Paint Anything

Karthikeya Jayarama, Varun Velankar, **Project Mentor**: Alok Patwa

***Abstract*—Neural Style Transfer (NST) has emerged as a powerful technique for generating artistic images by combining the content of one image with the style of another. This report presents a comparative analysis of three prominent methods used in NST: CycleGAN, VGG, and CNN (ResNet). CycleGAN leverages generative adversarial networks to achieve style transfer without paired training data, while VGG-based methods utilize pre-trained convolutional neural networks for feature extraction and style representation. CNN (ResNet), known for its deep residual learning framework, offers a robust architecture for capturing intricate style features. By evaluating these methods based on perceptual quality, computational efficiency, and scalability, we aim to identify the strengths and limitations of each approach. Our findings provide insights into the optimal choice of NST techniques for various applications in digital art and image processing. Image to mage transfer is also a distinct concept altogether but has decent overlap with neural style transfer.**

I. Introduction

Neural Style Transfer (NST) is a method applied nowadays to manipulate the content from one image into another with the style. Its initial presentation goes back to Gatys et al. [1] in the year 2015. NST has been mainly developed for the purposes of computer vision and digital art, as it produces stylized images that are visually appealing. The basic idea of NST is learning the content and style of the features extracted from images using deep learning models, especially CNNs.

Over the years, three notable approaches have been developed that help enhance the efficiency and quality of NST: CycleGAN, VGG-based methods, and CNN with Residual Networks (ResNet). Each of these methods has its own merit and poses its own challenges, specifically in the context of style transfer.

One of the generative adversarial networks that offer this and allow style transfer without the requirement of paired training data is CycleGAN. It comprises a cycle-consistency loss that secures the quality of the created images, ensuring that they are faithful to the content of the input images but have the requisite style. On the other hand, VGG-based methods use pre-trained convolutional neural networks, such as VGG networks, which are likely to effectively extract deep features modeling both content and style. Those methods optimize a loss function in the regularization term, which maintains the balance between content and style.

The famous ResNet architecture of CNN, presenting a deep residual learning framework, resolves this vanishing gradient problem that arises in a deep neural network with a large number of layers by means of skip connections. Thus, the deeper networks of ResNet learn rich information and, hence, are expected to be able to recognize intricate information about content and style, which may lead to better style transfer results. In this paper, we do a comparative analysis of CycleGAN, VGG-based methods, and CNN (ResNet) for NST. The methods are considered with respect to some metrics like the quality of the produced image and their perceptual quality, computational resource implications, and scalability. This comparative analysis will help in understanding the merits and shortcomings of the two approaches, which will further provide invaluable insights with regard to their appropriateness for different applications in the domain of digital art and image processing.

II. Background

Vanilla Neural Style Transfer

Neural Style Transfer (NST) emerged as a novel technique in computer vision aimed at recomposing images in the style of other images. Originally introduced by Gatys et al., the method leverages the feature extraction capabilities of convolutional neural networks (CNNs). By using a pretrained network like VGG, NST applies the style of a reference image to the content of a target image through a content loss and a style loss. The content loss ensures that the input content is preserved in the output, while the style loss ensures that the style of the reference image is effectively transferred. Neural Style Transfer (NST) emerged as a novel technique in computer vision aimed at recomposing images in the style of other images. Originally introduced by Gatys et al., the method leverages the feature extraction capabilities of convolutional neural networks (CNNs). By using a pretrained network like VGG, NST applies the style of a reference image to the content of a target image through a content loss and a style loss. The content loss ensures that the input content is preserved in the output, while the style loss ensures that the style of the reference image is effectively transferred.

VGG (Visual Geometry Group Network)

The VGG network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Developed by Simonyan and Zisserman [2], VGG was one of the first to use smaller filter sizes to increase depth, improving its ability to capture fine details in images. It has been widely used as a feature extractor in many image processing tasks due to its excellent performance in capturing image textures and patterns. The VGG network is characterized by its simplicity, using only 3×3 convolutional layers stacked on top of each other in increasing depth. Developed by Simonyan and Zisserman, VGG was one of the first to use smaller filter sizes to increase depth, improving its ability to capture fine details in images. It has been widely used as a feature extractor in many image processing tasks

due to its excellent performance in capturing image textures and patterns.

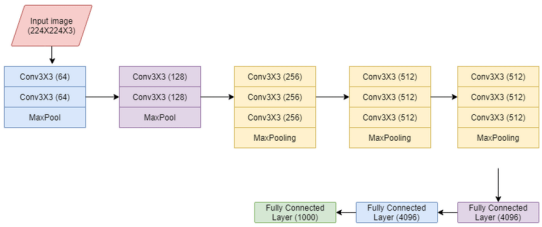


Fig. 1: VGG Block Diagram [2]

ResNet (Residual Networks)

ResNet, introduced by He et al. [3], addresses the problem of training very deep neural networks. As networks deepen, they tend to suffer from vanishing gradients, leading to a saturation in accuracy and then rapid degradation. ResNet introduces "skip connections" or "shortcuts" to allow gradients to flow through the network directly, bypassing several layers at a time. This architecture enables training of networks that are substantially deeper than those previously feasible, improving performance on tasks requiring very deep feature hierarchies. ResNet, introduced by He et al., addresses the problem of training very deep neural networks. As networks deepen, they tend to suffer from vanishing gradients, leading to a saturation in accuracy and then rapid degradation. ResNet introduces "skip connections" or "shortcuts" to allow gradients to flow through the network directly, bypassing several layers at a time. This architecture enables training of networks that are substantially deeper than those previously feasible, improving performance on tasks requiring very deep feature hierarchies.

