Art-Style Image Transfering

Oktay UĞURLU [*] **Koray KARA** [*] **Ahmet Deniz GÜNER** [*]

Abstract

Nowadays, Generative Adversarial Networks (GANs) are an emerging technology that is used in both supervised and unsupervised learning. These networks are also capable of producing high-quality data in an efficient way. Image to image translation is one of the core applications of GANs. For instance, a data augmentation that we have used in this project. In this study, we propose the methods that keep the quality of the style transferring high. For this purpose, we are using the CycleGAN that is an extension of the GAN architecture. Generative Networks allow easy mapping between the source image and the target image. It also calculates a loss function to greatly improve the quality of this generated target image. These models are often used to transfer the styles of famous artists to today's paintings. However, GAN modules work with very large data sets. This can cause the training time of the model to increase too much. All pictures of these artists are usually inserted into the model as a train set. In this study, we discussed the possibility of having more than one style in the paintings of the artist to be transferred in style and this situation may affect the model. We examined the effects of this situation on the cycle-GAN model. In our study, we compared the models trained with the whole data set, with the data entering this model trained with clusters after clustering. We used the K-means model and feature extraction methods to cluster the data set. We observed how clustered data affect the success of generative art models, and we aimed to reduce the training times of the models since we use smaller data sets.

1. Introduction

A lot of work has been done in the field of computer vision so far. Image to image translation is one of the core tasks in that area in a way that one of the source images is translated to the target image while keeping the originality of the source image. For this specific purpose of task, Generative Adversarial Network (GAN) is a helpful idea for image to image translation. These networks are actually a combination of two networks that are Generator and Discriminator. We are using CycleGAN (Zhu, Park, Isola, and Efros, 2017) that is a technique for training unsupervised image translation by using GAN architecture for unpaired image-to-image translation. A CycleGAN is composed of 2 GAN's. That means a CycleGAN has 2 generators and 2 discriminators in total. One of the generators takes the images as an input from first dataset, and outputs images for the second dataset. After that, the other generator takes images from the second domain as input and generates images for the purpose of the first domain. Discriminator models are used to determine how appropriately generated images are created and update generator models according to these determinations. In this project, we are also using an additional extension of the CycleGAN that is called cycle consistency loss. This is essentially based on the purpose that the image output of the first generator can be used as the input of the second generator and that the output of this generator also matches the original image. For this purpose, we have calculated the cycle consistency in order to find the differences between real photos as input and transformation of generated Van Gogh images by using the input. Then, we have used the loss values in the calculation of the gradient.

We trained the cycle-Gan model with clustered datasets instead of skewing with the whole dataset and observed the results. Here are the expected results from observations; When we cluster structural as structural, it is better to transform a test picture of that struct; If an artist has more than one style, we can aggregate them based on styles to get better output from GAN and to reduce the training time of the model as we shrink the datasets. In order to do these clustering operations, we had to extract features from the train set we have. We used VGG19 feature extraction methods to do this. VGG is basically a convolutional neural network model. The numbers 19 represent how many convolutional layers there are in the model. This structure is generally used for image object classification(Rashid, Khan, Alhaisoni, Wang, Naqvi, Rehman, and Saba, 2020). In our study, we tried to use these VGG layers for both object-based classification and style-based classification. We have run the K-means clustering method on the features we obtained from feature extraction. With the clusters we obtained, we trained our GAN models and observed the results. K-means is basically an unsupervised learning model that separates the given data set into K clusters with distance-oriented calculations. The Elbow method is one of the most common methods used to find the best K value. We analyzed the selective clustering method with K-means clustering. To summarize briefly, we ran the cycle-GAN model, pre-processed the training dataset by performing feature extraction and clustering, and ran the generative model again and compared the results.

