer Generator for the Convolutional Neural Networks Classification testing dataset.

Table. 1 Previous research

Author	Topic Research	Advantages	Disadvantages
(Huang, 2017)	Generation of Batik Patterns by Using Neural Transfer Styles by Using Color Costs	The discussion on Style Transfer is good and clear enough.	The style transfer method does not explain the use or benefits of style transfer.
(Phon-Amnuaisuk, 2019)	Image Synthesis and Style Transfer	Discussion about generating synthetic images using GAN and using Style Transfer images.	The style transfer method does not explain the use or benefits of style transfer.
(Zheng & Liu, 2020)	P2-GAN: Efficient Style Transfer using Single Style Image.	Style transfer aims to redraw the content image into an artistic style image while preserving its semantic content. It has a rich history when it comes to texture transfer. The shift in engineering style allows the amateur to produce fantastic and versatile.	Style transfer is used for creating cartoon images or videos.

From the results of previous studies where there are shortcomings from previous studies. This research is to provide additional or complementary to previous research. The results of the style transfer are used as a testing dataset for classification using the Convolutional Neural Network. This research provides different views and insights, where stages like this can be carried out in this research.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 2, April 2022
DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

2. METHOD

The proposed research method is to detect the image of a transport vehicle. Dataset as input in the Convolution Neural Network classification. Certain datasets are selected for augmented image as dataset testing. Augmented image uses a transfer style that changes the image dataset into an art image dataset. The research method proposal is as shown in Fig. 1.

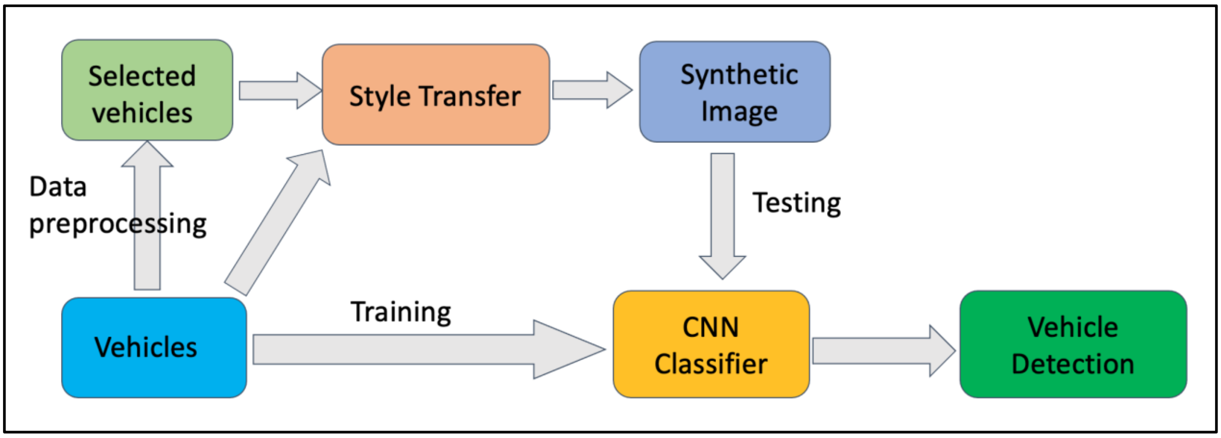

Fig. 1 Propose method in Style Transfer Generator research
Source : researcher property

The network transfer style uses the Encoder, AdaIN and Decoder methods to generate a new image. The encoder method uses the VGG16 algorithm, where the process uses 16 layers during the extraction process. So from the feature extraction process it becomes a feature that is passed into AdaIN. Both images were processed using VGG16. Photo images as image content and painting images as image style. AdaIN performs the process of taking the average value of the image content feature and the image style feature. In addition to the average feature value, it takes the variance value of the image content and image style features.