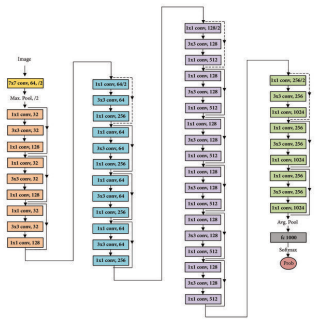


Fig. 2: ResNET 50 architecture block diagram [3]

HYPOTHESES

Given the distinct characteristics of Vanilla Neural Style Transfer, ResNet, and VGG, hypotheses regarding their overall performance can be drawn based on their architectural strengths and typical applications:

- **Hypothesis 1:** For tasks involving style transfer or applications requiring high-level texture synthesis, **VGG** might perform better due to its depth and effective feature extraction at various scales. This makes it particularly suitable for capturing and manipulating image styles.
- **Hypothesis 2:** In tasks that require recognizing objects across very large and deep networks, such as image classification across hundreds of classes, **ResNet** is likely to outperform VGG and NST. ResNet's ability to train deeper networks without degradation in performance due to its skip connections makes it ideal for complex classification tasks.
- **Hypothesis 3:** For combining style and content where the transformation of texture and adherence to high-level semantics are crucial, the **Vanilla Neural Style Transfer** using VGG as a backbone might provide the most visually pleasing results. However, its dependence on pre-trained feature representations may limit flexibility compared to ResNet's adaptable features in other domains.

III. CONTRIBUTIONS

- Our main efforts are towards understanding how to get the best neural style transfer for the given image. We compared two different techniques to which are the vanilla style transfer where we take the pre-trained model and fine-tune it and another approach is to do the image to image translation using CycleGAN and train it from scratch using Monet2Photo dataset [4].
- In the original CycleGAN paper [4] have shown the results for ResNet Generator but in our implementation, we also modified the Generator architecture to UNet [5]. From the Figure 4 we can see that UNet gives better transfer results than ResNet architecture.
- On the application side, we have implemented an OpenCV webcam application for real time inference 5.

IV. METHODOLOGY

The goal of Neural Style Transfer is to minimize the content difference between a content image C and a generated image G and the style difference between a style image S and the generated image G. The loss function is defined as:

$$\mathcal{L}(C, S, G) = \alpha \mathcal{L}_{\text{content}}(C, G) + \beta \mathcal{L}_{\text{style}}(S, G)$$

where α and β are weights that balance the contribution of the content and style losses, respectively.

Content Loss

The content loss is a function that represents a measure of dissimilarity between the content of the content image C and the content of the generated image G. It is usually defined as the squared error loss between the feature representations of C and G:

$$\mathcal{L}_{\text{content}}(C, G) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

where F^l and P^l are the activations of the l-th layer of a pretrained CNN for the generated and content image, respectively.

Style Loss

The style loss is defined using the Gram matrix, which captures the style of an image. The loss is the mean squared error between the Gram matrices of the style image S and the generated image G:

$$\mathcal{L}_{\text{style}}(S, G) = \sum_{l=1}^{L} w_l \cdot \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

where G^l and A^l are the Gram matrices corresponding to the activations of the l-th layer of the CNN for the generated and style images, respectively, and w_l are weighting factors for each layer.

VGG Networks

VGG networks have been extensively used in style transfer due to their simple architecture and effectiveness in capturing image content and style. The typical choice is VGG-19, trained on the ImageNet dataset.

Architecture of VGG Networks

The hallmark of VGG architecture is its simplicity and depth. The network uses repeatedly stacked convolutional layers with small receptive fields, followed by max-pooling layers. Here is a breakdown of its key architectural features:

Convolutional Layers

VGG networks utilize 3×3 convolution filters with a stride of 1 pixel to preserve spatial resolution after each convolution:

$$\text{Output size} = (\text{Input size} - \text{Filter size} + 2 \times \text{Padding}) / \text{Stride} + 1$$

For VGG, padding is typically set to 1 to maintain the size of the output equal to the size of the input.

Max-Pooling Layers

Max-pooling is performed over a 2×2 pixel window, with stride 2:

$$\text{Output size} = \frac{\text{Input size}}{2}$$

This operation reduces the spatial dimensions (height and width) by half, effectively downsampling the feature maps and reducing the computation for upper layers.

Fully Connected Layers

After several convolutional and max-pooling layers, VGG networks use three fully connected layers. The first two have 4096 channels each, and the third performs the final classification and has 1000 channels (one for each class of the ImageNet challenge).

Activation Functions

The Rectified Linear Unit (ReLU) is used as the activation function throughout the network, which adds non-linearity to the model allowing it to learn more complex patterns:

$$f(x) = \max(0, x)$$

Stacked Convolutional Layers

One of the innovative aspects of VGG is the stacking of two to four convolutional layers before each max-pooling step. This design allows the network to learn more complex features at each level of the hierarchy before spatially downsampling them. The depth of the network, which in some configurations reaches 19 layers, enables the extraction of high-level features.

CycleGAN

CycleGAN [4] introduces a way to perform image-to-image translations without paired examples. It is designed to capture special characteristics of one image collection and figure out how these characteristics could be translated into the other image collection, all in the absence of any paired training examples. Adaptability Across Domains: CycleGANs are exceptionally versatile, capable of being applied to a variety of domains beyond mere image translation, including tasks in video generation, style transfer for music, and even biological image transformations where paired examples are rare or non-existent.

Unsupervised Learning Capability: One of the most significant advantages of CycleGANs is their ability to perform unsupervised learning, leveraging unpaired data to understand and translate intricate patterns and styles between two unrelated datasets without explicit paired input-output examples.

Preservation of Key Attributes: CycleGANs are designed to preserve key attributes between the source and target domains, ensuring that important structural details remain intact during the translation process, which is crucial for applications like medical imaging where accuracy and detail are paramount.