2. Related Work

The research about Style Transfer in the image to image translation field has been an active research field and it is widely used because it allows us to recompose the content of the image into the another style. Style transfer is a computer vision technique that turns the paintings by a famous artist into a reality. There are many studies and attempts of developers and data scientists about style transferring. One of these studies is Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (Zhu, Park, Isola, and Efros, 2017). This article was the preliminary reference for choosing the cycle-GAN as the base model while choosing our generative art model. The study compares some models for image to image translation and makes a detailed study on the cycle-GAN model. In our study, we used the important choices for our base model from this study.

According to this study, which is one of the base studies for Generative Adversarial (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville, and Bengio, 2014) models, two models are trained; these are generative G and discriminator D models. Basically, Discriminator calculates the probability that the data produced by the generator model will conform to the target domain. Generator also tries to produce data similar to the target domain. In this mutual process, the modules train each other. This process takes place within the framework of min-max game logic and models are updated using perceptron. Examples are open to evaluation with both objective and subjective metrics, and they did not need any other structure for these processes.

This study, which is one of the recent studies(Li, Lin, Ding, Liu, Zhu, and Han, 2020), has encountered many generative adversarial network models and has supported the conditional generative adversarial networks (cGAN) model. In this way, they provided controllable image production. The study also offered some approaches to shorten the production time of the cGAN model. Emphasizing that using the CNN model directly does not give a very good result in the GAN model, which is already difficult to train, the study presented a different architecture (NAS). In addition, they compared and evaluated the performance of many generative models such as pix2pix, GauGAN, CycleGAN.

In this study(Isola, Zhu, Zhou, and Efros, 2018), which was published 3 years ago, they conducted research on some common problems related to image-to-image translation. The study has presented a general approach to mapping the pictures on the knowledge that only the image is not learned and the loss values are learned. The study used CNN-based models, the cGAN model and the pix2pix library. The patch-GAN model is used as the discriminator. They investigated the most appropriate learning technique for mapping with different loss values consisting of L, GAN, cGAN and their combinations. They used AMT as the evaluation metric.

In this image style transformation study (Gatys, Ecker, and Bethge, 2015) made using deep neural network, it is aimed to transfer the style features extracted from the source domain to the target domain with feature extraction methods. While doing these, they used the CNN-based VGG-19 feature extraction method. They used the low level layer especially for style transformation and carried out the transfer process. Style parsing is done by the low-level convolutional layer in the network. (Between Conv1-1 and Conv1-5). In this way, the content-based features in the source image were eliminated, and style-based ones were determined and combined.

In this article(Zhang, Ji, and Lin, 2017), which was published on the spread of neural style transfer research, a study was made on the conversion of a painting style to anime style. For this, AC-GAN and U-net generator, which is a derivative of GAN and used for gray scale transmission, are used in the network. In the study, VGG methods were used for the feature extraction method, which is required for style transfer.

In this paper, it is proved that the selective clustering (deng, Tang, Dong, Wu, Deussen, and Xu, 2019) improves the performance of the style transformation to eliminate the unrelated data. It also provides a representative painting selection for an artwork of artists. This study is also related to comparison different clustering to compare with representative painting selection methods. Some of them are K-means, S-cluster and A-cluster.

In this study(Borji, 2018) on the evaluation metrics of GAN models, they conduct a research on pre-existing evaluation metrics for GAN models and present their own methods. The study offers a total of 24 quantitative and 5 qualitative evaluation methods about generative adversarial networks models. Although the study has conducted many GAN studies so far, the low evaluation criteria for them are determined as motivation.

By introducing convulational neural networks to the structure called DCGAN(Radford, Metz, and Chintala, 2016), they aimed to close the neural network usage difference between supervised and unsupervised networks. In this study, DCGAN structure was created by using CNN networks in the unsupervised learning area for images. As activation functions, ReLU and LeakyReLU were used for the generator and the discrimantor, respectively. They used many conventional methods for classification and clustering and tested their models separately on a train set that included people and objects.