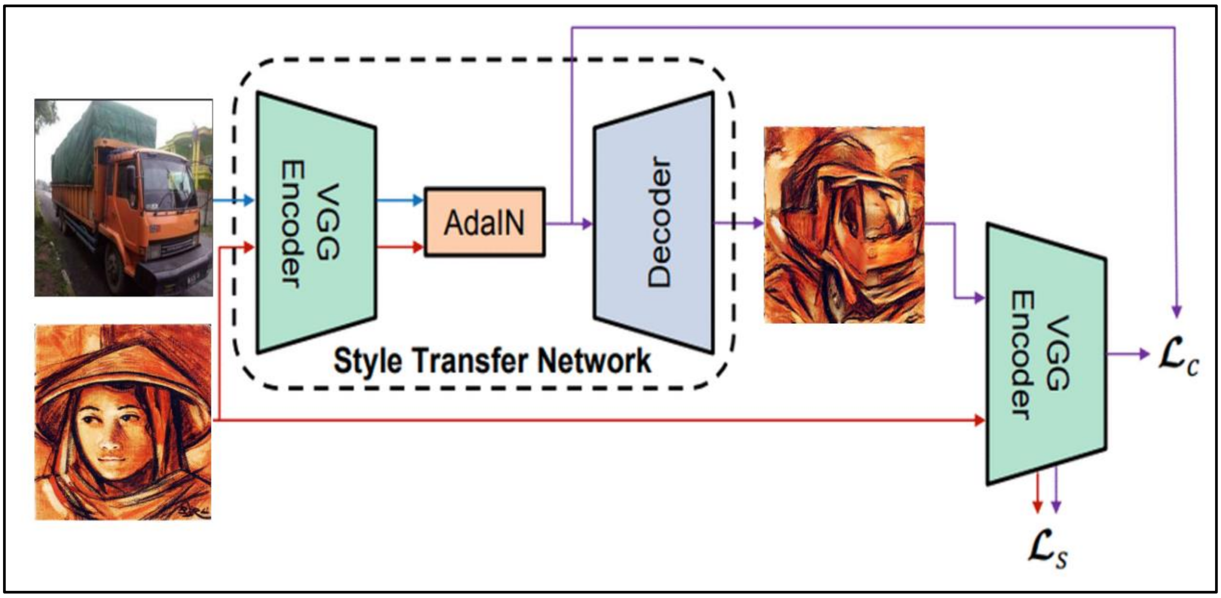

Fig 2. Style Transfer VGG and AdaIN
Source : (Huang, 2017)

After that it will generate Loss value of image content, loss of image style and total loss of image. Where the total loss of the image is the difference between the loss of image content and the loss of image style. The process of transferring the image style is as shown in Figure 2.

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 2, April 2022
DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Batch Normalization
A common problem in data science is the problem of overfitting. For example, the dataset training process produced a very good model but failed in the testing process. The training process to become a classification model accuracy reaches 95%, meaning that the model already has high accuracy. However, in the classification testing process there was an error in the process. This is one result of overfitting. The solution is regulation. The regulation technique is a technique that really helps improve the model and makes the training process faster. This technique also avoids a model that fits too well. Before discussing about Batch normalization, it is necessary to discuss Normalization. Normalization is data processing by pre-processing data using numeric data with a scale and usually marked without changing its original form.

Batch Norm is a layer of the Neural Network (in this case not the raw dataset) that normalizes between layers of the neural network. The process is carried out on mini-batches that are not included in the full dataset. The normalization process is very useful in speeding up training in larger datasets because the training process is simpler. The reason for the normalization process is to ensure that the model process can generalize correctly and precisely. The process for batch normalization is not done with one input, but can be done in stages using batches, so that the normalization process uses in stages or in batches. The following is the Batch Normalization equation in (1):

$$\mathrm{BN}(x) = \gamma\ \left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta \qquad (1)$$

AdaIN (Adaptive Instance Normalization)
In the feed-forward stylization paper the force transfer network contains a BN layer after each con-volutional layer (Ulyanov et al., 2016). According to paper (Ulyanov, Dmitry;Andrea, 2017) found a significant improvement was achieved by replacing the Batch Normalization layer with the IInstance Normalization layer equation in (2):

$$\mathrm{IN}(x) = \gamma \left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta \qquad (2)$$

The AdaIN process works by reading the extraction features from VGG16, namely the map features of the image content and the features of the image style. The following is the AdaIN equation in (3):

$$\mathrm{AdaIN}(x,y) = \sigma(y)\ \left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y) \qquad (3)$$

If the normalization instance can normalize the input to a single style that takes the affine parameter. Then it is possible for adaptation to an arbitrarily given force by using an adaptive affine transform? AdaIN accepts both content input and style input, aligning the mean and channel-wise variance of x to match them. There is a difference when using Batch Normalization and Instance Normalization, AdaIN does not have an affine parameter to study it. However AdaIN adaptively calculates the affine parameter from the style input.