Contribution to Artistic Endeavors: Artists and designers have adopted CycleGAN technology to push the boundaries of digital art, enabling the creation of complex hybrid artworks that blend features from multiple styles or eras seamlessly, thus expanding creative possibilities.

Impact on Data Augmentation: In machine learning, especially in situations with limited data, CycleGANs provide a powerful tool for data augmentation. They can generate realistic, varied data from existing datasets, thus enhancing the robustness of models trained on such augmented datasets.

Architecture of CycleGANs

The CycleGAN architecture comprises two main components: two Generative Adversarial Networks (GANs) and a cycle consistency loss that links them. Each GAN includes a generator and a discriminator:

- Generator $G : X \to Y$ — Transforms an image from domain X to domain Y.
- Discriminator D_Y — Discriminates between images from domain Y and transformed images $G(X)$.
- Generator $F : Y \to X$ — Transforms an image from domain Y to domain X.
- Discriminator D_X — Discriminates between images from domain X and transformed images $F(Y)$.

Adversarial Loss

The adversarial loss for the generators is given by:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) &= \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &\quad + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))] \\ \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) &= \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_X(x)] \\ &\quad + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y)))]\end{aligned} \tag{1}$$

Cycle Consistency Loss

To ensure that each image translation cycle (from X to Y and back to X, and vice versa) is consistent, the cycle consistency loss is used:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) &= \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &\quad + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]\end{aligned} \tag{2}$$

Total Loss

The total loss function for training the CycleGAN model combines the adversarial losses and the cycle consistency loss:

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) &= \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ &\quad + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F)\end{aligned} \tag{3}$$

where λ controls the relative importance of the cycle consistency loss.

Training CycleGANs

Training involves alternating updates to the generators and discriminators:

- Update the generators G and F by minimizing $\mathcal{L}_{\text{GAN}}$ and $\mathcal{L}_{\text{cyc}}$.
- Update the discriminators D_X and D_Y to maximize $\mathcal{L}_{\text{GAN}}$.

Image to Image vs Neural Style Transfer

CycleGANs have been used successfully in various applications such as photo enhancement, style transfer, image synthesis, and domain adaptation. Their ability to learn from unpaired data makes them particularly useful for tasks where paired training data is scarce or difficult to obtain.

Comparison

- **Performance:** VGG-based methods are straightforward but might not handle complex transformations between drastically different styles. CycleGAN excels in cases where the style transfer involves significant changes in texture and overall style.
- **Flexibility:** VGG is less flexible as it requires changes at deeper network layers for significant style shifts. CycleGAN is more flexible due to its ability to learn mappings without paired examples.
- **Realism:** CycleGAN generally produces more realistic transformations compared to VGG when transferring styles between unpaired images.

Introduction to U-NET

U-Net [5] is a convolutional neural network developed primarily for the task of biomedical image segmentation. The architecture is shaped like a U, which includes a contracting path to capture context and an expanding path that enables precise localization.

Architecture

The U-Net architecture consists of a contracting path, a bottleneck, and an expansive path. Refer to Figure 3 for a detailed block diagram.

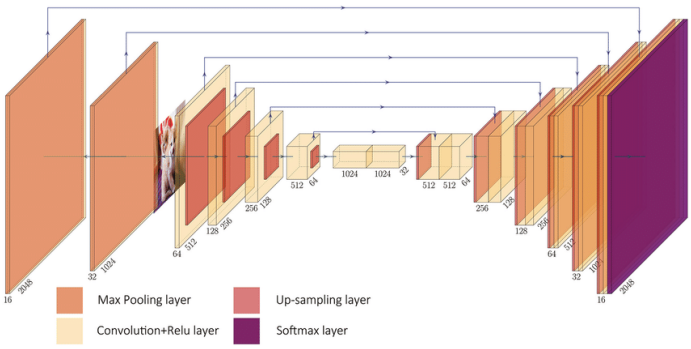


Fig. 3: U-NET Architecture [5]

Detailed Explanation

Contracting Path

The contracting path involves repeated application of convolutions, each followed by a rectified linear unit (ReLU) and a max pooling operation with stride 2 for downsampling. At each downsampling step, the number of feature channels is doubled.

Bottleneck

This section transitions between the contracting and expanding paths and includes two consecutive convolutional layers.

Expanding Path

The expanding path includes several steps of upsampling and concatenation followed by regular convolutions. Each upsampling step increases the resolution of the output.

U-Net vs ResNet Generators

The U-Net [5] and ResNet [3] architectures, both potent in their respective domains, cater to distinctively different tasks within the realm of deep learning applications. U-Net, with its symmetrical encoder-decoder structure, excels in detailed, pixel-wise tasks such as medical image segmentation, where precision in localizing each pixel is paramount. Its architecture effectively combines contextual and localized information through skip connections that help preserve spatial hierarchies, making it ideal for applications requiring precise segmentation details. On the other hand, ResNet's design addresses the challenge of training very deep networks through its innovative use of residual blocks, allowing it to perform effectively

Fig. 4: Experimental Results

Model Type	GPU Used(Training)	No. of Parameters	Training Time	# of Epochs	Batch Size	Crop Size
VGGNet-19	Tesla P4	555328	2 mins/image	10	1	128 x 128
ResNet CycleGAN	Tesla P100	7.84 M	2.1 hrs	60	32	128 x 128
UNet CycleGAN	Tesla P100	11.38 M	2.3 hrs	42	32	128 x 128

TABLE I: Table containing model performance metrics following training and validation runs, along with select model parameters

across a broader spectrum of tasks beyond segmentation, such as image classification and feature extraction for complex vision tasks. These residual blocks help mitigate the vanishing gradient problem, enabling the network to learn more complex and abstract representations as depth increases. While U-Net is specifically tailored for high-resolution output that closely aligns with the input structure, ResNet offers versatility and efficiency in handling diverse and computationally intensive tasks that benefit from deeper network architectures.