3. The Approach

3.1. Problem Defination

In our study, we investigated the effects of the cycle-GAN model, which was trained with very large data sets, when we made changes on the training sets. The cycle-GAN model is used for image-to-image translation. In the research we do in these modules, all the works of the artist who wants to transfer style are given to the model for train. However, whether that artist has more than one style has not been discussed much in the studies. In our study, we sought the answer to the question of how style-based clustering on the artist's works affects the outputs of the cycle-GAN model.

3.2. Dataset

We have used the Van Gogh paintings and Monet paintings in this paper as we discussed earlier. As an additional to the Best Artworks of All Time dataset, we have also used real photo dataset which is also used in Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks dataset. We split our real photo dataset 70% as train dataset, and 30% as validation dataset. We didn't split our painting datasets, because the paintings are not used on validation part.

3.3. Our Proposed Solution

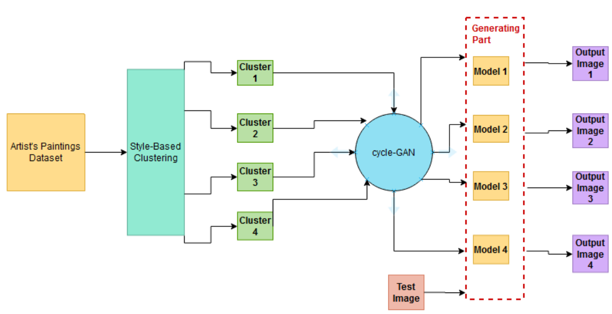

Figure 1. Our Proposed Model

As we stated in the above section, we decided to use the cycle-GAN model as the base model. We examined the behavior of the model by playing with the train sets without directly touching the content of the model. Actually, the part of playing with the train sets was the model we originally presented. We created a model that clusters all the artist's works based on style. For this, we first used the VGG19 feature extraction method. We conducted an extensive literature search to identify the layers that will give us style-based features from features from VGG19. Finally, using the features we identified, we performed style-based clustering on the K-means model. We trained our cycle-GAN model with each cluster and observed the results.

3.4. Our Base Model: CycleGAN

CycleGAN is a method that includes a mechanism that performs automatic image-to-image training translation by not requiring any paired examples. We have used this method instead of using GAN to transform real paintings in a specific artist style. Because GANs have a lot of problem with training that make them quite difficult to use in practice. One of them is something that is called $modecollapse$. That means the generator may completely failed to produce interesting pictures. It can just produce the same picture every time. So, a CycleGAN is essentially a two GAN system that deals with this problem.That means a CycleGAN has 2 generators and 2 discriminators in total. As we can see the Figure 2, Generator G learns how to convert X image to Y image. In our case, X is the real image, and Y is the styled image that we are trying to draw. $(G : X- > Y)$. Generator F learns how to convert Y image to X image. (in our case Y is the styled image that is generated by the first generator G, and X is the image that has been converted back to the real image and should belong to the real domain. $(F- > X)$). The reason we use CycleGAN is not only to take our image

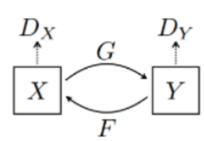

Figure 2. CycleGAN Model

and turn it into something that looks good, but also to go back and check it again to prove that we are not constantly generating the same image. In this work, we have used some terminologies to generate our model. These are explained in the sub-sections below.

Discriminator The discriminator determines whether a styled image that the generator produces is a real or fake image. If the image is determined to be a fake, the generators can get better at changing the real image to styled image of an artist. We have 2 discriminator in this model. One of the discriminator is responsible for determining whether the first generated image is real picture of a styled image or not, and the other discriminator is responsible for determining whether the second generated image belongs to the real domain or not. So, there is a lot of loss functions that is being applied here. The discriminator loss are explained in the 3.4.1

Generator In the model, we have a generator network G that is going to generate a picture of styled images of an artist. We have also another network that is called F which is taking the picture of a styled images that the G generator has created, and turning them back into pictures of real image again. The generator has three parts. These are Encoder, Transformer and decoder.