Gram Matrix
Loss measurement recommends correlation between filters as texture information of an image, and it can also be represented by Gram Matrix (Lu, 2015), (Makuracki & Mróz, 2021), (Yaskov, 2016). To get a stylistic representation of the input image, use the feature space originally designed to capture texture information. This feature space builds on filter responses at each layer of the network. It consists of correlations between different filter responses over the spatial level of the feature map. By inputting feature correlations from multiple layers a stationary representation is obtained, then the next step is multi-scaled from the input image, which captures its texture information but not its global settings (Gatys;, Leon A., Alexander S. Ecker, 2015). Gram matrix is the product of a matrix with the transpose of itself.

$$G = X^T X \qquad (4)$$

Because in deep learning we represent matrix columns as features, the Gram Matrix calculation becomes:

$$G = X X^T \qquad (5)$$

Image data with the size of NCHW, then we can do the alignment, and the Gram Matrix using PyTorch.
```
def gram_matrix(X):
    n, c, h, w = X.shape
```
*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 2, April 2022
DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

```
X = X.view(n*c, h*w)              # Flattening
G = torch.mm(X, X.t())
G = G.div(n*c*h*w)                 # Normalization
return G
```

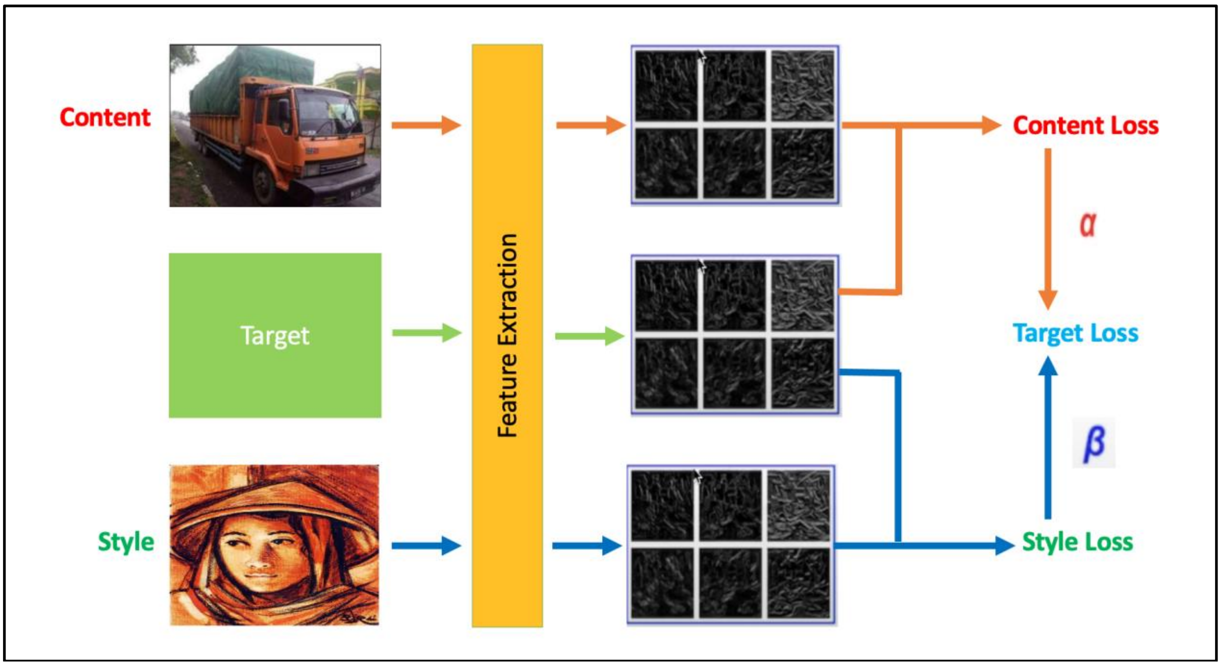

Fig. 3 Content Loss, Target Loss and Target Loss
Source : https://github.com

Content loss how to do the calculation in the following way:
Image Content as input for feature extraction using the VGG16 algorithm to produce a feature map content. Image Style is also used as input for feature extraction using VGG16 to produce a feature map style. Use of Mean Square Error loss between input and target. If there is no constraint on the output style, then the MSE formula between the input and the content image. The MSE loss between the input image feature and the content image feature can be formulated as follows:

$$L_{content}(p, x, l) = \frac{1}{2}\sum_{i,j}(Fij - Pij)^2 \qquad (6)$$

The generated image will be feature extracted using VGG16, the result is a feature map of the image generated by the generator. Image Style is also used as input for feature extraction using VGG16 to produce a feature map style. Use of Mean Square Error loss between input and target.

$$L_{style} = L\sum I = \|\mu(\varphi i(g(t))) - \mu(\varphi i(s))\|_2 \ + \ L\sum I = \|\sigma(\varphi i(g(t))) - \sigma(\varphi i(s))\|_2 \qquad (7)$$

where each of the values represents the layer in VGG-16 that is used to calculate the style loss. The use of Relu11, Relu21, Relu31, and Relu41 layers was carried out at the same weight.

Total loss, the loss is calculated from the sum of the loss content with the loss style. The equation used to calculate the total loss is as follows:

$$\text{Total loss} = \ \ * \text{ Content Loss} + \beta \ * \text{ Style Loss} \qquad (8)$$

Image content, image target(o), and image style is extracted using VGG16 (Song et al., 2019), (Paymode & Malode, 2022), so it can produce content loss and style loss. Total loss is the sum of the content loss and style loss, then looped with backpropagation to the target image(o) and style image. After backpropagation then update the image Target(o).

*name of corresponding author

DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

RESULT

The results of the Style Transfer algorithm on several datasets related to transportation means such as trucks or other transport vehicles.