V. Results

A. Vanilla Neural Style Transfer

In the previous section we introduced Vanilla Neural Style Transfer using VGG-19 model. We used the pretrained VGGNet-19 on ImageNet dataset and we fine tuned the model using our content and style images with Content Loss and Style Loss layer respectively. We used a Normalization Layer with mean of [0.485, 0.456, 0.406] and standard deviation of [0.229, 0.224, 0.225] of all three channels before passing down to VGG-19 layers. The content and Style Image were set to the size of 128×128 with RGB channels hence input image to the model is of size $(1 \times 3 \times 128 \times 128)$ where 1 is the batch size. The total number of trainable parameters used is 555,328. The summary of training setup of VGGNet and other models are presented table.

LBFGS Optimizer

The LBFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) algorithm is an optimization method designed for parameter estimation in machine learning models, particularly effective for large-scale problems. It belongs to the quasi-Newton methods category and aims to approximate the inverse Hessian matrix to guide the search for a function's minimum. We have used this as our optimizer to fine tune our existing model to generate rich style images. We train the model for 10 epochs for each content image and

α and β defined in the previous section we have defined has the values of 1 and 1000000.0.

B. CycleGAN (ResNet Generator and UNet Generator)

As in the original paper the authors have introduced about CycleGAN generator using ResNet architecture. In our implementation we tried out on UNet as the Generator get more rich image to image translation features. The input and output shape of the image and painting is $(32 \times 3 \times 128 \times 128)$ where 32 is the batch size 3 is the input and output channels of both ReSNet and UNet Generator. The learning rate used is 0.0002 and we used Adam Optimizer as suggested in the paper. The hypermeter λ is 0.99 as decribed in the previous section 2.

Data Augmentation

In order to load to the dataloader we need to resize all the Images into equal sizes. so we choose the input size as described above with 128×128. We used random crop based on the output size. We also accounted for Random horizontal flip for real time application we get random noise input.

We trained the CycleGAN with the whole Monet2Photo dataset which is provided by UC Berkeley [4] for 100 epochs. As with the VGG-19, we just used a single Monet Painting to do the efficient Style Transfer. We can concluded that training VGG-19 is ineffective to capture accurate style of Monet painting whereas the CycleGAN is able to get good predictions based on the output image.

C. WebCam

Once we have trained the model we have written a custom OpenCV application to validate the results we got. We saved our model as .pth file and load it back for inference. Below is our effort to infer UNet Generator 5. We got 2 fps on Intel i5 8th Gen CPU which is pretty low speed but it does the job well.

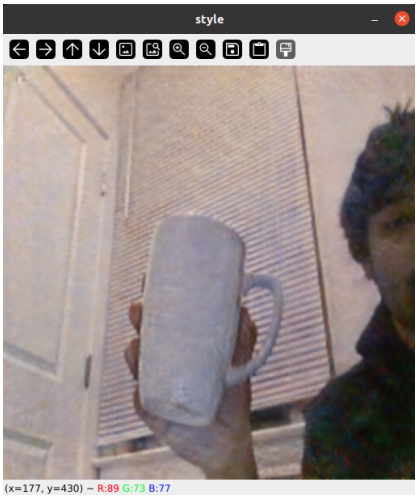


Fig. 5: Webcam Results

VI. Conclusion

Neural Style Transfer (NST) represents a powerful blend of art and technology, using neural networks to merge the content of one image with the artistic style of another. Our exploration of NST techniques, including VGG, ResNet, and CycleGAN, has shown how these methods produce visually stunning results, enhancing digital art. By evaluating perceptual quality, computational efficiency, and scalability, we identified the strengths and limitations of each approach, aiding in selecting the best method for specific needs.

NST aims to create aesthetically appealing and artistically coherent images. The methods we studied ensure high aesthetic quality by effectively merging intricate style elements with the original content. This opens new creative possibilities for artists and designers, allowing them to create unique and captivating visuals. Furthermore, CycleGAN's image-to-image translation capabilities enable unpaired transformations, broadening the versatility of NST techniques.

Future research should focus on optimizing NST methods to improve visual quality and computational efficiency. As NST techniques advance, their applications can expand beyond digital art to areas like virtual reality, augmented reality, and interactive media, creating immersive experiences.

Your report title and the list of team members will be published on the class website. Would you also like your pdf report to be published? **YES**. Our github Repo: https://github.com/jayaramakarthikeya/paint_anything

VII. Ethical considerations and broader social impact

Success in our project could democratize access to artistic tools and foster creativity among individuals regardless of artistic expertise. However, we must be mindful of potential misuse of neural style transfer, such as generating deceptive or misleading content. We will prioritize ethical considerations and advocate for responsible use of our technology to mitigate potential negative impacts on society.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[4] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, 2017.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015.

[6] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," 2020.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[9] S. Desai, "Neural artistic style transfer: A comprehensive look," 2017.

[10] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei, "Characterizing and improving stability in neural style transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4067–4076, 2017.

[11] Y. Cui, Y. Luan, and J. Guo, "Improved cyclegan for natural scenery images style transfer," in *2022 2nd International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*, pp. 16–22, IEEE, 2022.

Project Check-in

Graded

Group

Karthikeya Jayarama
Varun Velankar
View or edit group

Total Points

1 / 1 pts

Question 1

Sufficiency?

1 / 1 pt

✔ **+ 1 pt** See notes.