General Encoder network performs convolution operations and by doing this, it simply encodes the input image from any domain. However, in our case we have used separate encoders because we use CycleGAN.

Decoder network simply performs deconvolution operation and learns how the decoding operation is done on the encoded image from any domain. Also here, we have used separate decoders for our domain.

Translator is actually created by using Residual Networks. That means it simply adds the residue of the previous result.

U-NET Generator To get the result, we have used U-net generator. For our architecture,we have used instance normalization instead of batch normalization because the batch normalization adds extra noise to the phase of training. It is not appropriate for our network. By using instance normalization, instead of normalizing through input features in a training example, we have normalized through each channel for each training example and simply converted an image into a vector and use the same mapping to convert that vector to an image again. This process reduces the distortions that may occur in the picture so that the original structure of the image remains the same.

Cycle-Consistency Loss: As a loss function, we have used this loss to calculate how we wrong or right a particular network is. While we were training our Deep Learning model, we need to some metric that calculate the differences between a real photo as input and transformation of a generated artist styled images by using the input. So we have used Cycle-Consistency loss for this calculation. The purpose of this loss is that the image output of the first generator can be used as the input of the second generator and that the output of this generator also matches the original image. This loss also measures a distance between the image that is entered as input of the first generator, and the image that is produced by the second generator. This function can be defined as:

$$\mathcal{L}_{cyc}(G,F) = E_{x\sim p_{data}(x)}\left[||F(G(x)) - x||_1\right] + E_{y\sim p_{data}(y)}\left[||G(F(y)) - y||_1\right]$$

formula above, the images are recreated by $F(G(x))$ and $G(F(y))$ will be similar to x and y.

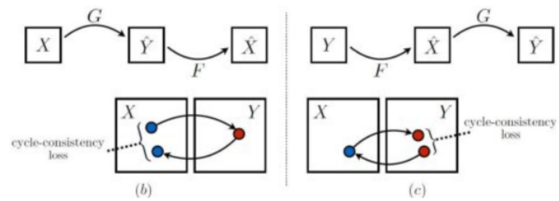

Figure 3. Cycle Consistency Loss Mechanism

3.4.1. Other Loss Functions:

After training our Deep Learning model, we have also used different types of loss functions than cycle-consistency loss to track how well the model learned. These are explained below separately.

Identity Loss: We have used this identity loss function to get benefit of its mapping feature. It has helped us to preserve the image colors while doing operation over the input paintings. The related formula is shown in below.
$IdentityLoss = |G(Y) - Y| + |F(X) - X|$
However, we have got the most smooth loss by using Cycle-Consisteny loss because this loss function allows the network to learn the correct mapping.

Discriminator Loss: While we were training our discriminator, it tries to classify both the real image and and the fake image that are produced by the generators. It basically punishes itself when any misclassification occurs and maximizes the $BinaryCrossentropy$ loss function that is called discriminator loss.

Generator Loss: While the generators are trained, it gets some random picture(noise), and generate an image output by using this noise. This image output goes back to the discriminator and the discriminator basically classifies the output of image as real or fake image. Then, the generator loss is calculated according to the determination(classification) of the discriminator. This time, it should trick the discriminator successfully. Otherwise, it have to punish itself for the misclassification that determined by the discriminator. While the generator is being trained, the $BinaryCrossentropy$ loss function is minimized this time.

Adam Optimizer: The Adam optimization algorithm is one of the algorithms that has proven to perform very well in a wide variety of deep learning models. While we were training our Deep Learning model,in order to update weights of the network, we have used Adam Optimizer instead of the classical stochastic gradient descent that is because it is easy to implement and also computationally efficient for the problems of very noisy and sparse gradients.