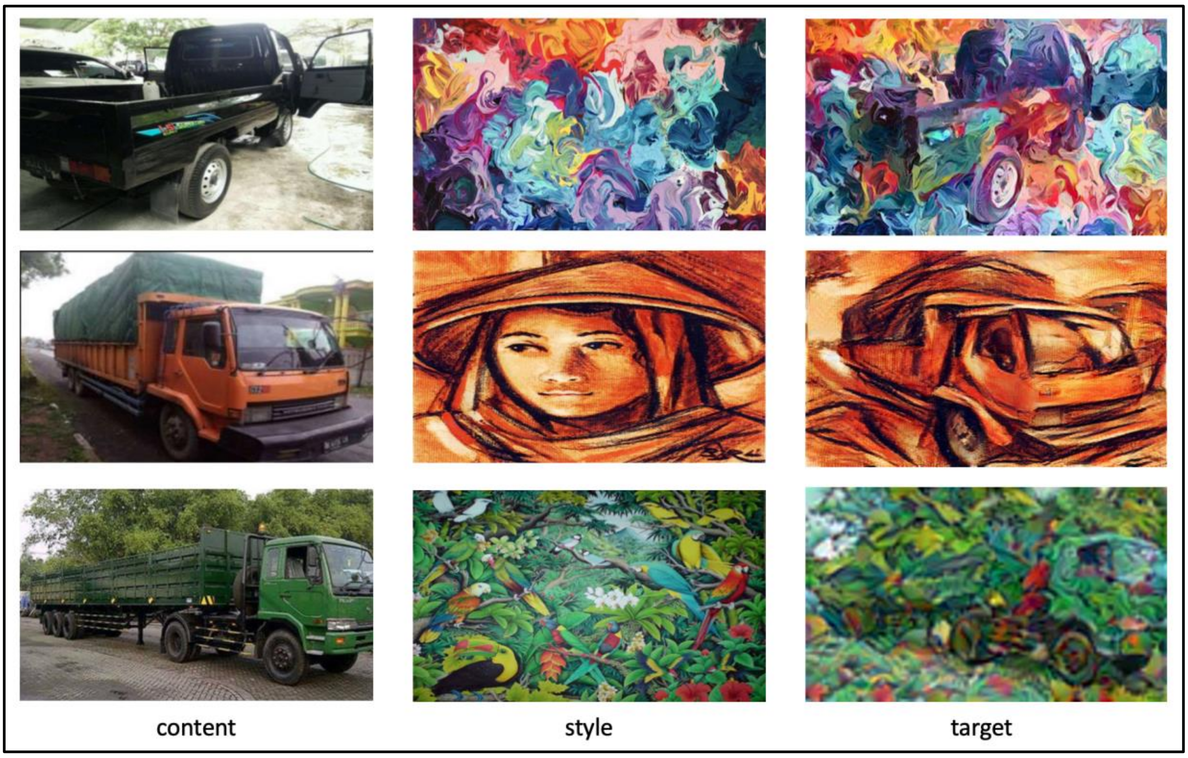

Fig. 4 Result style transfer
Source : researcher property

The style transfer process, it shows that the previous image content is then transferred to the texture of the image style so that the image content changes according to the texture as in the image style. The resulting image target will resemble the image content but use the texture of the image style. This is very different from the Generative Adversarial Network algorithm, where the target image generated will use random. This difference is shown in Fig. 4, where the target image does not change in the image content, only the texture of the image changes.

Results in fig. 4, shows the difference where the results of the target image are different. The target image in image one and image three shows that the content image and image style are not yet integrated. Because the process of moving the image style has not been maximized. Meanwhile for the second image, the process of moving the image style is maximum, thus creating a different image from the image content. The choice of image style is very influential, where the second image style uses rough strokes or textures, when compared to the first and third texture images.

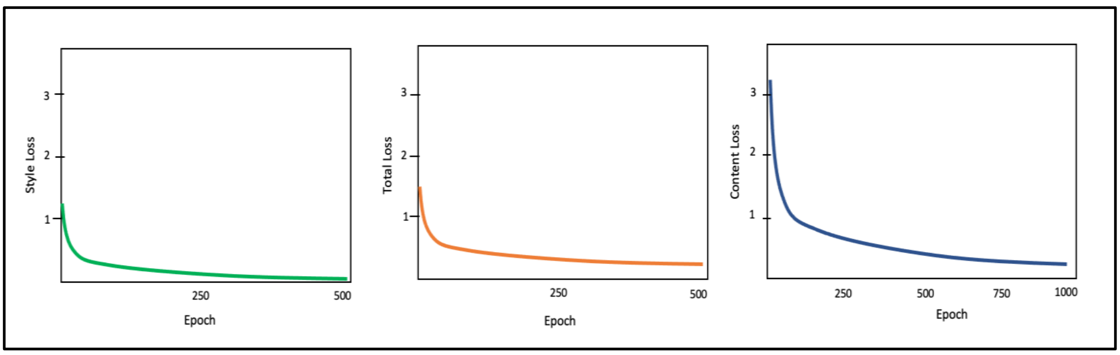

Fig. 5 Loss Style, Loss Total, Loss Content
Source : researcher property

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 6, Number 2, April 2022
DOI : https://doi.org/10.33395/sinkron.v7i2.11375
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Style loss is the process of whether the resulting image is similar to the image style, meaning that the texture has been transferred to the image content. The style loss in epoch 1 is still high, after that it decreases until the value is brought to 0, 13212. Content loss is an image change in the image content that will be added texture by the image style. The content loss at the beginning of epoch 1 is still high until it is close to a value below 0.2342. This indicates that an image change is being made. Total loss is the sum of the content loss and style loss.

The total loss calculation is loading the loss or similarity of the image generated results from the content loss and style loss. Total loss as in fig. 5 experienced a decrease in the total loss value.

DISCUSSIONS

In this section, the researchers can give a simple discussion related to the results of the research trials. This section contains the author's opinion about the research results obtained. Common features of the discussion section include the comparison between measured and modeled data or comparison among various modeling methods, the results obtained to solve a specific engineering or scientific problem, and further explanation of new and significant findings

CONCLUSION

In the style transfer research, the results have been presented above, where the process produces a new image which is a combination of image content and image style. The method used is AdaIN, which takes the average and variance of the VGG16 feature extraction. Transferring the texture image style to the image content results in a new target image with a combination of image content and image styles. There is a difference between Style Transfer and Generative Adversarial Network, where Style Transfer will produce a target image similar to image content by adding an image style, while the Generative Adversarial Network will generate images randomly. Generative Adversarial Network can be done the same as Style Transfer if conditions are made so that the target generated is in accordance with the generator input or Condition Generative Adversarial Network (CGAN)..