Question assigned to the following page: 1

Image to Image Translation using CycleGANs

Team: Karthikeya Jayarama, Varun Velankar Project Mentor TA: Alok Patwa

1) Introduction

Set up the problem: We are more interested in the Style transfer part of Image to Image translation problems. We are using Monet paintings and photos as a dataset to train the CycleGAN model architecture. Since the CycleGAN model learns a bidirectional mapping between the two domains, it is important to evaluate the cycle consistency. This means that if an image from domain A is translated to domain B, and then the resulting image is translated back to domain A, the final output should be as close to the original input image as possible. This can be measured using metrics like cycle consistency loss or mean squared error between the original and cycle-reconstructed images.

Motivation: Making progress on image-to-image translation using CycleGANs for style transfer could have significant impacts. It would enable new creative possibilities across digital art, photography, multimedia, and more by allowing artists and creators to explore novel visual styles and effects. Additionally, it could benefit fields like computer vision through data augmentation and compression applications. However, ethical concerns must be carefully addressed. This technology raises issues around potential misuse for nefarious purposes like generating deep fakes or violating intellectual property rights if copyrighted artistic works are replicated without permission.

2) How We Have Addressed Feedback From the Proposal Evaluations

From the previous email we received from the TA we have incorporated the following suggestions:

- Narrow down the dataset: From several available datasets and use cases of Image to Image Translation we are using only the Photo_to_Monet dataset for training and testing CycleGANs.
- Try out different architectures: We are trying out state of the art cycleGANs for our project. If time permits we will start implementing other architectures. As we are not using simple CNN for style transfer, we believe the GAN architecture would be sufficient. (Open to suggestions please :))
- Human evaluation loop: We will be building a webcam application by taking input from the webcam and output style transferred image on the screen.

3) Prior Work We are Closely Building From

Unpaired image-to-image translation, where images from two different domains are mapped without paired training examples, has important applications in computer vision and multimedia.

Question assigned to the following page: 1

Here we highlight two influential prior works related to our project on image translation using CycleGANs:

1. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" by Zhu et al. (2017) (https://arxiv.org/abs/1703.10593). This seminal paper introduced the CycleGAN framework which uses cycle consistency to learn mappings between two domains from unpaired training data. Code: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

2. "Contrastive Unpaired Translation" by Park et al. (2020) (https://arxiv.org/abs/2007.15135). This work extends CycleGANs by using a contrastive learning strategy to improve the translation quality and disentangle content and style more effectively. Code: https://github.com/clovaai/cont-cycleGAN

4) What We are Contributing

Our primary contribution lies in the practical implementation and refinement of neural style transfer techniques, with a focus on optimizing visual quality and computational efficiency. While prior work has established the theoretical framework for neural style transfer, our project aims to explore practical applications and potential enhancements to existing methods.

5) Detailed Description of Each Proposed Contribution, Progress Towards It, and Any Difficulties Encountered So Far

5.1 Methods

Our goal is to replicate the state of the art work of [Zhu et al., 2017], which introduced the CycleGAN framework for unpaired image-to-image translation. The key challenge in this task is to learn the mapping between two different image domains, X and Y, without paired training examples. Previous approaches, such as Pix2Pix [Isola et al., 2017], required paired data, which can be difficult to obtain in many scenarios.

The CycleGAN framework addresses this limitation by using a cycle-consistent adversarial network to learn the mapping functions G: $X \rightarrow Y$ and F: $Y \rightarrow X$ in an unsupervised manner. The core idea is to introduce two adversarial discriminators, DX and DY, which ensure that the translated images G(x) and F(y) are indistinguishable from real images in their respective target domains, Y and X.

Additionally, the framework enforces cycle consistency by requiring that $F(G(x)) \approx x$ and $G(F(y)) \approx y$, which helps preserve the content of the input images during translation. This is achieved by introducing a cycle consistency loss, which minimizes the reconstruction error between the input and the cycle-reconstructed images.

Formally, the CycleGAN objective function can be expressed as:

Question assigned to the following page: 1

L(G, F, DX, DY) = LGAN(G, DY, X, Y) + LGAN(F, DX, Y, X) + λ * Lcyc(G, F)

where LGAN is the adversarial loss based on the discriminators DX and DY, and Lcyc is the cycle consistency loss, weighted by a hyperparameter λ.

The adversarial losses are formulated similar to the traditional GAN framework [Goodfellow et al., 2014], using a least-squares objective [Mao et al., 2017] for improved stability during training.

After implementation of the CycleGAN we would try to implement a real time style transfer using a webcam. So this helps us complete the objective of the useful cases of style transfer and people will be excited when they use it.

5.2 Experiments and Results

Our replication effort will involve in implementing the CycleGAN architecture as described in the paper [Zhu et al., 2017] using Pytorch. We are building a machine learning pipeline involving creating Pytorch dataset to convert images to tensors.

`Result From Image preprocessing pipeline (Right side is monet painting for Gen1 and Photo image for Gen2). We are also using data augmentation in order to reduce covariance shift.`

We have also implemented the CycleGAN architecture in Pytorch and also the adversarial and identity loss. Currently, we are implementing the training loop and by the end of the Wednesday we will have the full setup.

Question assigned to the following page: 1

We are evaluating our implementation using generator loss and photometric regeneration (to check how well the model generates the image translation for the given photo or vice versa).

6) Risk Mitigation Plan

In this project, we deploy CycleGANs for image-to-image translation, incorporating strategic risk mitigation to navigate anticipated challenges effectively. To ensure the project remains feasible within the allocated time, our approach prioritizes the construction of a minimum viable project (MVP). This MVP focuses on the simple, yet functional capabilities of the CycleGAN, handling basic tasks of image translation with a streamlined selection of image categories or simpler transformations. This foundational step ensures an operational baseline early in development, allowing for prompt detection and rectification of fundamental issues.

Initiating the project with a simplified dataset—comprising less complex images—enables quick, actionable insights and provides the flexibility to adjust methodologies early in the project lifecycle. This step is crucial for mitigating the risk of significant, time-consuming pivots later in the project.