3.5. Data Augmentation

Since the generative models are muchly data-hungry algorithms, some data augmentation methods are used. According to these methods, images are randomly cropped and randomly flipped. By flipping, the model is trained with more perspective on different batches.

3.6. Clustering

Because we transform a normal photograph into an artist based style, the style of the artist's paints domain takes an important role in Cycle-GAN. Therefore, we implemented clustering methods to cluster an artist domain into subdomains, and we trained our model to transform a normal image with more style. In this part, we implemented 2 different clustering methodology to use and compare their efficiency clustering of dataset for Cycle-GAN.

Feature extraction: Because this is a sort of image clustering problem, feature extraction takes a substantial role in the clustering. The VGG19 is a mainly used neural network model to extract features from images with the ImageNet database. The ImageNet is the image database which is commonly used for computer vision purpose. We have used the VGG19 model which is pretrained on the ImageNet. It has 24 layers. There are 5 max pool layers in these layers, and from the start layer to the fifth max pool layer has been used as filters to extract features of the image matrix which are the convolutional layers and max pool layers. After the fifth max pool layer, there are 3 fully connected layers which are used classification purpose. We have used the first 21 layers fully and partially for our 2 different feature extraction type which are style-based and object-based features.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 300, 300, 3)	0
block1_conv1 (Conv2D)	(None, 300, 300, 64)	1792
block1_conv2 (Conv2D)	(None, 300, 300, 64)	36928
block1_pool (MaxPooling2D)	(None, 150, 150, 64)	0
block2_conv1 (Conv2D)	(None, 150, 150, 128)	73856
block2_conv2 (Conv2D)	(None, 150, 150, 128)	147584
block2_pool (MaxPooling2D)	(None, 75, 75, 128)	0
block3_conv1 (Conv2D)	(None, 75, 75, 256)	295168
block3_conv2 (Conv2D)	(None, 75, 75, 256)	590080
block3_conv3 (Conv2D)	(None, 75, 75, 256)	590080
block3_pool (MaxPooling2D)	(None, 37, 37, 256)	0
block4_conv1 (Conv2D)	(None, 37, 37, 512)	1180160
block4_conv2 (Conv2D)	(None, 37, 37, 512)	2359808
block4_conv3 (Conv2D)	(None, 37, 37, 512)	2359808
block4_pool (MaxPooling2D)	(None, 18, 18, 512)	0
block5_conv1 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block5_pool (MaxPooling2D)	(None, 9, 9, 512)	0

Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0

Figure 4. Summary of VGG19 for feature extraction

Style-Based Feature Extraction: The style-based features demonstrates the texture and colour features of images generally. If we say in more art manner, mainly used colours, brushmarks, colour tones, and used paint shows us the artist's style. As discussed in Selective Clustering (deng, Tang, Dong, Wu, Deussen, and Xu, 2019) and Neural Algorithm of Artistic Style (Gatys, Ecker, and Bethge, 2015) papers, all of the top convolutional layers are responsible filter the texture-based features, while others are filter more positional information of the texture which is mostly used for content-based feature extracting. In our model, these layers are block1_conv1, block2_conv1, block3_conv1, block4_conv1, and block5_conv1. We extracted the activation values of these layers, and multiply these with the gram matrix to extract more intensive style information. After getting these layers, we have vectorized and composed these layers to use in the K-means algorithm.

Object-Based Feature Extraction: Our object-based approach directly based on the usages of last activation output coming from the max pool layer of the VGG19 model. After this feature extraction, we have used these features as another K-means model.

K-means We have used the K-means algorithm which is widely used in most image clustering problem. When we researched the proper way to cluster arts according to their style, we found that the K-means gives good results which have been referred to in Selective Clustering paper.

Elbow method: After extracting the features, the other important process for the clustering is finding the number of clusters. The elbow method is one of the evaluation methods to choose the number of clusters which is commonly used when the number of clusters is not exactly known. According to this method, we train the K-means method with different sequential k values and calculate the sum of the squared distances (WCSS) between data points and centroids for each model. After running for all k value, these values are plotted on a graph and tried to find the most decreasing interval which looks like an elbow. The final point of the decreasing curve points to the best k value for our dataset.