REFERENCES

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

Gatys;, Leon A., Alexander S. Ecker, M. B. (2015). *A Neural Algorithm of Artistic Style*.

Guo, L., Zhang, Y., & Li, Y. (2021). An intelligent electromagnetic environment reconstruction method based on super-resolution generative adversarial network. *Physical Communication*, *44*, 101253. https://doi.org/10.1016/j.phycom.2020.101253

Huang, X. S. B. (2017). *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. arXiv:1703.06868v2.

Jiang, M., Zhi, M., Wei, L., Yang, X., Zhang, J., Li, Y., Wang, P., Huang, J., & Yang, G. (2021). FA-GAN: Fused attentive generative adversarial networks for MRI image super-resolution. *Computerized Medical Imaging and Graphics*, *92*(April), 101969. https://doi.org/10.1016/j.compmedimag.2021.101969

Lu, L. (2015). Gram matrix of Bernstein basis: Properties and applications. *Journal of Computational and Applied Mathematics*, *280*, 37–41. https://doi.org/10.1016/j.cam.2014.11.037

Makuracki, B., & Mróz, A. (2021). Coefficients of non-negative quasi-Cartan matrices, their symmetrizers and Gram matrices. *Discrete Applied Mathematics*, *303*, 108–121. https://doi.org/10.1016/j.dam.2020.05.022

Modanwal, G., Vellal, A., & Mazurowski, M. A. (2021). Normalization of breast MRIs using cycle-consistent generative adversarial networks. *Computer Methods and Programs in Biomedicine*, *208*, 106225. https://doi.org/10.1016/j.cmpb.2021.106225

Paymode, A. S., & Malode, V. B. (2022). Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG. *Artificial Intelligence in Agriculture*, *6*, 23–33. https://doi.org/10.1016/j.aiia.2021.12.002

Phon-Amnuaisuk, S. (2019). *Image Synthesis and Style Transfer*. http://arxiv.org/abs/1901.04686

Shim, S., Kim, J., Lee, S. W., & Cho, G. C. (2022). Road damage detection using super-resolution and semi-supervised learning with generative adversarial network. *Automation in Construction*, *135*. https://doi.org/10.1016/j.autcon.2022.104139

Song, Z., Fu, L., Wu, J., Liu, Z., Li, R., & Cui, Y. (2019). Kiwifruit detection in field images using Faster R-CNN with VGG16. *IFAC-PapersOnLine*, *52*(30), 76–81. https://doi.org/10.1016/j.ifacol.2019.12.500

Teramoto, A., Yamada, A., Tsukamoto, T., Kiriyama, Y., Sakurai, E., Shiogama, K., Michiba, A., Imaizumi, K., Saito, K., & Fujita, H. (2021). Mutual stain conversion between Giemsa and Papanicolaou in cytological images using cycle generative adversarial network. *Heliyon*, *7*(2), e06331.

*name of corresponding author

https://doi.org/10.1016/j.heliyon.2021.e06331

Ulyanov, Dmitry;Andrea, V. (2017). *Improved Texture Networks: Maximizing Quality and Diversity inFeed-forward Stylization and Texture Synthesis*. Computer Science.

Ulyanov, D., Lebedev, V., Vedaldi, A., & Lempitsky, V. (2016). Texture networks: Feed-forward synthesis of textures and stylized images. In *33rd International Conference on Machine Learning, ICML 2016* (Vol. 3, pp. 2027–2041).

Yaskov, P. (2016). Controlling the least eigenvalue of a random Gram matrix. *Linear Algebra and Its Applications*, *504*, 108–123. https://doi.org/10.1016/j.laa.2016.03.048

Zheng, Z., & Liu, J. (2020). *P2GAN: Efficient Style Transfer Using Single Style Image*. http://arxiv.org/abs/2001.07466