Should the initial methodologies prove suboptimal, all outcomes, both successful and unsuccessful, will be documented. This documentation will articulate the approaches attempted, delineate the failures, and analyze the underlying reasons, turning every outcome into a learning opportunity and a detailed segment of the final project report.

Additionally, computational efficiency is a pivotal concern given the resource-intensive nature of training CycleGANs. Should computational limitations arise, the project plan includes strategies such as reducing the complexity of tasks, utilizing pre-trained models, or employing smaller, less demanding datasets. Furthermore, experimenting with a synthetic "toy" dataset may be considered to address specific challenges in a controlled environment, thus isolating problems without the confounding variables present in real-world data.

The project also entails a thorough analysis of specific instances where the CycleGAN model excels or underperforms. Identifying patterns in these instances will guide targeted improvements and might uncover niche applications where the model offers superior performance. Through these strategies, the project not only tackles technical hurdles but also ensures an adaptable, insightful progression toward achieving its objectives, thereby maximizing the learning gleaned from this academic endeavor.

Question assigned to the following page: 1

Full Work Plan:

References:

1. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" by Zhu et al. (2017) (https://arxiv.org/abs/1703.10593). This seminal paper introduced the CycleGAN framework which uses cycle consistency to learn mappings between two domains from unpaired training data. Code: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

2. "Contrastive Unpaired Translation" by Park et al. (2020) (https://arxiv.org/abs/2007.15135). This work extends CycleGANs by using a contrastive learning strategy to improve the translation quality and disentangle content and style more effectively. Code: https://github.com/clovaai/cont-cycleGAN

3. "Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs" by L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille (2018) (https://ieeexplore.ieee.org/document/7904630). This paper presents DeepLab, a method for semantic image segmentation that leverages deep convolutional networks, atrous convolution, and fully connected Conditional Random Fields (CRFs).

4. "ImageNet: A Large-Scale Hierarchical Image Database" by J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009) (https://ieeexplore.ieee.org/document/5206848). Presented

Question assigned to the following page: 1

at CVPR09, this work introduced ImageNet, a substantial image database organized according to the WordNet hierarchy that significantly impacts machine learning and computer vision.

5. "Neural Artistic Style Transfer: A Comprehensive Look" by S. Desai. This review explores the developments and methodologies of neural style transfer, analyzing various approaches and their artistic implications.

6. "A Learned Representation of Artistic Style" by V. Dumoulin, J. Shlens, and M. Kudlur (2017) (https://arxiv.org/abs/1610.07629). Presented at ICLR 2017, this research introduces techniques for learning representations of artistic styles in neural networks.

7. "A Neural Algorithm of Artistic Style" by L. A. Gatys, A. S. Ecker, and M. Bethge (2015) (https://arxiv.org/abs/1508.06576). This influential paper proposes a method for applying the style of one image to the content of another using convolutional neural networks, laying foundational work for neural style transfer.

8. "Characterizing and Improving Stability in Neural Style Transfer" by A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei (2017) (https://openaccess.thecvf.com/content_ICCV_2017/papers/Gupta_Characterizing_and_Improving_ICCV_2017_paper.pdf). Presented at ICCV 2017, this paper addresses the stability issues in neural style transfer, proposing methods to enhance the robustness of style transfer applications.

Work Plan over the next ~5 weeks:

TASK (S)	Week 1	Week 2	Week3	Week 4	Week 5
	March	April	April	April	April
Collecting dataset					
Reading papers					
Developing dataset preprocessing pipeline					
Designing the model					
Evaluation					
Debugging					

Paint Anything - Neural Style Transfer

Karthikeya Jayarama
Varun Velankar

Introduction

1. Introduction to Neural Style Transfer (NST)
    - Definition: Technique to blend content and style of different images using neural networks
    - Origin: First introduced by Gatys et al. in 2015
    - Applications: Primarily used in digital art and computer vision for creating stylized images
    - Equation: NST objective function

$$L_{\text{total}}(C, S, G) = \alpha L_{\text{content}}(C, G) + \beta L_{\text{style}}(S, G)$$

2. Importance and Goals of NST
    - Purpose: Produce visually appealing images by merging content and style
    - Learning Mechanism: Utilizing CNNs to extract and recombine features
    - Equation: Content loss

$$L_{\text{content}}(C, G) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

    - Equation: Style loss using Gram matrices

$$L_{\text{style}}(S, G) = \sum_{l=1}^{L} w_l \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

Background

Neural Style Transfer (NST) is a technique in computer vision that blends the content of one image with the artistic style of another. Originally introduced by Gatys et al. in 2015, NST leverages the feature extraction capabilities of convolutional neural networks (CNNs) to achieve this combination. The process involves two main components: content loss and style loss.

Vanilla Neural Style Transfer:

- **Content Loss:** Measures the dissimilarity between the content of the content image (C) and the generated image (G). It is calculated as the squared error loss between their feature representations extracted from a pre-trained CNN.

$$L_{\text{content}}(C, G) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2$$

- **Style Loss:** Captures the style of an image using Gram matrices, which are correlations between different filter responses in the CNN. The style loss is the mean squared error between the Gram matrices of the style image (S) and the generated image (G).

$$L_{\text{style}}(S, G) = \sum_{l=1}^{L} w_l \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

- **Total Loss Function:** Balances content and style losses to produce the final stylized image.

$$L_{\text{total}}(C, S, G) = \alpha L_{\text{content}}(C, G) + \beta L_{\text{style}}(S, G)$$

Evolution of NST Approaches:

- **VGG-Based Methods:** Utilize VGG networks for effective feature extraction. VGG's deep architecture captures fine details, making it suitable for high-level texture synthesis.
- **CycleGAN:** Uses generative adversarial networks (GANs) for style transfer without paired training data. CycleGAN introduces a cycle consistency loss to ensure the generated image remains faithful to the content while adopting the desired style.

$$L_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

- **ResNet:** Leverages a deep residual learning framework with skip connections to address the vanishing gradient problem, enabling the training of very deep networks that can capture intricate style features.