4. Experimental Result

We studied over Van Gogh paintings firstly to evaluate first results of our approach. As addition, we have run our model over Monet paintings. Firstly, we discussed our results over Van Gogh.

4.1. Results for Object-Based Clustering

Our first study is trying to cluster the artist's works into sub-categories. For this purpose, we research different feature extraction techniques for images. At the start, we applied the VGG19 feature extraction method which clusters the images based on the content. You can see images belongs to the clusters here.

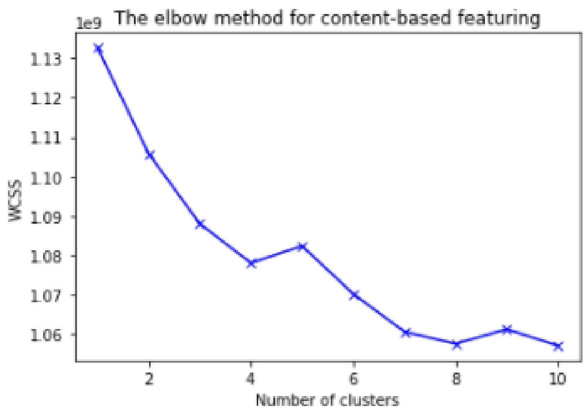

Figure 5. Elbow method for Object-Based Clustering

When we applied the object-based clustering, we clustered into 4 different groups according to the elbow method. The clusters are mostly grouped as human portraits, flower paintings, and other scene paintings. However, the Cycle-GAN was not performed very well over these clusters. The transformed images are mostly distorted. The reason for this, these clusters contains different styles of the artist.

4.2. Results for Style-Based Clustering

After our researches, we implemented the style-based feature extraction method as we mentioned on section 3. We first tried to find the best k parameter for style-based clustering. You can see that the elbow finishes at 3 in Figure-5. Therefore, we chose the number of clusters as 3 for style based clustering.

After applying the style-based clustering, results are grouped into 3 different styles. The most bright coloured paintings are grouped into cluster 1. These are the most represents Van Gogh's style, The second cluster includes more dark tones images. The last cluster contains less dataset which

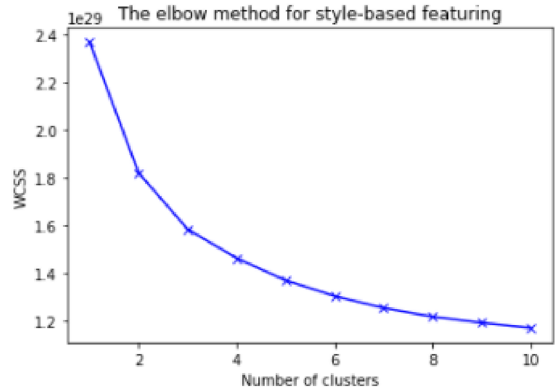

Figure 6. Elbow method for Style-Based Clustering

is mostly sketch works of Van Gogh. You can see images belongs to the clusters here.

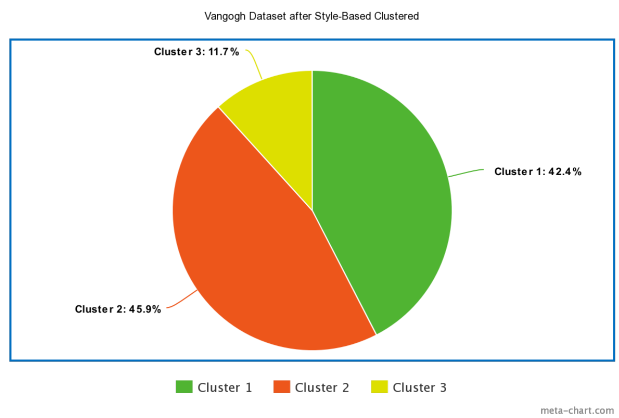

Figure 7. Vangogh Dataset after Style-Based Clustered

4.3. Evaluation and Comparing The Results

We trained our model on these 3 different clustered datasets, and full dataset. To make the evaluation of our model, we have used two different metrics that consist of qualitative and quantitative metrics. We used preference judgement as the qualitative metric, and Frechet Inception Distance scoring as the quantitative metric. The details of these methods are explained in the subsections below.

4.3.1. Frechet Inception Distance (FID) Scoring:

We have used this scoring metric to compare the images which are generated from CycleGAN and images from our train datasets. While we are using in implementation, we basically transform the images to feature vectors and calculate the distance between the images that are generated

from the methods that we have used. This calculation gives us a score that tells how similar the two groups are. The lower scores implies that the two groups of images are more similar. If we have a score of 0.0, it means that the two groups of images are identical of each other. We have used this FID score to evaluate the quality of our images that are generated from CycleGAN.
In order to calculate this score, we have loaded a pre-trained model that is called as Inception v3 model because the FID score uses that model. It means that the FID(Borji, 2018) score is calculated based on this loading. As an implementation details of the Frechet Inception Distance, the function that we have used to obtain the FID scores gets the parameters of activations for clustered and generated images and returns the FID score.

We compared the FID scores for 4 different CycleGAN modes. The first model was directly trained over full dataset. The others are trained on 3 different style-based clustered sub-datasets. You can see the results below:

FID Scores over Van Gogh Dataset	
Dataset	FID Score
Full Dataset	124.683
Cluster 1	122.713
Cluster 2	175.632
Cluster 3	219.317

According to the FID scores, the features of the transformed images according to the cluster-1 dataset are more closer each other than others, because the cluster-1 includes more specific style paintings.

4.3.2. Preference Judgement:

According to the preference Judgement method, a group of participant rates the generated images by generative models. In our works, we firstly worked with ordinary people who competent about visual art. However, style-based image transformation domain needs expert point of view to critize its success, we select 6 student from faculty of art. We prepared 4 different transformed results for each 10 different real images. The participants rated these images over 10 according to their similarities with any styles of Van Gogh's paintings. The most accurate rates has been given to our model which is trained over the Cluster 1. However, :

Van Gogh Preference Judgement Rates (%)	
Dataset	Accuracy
Full Dataset	40 %
Cluster 1	55 %
Cluster 2	31 %
Cluster 3	21 %

For all preference judgement scores for 10 different images, you can see here.

Model Training Time According To Dataset	
Dataset	Model Training Time
Full Dataset	278.49 min
Cluster 1	167.43 min
Cluster 2	184,11 min
Cluster 3	141.03 min

We train our CycleGAN models over 50 epoch. As shown in the table above, the training times according to the full-data and all of the clusters are given in terms of the minutes.

4.4. Monet Results

After studying on Van Gogh's paintings, we also run our model over Monet paintings. However, results of the Monet is not accurate like Van Gogh based style transformation. Most of the images are not transformed into a style, they are just distorted. The cause of this, Monet has not various styles. Therefore, the FID scores are more bigger than the model which is run over full dataset. You can see the FID scores of the Monet below:

FID Scores over Monet Dataset	
Dataset	FID Score
Full Dataset	93.330
Cluster 1	160.517
Cluster 2	105.991
Cluster 3	135.037

4.5. Implementation Details and Running Environment

During our study, we made our codes in the python programming language. We also used scikit-learn, Tensorflow, Keras, scipy libraries. We ran our codes in Google Colaboratory and made all our measurements here. We used Google Colaboratory's GPU support, especially for time-related measurements, and the hardware we use is as follows: GPU: 1xTesla K80 , compute 3.7, having 2496 CUDA cores , 12GB GDDR5 VRAM - CPU: 1xsingle core hyper threaded Xeon Processors @2.3Ghz ie(1 core, 2 threads) - RAM: 12.6 GB Available