Impact and Applications:

NST has significantly impacted digital art and computer vision by enabling the creation of visually appealing images that combine different content and styles. Artists and designers can use NST to produce unique and captivating visuals, expanding creative possibilities. Additionally, CycleGAN's image-to-image translation capabilities allow for unpaired transformations, broadening the scope of NST applications in various domains.

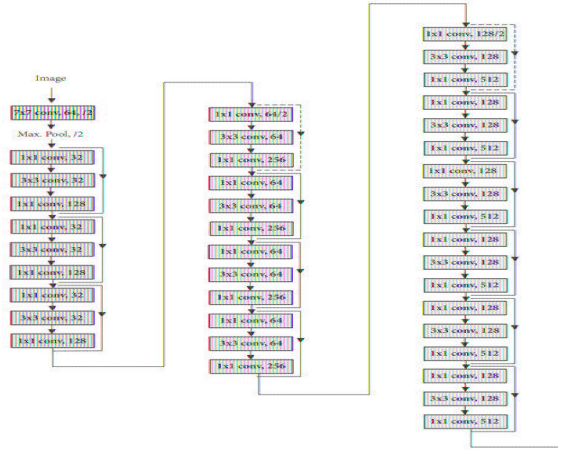
Methodology

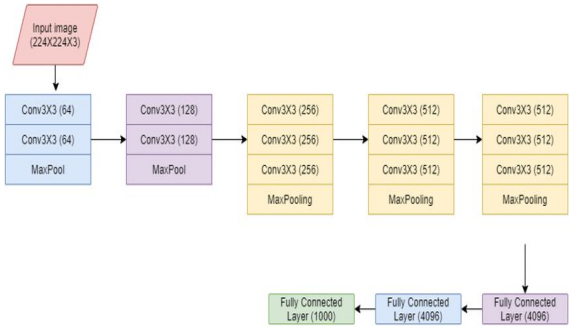
1. Key Approaches in NST
    - CycleGAN: Leverages GANs for style transfer without paired data
    - VGG-Based Methods: Uses VGG networks for effective feature extraction
    - ResNet: Deep residual learning framework to handle intricate style features
    - Diagram: Architecture of CycleGAN and VGG
2. Comparative Analysis
    - Metrics: Perceptual quality, computational efficiency, scalability
    - Evaluation: Strengths and limitations of each approach for digital art and image processing
    - Equation: Overall loss function for CycleGAN

$$L(G, F, D_X, D_Y) = L_{\text{GAN}}(G, D_Y, X, Y) + L_{\text{GAN}}(F, D_X, Y, X) + \lambda L_{\text{cyc}}(G, F)$$

    - Equation: Adversarial loss

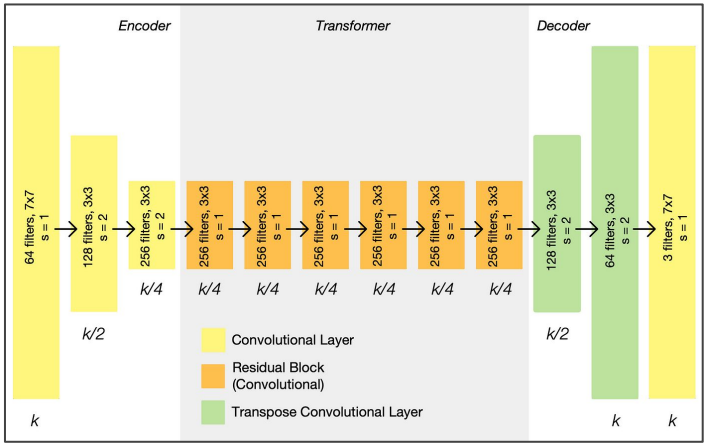
$$L_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x)))]$$





CycleGAN

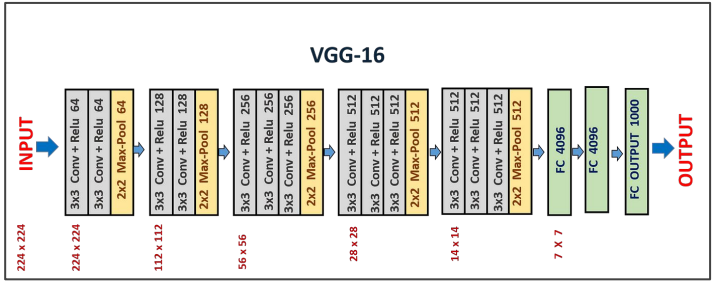
1. Overview of CycleGAN
    - Functionality: Enables unpaired image-to-image translation
    - Architecture: Includes two GANs and a cycle consistency loss
    - Diagram: CycleGAN architecture with generators and discriminators
    - Equation: Cycle consistency loss



$$L_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1]$$

2. Advantages and Applications
    - Unsupervised Learning: Effective without paired training data
    - Versatility: Applied in diverse domains including video generation and music style transfer
    - Diagram: Example of CycleGAN results in different domains

VGG



1. Architecture of VGG Networks
    - Design: Simple, deep networks using small convolution filters
    - Feature Extraction: Effective in capturing image textures and patterns
    - Diagram: VGG network architecture
    - Equation: Calculation of output size for convolutional layers

$$\text{Output size} = \left(\frac{\text{Input size} - \text{Filter size} + 2 \times \text{Padding}}{\text{Stride}} \right) + 1$$

2. Applications and Performance
    - Tasks: Particularly good for high-level texture synthesis and style transfer
    - Hypotheses: Expected to perform well in style-centric applications
    - Diagram: Comparison of VGG-based NST results with other methods