5. Conclusion

We started our research process by primarily investigating generative adversarial modules and types. Afterward, we determined the cycle-GAN as the base model, investigated its details, and ran this model with the usual methods, that is, with all the train data. At the last stage, we ran the cycle-GAN again with each cluster by performing style-based clustering of the train set and observed the results on Van Gogh artworks. We concluded that the solution we proposed for painters with more than one style yielded better results in the participant survey. Moreover, if the painters have several styles which are not connected to their actual styles, our model can find more represented styles by collecting them into one cluster as we have seen on the Van Gogh paintings. By reducing the training datasets for paintings, the time complexity is reduced. This is also increase the performance of the model in this way. However, if the artist has not present to the model enough data after the clustering process, the model is not trained efficiently.

Some artist's works are also more focused on a style or concepts like the Monet. The Monet paintings composed on nature objects like trees, seasides and rocks which are focused only one style. Despite of applying the style-based clustering, the clusters are created based on object paintings. Therefore, the transformed images includes some irrelevant objects from the paintings like trees and rocks as you can see on Figure-8. This is the weakness of our model. On the other hand, if the artist has lots of style, the cluster numbers increase and training dataset decrease on each cluster.

Figure 8. Transformed by Monet-based Style-Based Clustering

6. Future Works

As we mentioned during the study, we could only test the effects of clustering the train set on the cycle-GAN model. In the future, we can also investigate the effects of this hypothesis on GAN, cGAN, wGAN, or other generative models. In addition, we used the VGG-19 model as a feature extraction model in our study, the main reasons for this were that it is a very common and good model and we have enough resources to extract style-based invoices from VGG-19. Another step that can take this work forward is to use a more advanced feature extraction model and detect and use style-based features in that model, for example, this could be ResNet50. We used the K-means model in the clustering phase of the study because it was a successful model in image clustering problems, it was simple and fast to use. In future work, one of the goals we can set ourselves will be to try clustering using other models. Examples of these models are S-cluster, A-cluster, selective clustering.Also, instead of using clustering directly, we can try using the Graph-Based Representativity Learning (Deng, Tang, Dong, Ma, Huang, Deussen, and Xu, 2020) method, this method assigns the artist's images a score according to how they represent the artist's style so that the images that show the artist's style the most are found. It does this by using deep learning methods. Finally, we can do a more detailed study on the evaluation metric of generative models and develop our quantitative metrics.

References

A. Borji. Pros and cons of gan evaluation measures, 2018.

Y. deng, F. Tang, W. Dong, F. Wu, O. Deussen, and C. Xu. Selective clustering for representative paintings selection. *Multimedia Tools and Applications*, 78, 07 2019. doi: 10.1007/s11042-019-7271-7.

Y. Deng, F. Tang, W. Dong, C. Ma, F. Huang, O. Deussen, and C. Xu. Exploring the representativity of art paintings. *IEEE Transactions on Multimedia*, pages 1–1, 2020. doi: 10.1109/TMM.2020.3016887.

L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. URL http://arxiv.org/abs/1508.06576.

I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

M. Li, J. Lin, Y. Ding, Z. Liu, J.-Y. Zhu, and S. Han. Gan compression: Efficient architectures for interactive conditional gans, 2020.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

M. Rashid, M. Khan, M. Alhaisoni, S. Wang, S. Naqvi, A. Rehman, and T. Saba. A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection. *Sustainability*, 12:1–24, 06 2020. doi: 10.3390/su12125037.

L. Zhang, Y. Ji, and X. Lin. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier GAN. *CoRR*, abs/1706.03319, 2017. URL http://arxiv.org/abs/1706.03319.

J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017. URL http://arxiv.org/abs/1703.10593.