ResNet

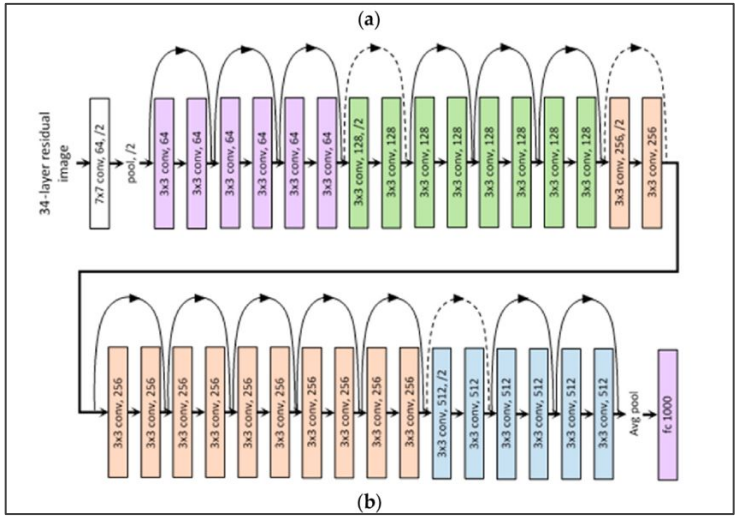
1. ResNet Architecture
    - Innovation: Skip connections to address vanishing gradient problem
    - Depth: Enables training of very deep networks for complex tasks
    - Diagram: ResNet block diagram
    - Equation: Residual block function

$$y = F(x, \{W_i\}) + x$$

    - Equation: Activation function (ReLU)

$$f(x) = \max(0, x)$$

2. Applications and Performance
    - Tasks: Effective in object recognition and complex classification tasks
    - Hypotheses: Likely to outperform in tasks requiring deep feature hierarchies
    - Diagram: ResNet performance in image classification tasks



Results

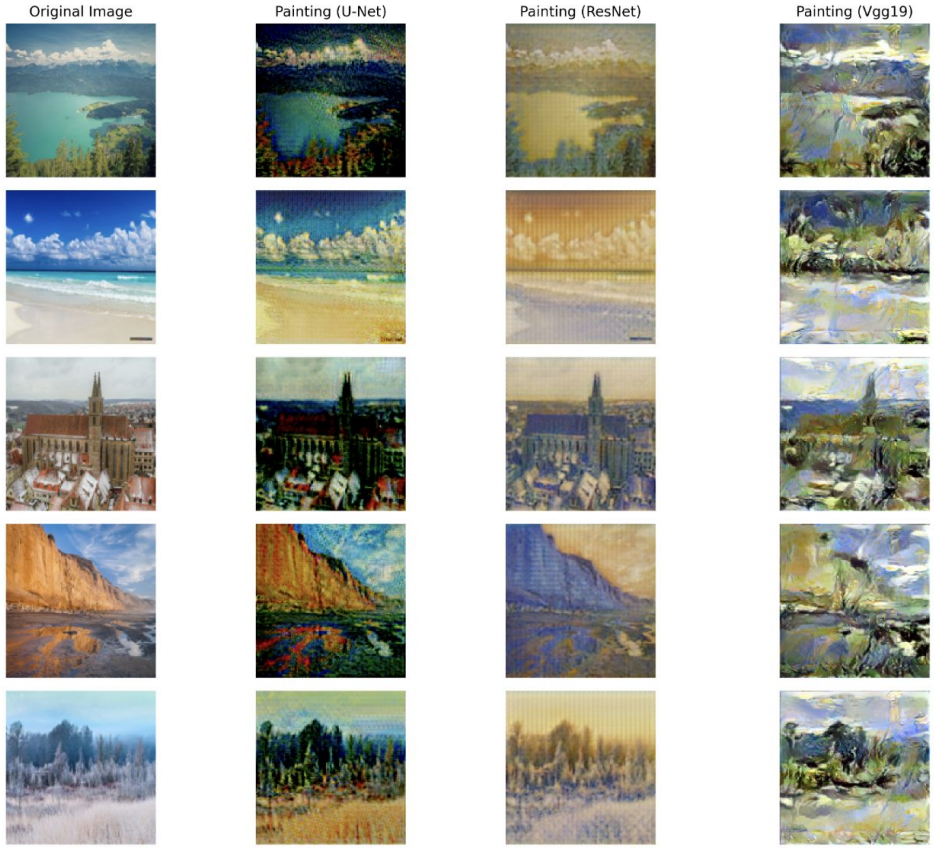


Fig. 4: Experimental Results

Model Type	GPU Used(Training)	No. of Parameters	Training Time	# of Epochs	Batch Size	Crop Size
VGGNet-19	Tesla P4	555328	2 mins/image	10	1	128 x 128
ResNet CycleGAN	Tesla P100	7.84 M	2.1 hrs	60	32	128 x 128
UNet CycleGAN	Tesla P100	11.38 M	2.3 hrs	42	32	128 x 128

TABLE I: Table containing model performance metrics following training and validation runs, along with select model parameters

Output Parameters

Actual image outputs:
We think that the U-Net paintings have a better aesthetic appeal and hence we have concluded that U-Net is a much better model than others.

References

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, 2017.

[2] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," 2020.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[5] S. Desai, "Neural artistic style transfer: A comprehensive look," 2017.

[6] A. Gupta, J. Johnson, A. Alahi, and L. Fei-Fei, "Characterizing and improving stability in neural style transfer," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4067–4076, 2017.

[7] Y. Cui, Y. Luan, and J. Guo, "Improved cyclegan for natural scenery images style transfer," in *2022 2nd International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*, pp. 16–22, IEEE, 2022.