IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11543

Image Style Transfer Application With Text Condition Using CNN And CLIP

Rohith D'souza M[1], Prateek Sharma S[2], Pranavi Shree V[3], Lilly Florence M[4]

U.G. Student, Department of CSE, Adhiyamaan College of Engineering, Tamil Nadu, India[1,2,3]

Assistant Head of Department, Department of CSE, Adhiyamaan College of Engineering, Tamil Nadu, India[4]

Abstract: "What if we can bring our imagination into reality?" This is something this paper is trying to address with the help of image style transfer and CLIP.Image Style transfer is a technique used to take two images- a content image and a style reference image and blend them together so the output image looks like the content image, but "painted" in the style of the style reference image. As stated above, existing style transfer methods require reference style images to transfer texture information to content images. But what if one does not have a reference image but is still interested in transferring styles by just imagining them? To overcome the above issue, here we are using text-to-image embedding model of CLIP to perform text-driven image style transfer without the use of a style image as a reference. We also use a CNN Encoder-Decoder Model to capture the visual features of the content image and simultaneously stylize the image to obtain a realistic texture representation. Finally, the above model is deployed as a web application from which users can performstyle transfer by uploading an image and specifying the desired styling with thehelp of a text description.

Keywords: Style Transfer, CNN, CLIP, VGG, U-Net.

1. INTRODUCTION

Consider you have a portrait image of yourself and you wonder how would it look like if Leonardo da Vinci had painted it like the Mona Lisa painting? or how does a scenic landscape photo- graph taken in summer would like if it was winter? We as humans are capable of accurately describing what we think or imagine through text. It is also possible to infer that the painting of Mona Lisa has a certain style to it (color, brushstroke etc.) and if this style was somehow transferred to our given image we would obtain the required image of our imagination.

In this paper, we present an application that can learn to do the same: capturing special characteristics of a style through a text condition and figuring out how these characteristics could be transferred into our given input image. This problem can be more broadly described as Image Style Transfer or Neural Style Transfer *(NST)*. Image Style Transfer is a technique for blending two pictures-a content image and a style reference image-so that the output image appears like the content image but is "painted" in the style of a reference style image[1].

Inspired by Convolutional Neural Networks *(CNNs)*[2] first proposed a technique where they make use of a pre- trained VGG network to transfer the style of an artwork to a content image. Here they make use of a style loss function which matches the Gram matrices of the content and style features, which has become the standard for NST. There has been further progress in the field with different type of style transfer[3] such as semantic style transfer[4][5][6], instance style transfer[7], portrait style transfer[8][9] etc.

All the above style transfer have one crucial limitation which is the need for having two images - A content image and a style reference image. This becomes an issue when a user is unableto find a reference image to the liking of his imagination but is very capable of expressing it via text. Hence this is the issue this project is trying to address with the help of text driven image style transfer.

There has been several different approaches towards this problem, but the use of pre-trained text-image embedding model to deliver the style information of a text condition to the content image has found more success especially with recent release of Contrastive Language-Image Pretraining (CLIP) [10].
As shown in [11] this paper makes useof CNN to both simultaneously represent and stylize the content image which is then fed to CLIP which guides the CNN to stylize the image towards the style represented in the given text condition.

Finally, we propose a novel way to implement the trained model as a web application. It is imperative to implement deep learning model as a practical application such that users are able to use it and obtain the fruits gained from the modern research and to have a positive impact in society. There are various practical applications of NST such as photo editing, digital art, entertainment applications, home decor etc. [3].
Even with the recent hype behind Non-Fungible Tokens (NFTs) image styletransfer has a wide-ranging application as artists can use it to augment theircreative process for inspiration or produce different versions of a particular image in an instance. Hence producing apractical application performing Image Style Transfer by overcoming the limitations of existing applications which makes use of a reference image becomesan important problem with added socialand monetary significance.

2. RELATED WORK

2.1 Style Transfer:

There has been significant research in style transfer, which evidently started with Gatys et al. [12] who first proposed a pixel-based optimization technique andstarted the field of Style transfer. Later on Gatys et al [2] again showed thatCNNs can be used effectively along with Gram matrix loss functions to perform Neural Style transfer.
Later which, Ulyanov et al. [13] and Johnson et al. [14] introduced a per- style-per-model based technique where the results were very similar to [2] al-though Li and Wand [15] used a similar technique but the results were less appealing compared to [14] and [13]. Li and Wand [15] uses MRF loss whichis only more effective when the content and style are similar in shape and perspective, compared with Gram Loss. Whereas Johnso et al. [14] uses content loss based on perceptual similarity, which is now widely adopted. The primary limitation with these model is that each model can perform style transfer for one style. These above mentioned methods can be classified as Instance Normalization technique.

There has also been a different class of approach which can be referred toas multi-style-per-model which is done in Dumoulin et al. [16] and Chen etal. [17] but the model size generally becomes larger with the increased number of learned styles. Li et al. [18] and Zhang and Dana [19] have fixed model size but there seems to be some interferences among different styles. Li etal [18] makes use of Mean-subtraction Gram loss which is the reason why the models have fixed size as it works by subtracting the mean of feature representations before computing Gram Loss.Zhang and Dana [19] instead make use of Multi-scale Gram loss which computes Gram loss over multi-scale feature representation. These type of models can also be classified as Conditional Instance Normalization.
Finally to overcome the above limitations of the two, Arbitrary-Style-Per- Model [3] is used where the primary idea is one-model for all, although the results can be a tad bit less impressive compared to the gold standard, but it is a common trade-off in research between speed, flexibility and quality. Some of the models implementing ASPM are Chen and Schmidt [20] but it does not combine enough style elements. Huang and Belongie [21] and Ghiasi et al. [22] are also a type of ASPM but are not effective in producing complex style patterns. Whereas Li et al. [23] is not good at producing sharp details and finestrokes. These can also be classified as Adaptive Instance Normalization.

Some other type of style transfer exists which are designed for various otherspecific instances such as Ruder et al[24] which is designed for video style transfer, Capable of maintaining temporal consistency among stylised video frames or Chen et al[17] which uses disparity loss and is specifically designed to perform stereoscopic style transfer and is capable of consistent strokes for different views.

Whereas in more recent research in style transfer, it can be seen that Liuet al. [25] proposed adaptive attention normalization which is an improvementover SANet by Park et al. [26] which is an attention-based style transfer technique. Kotovenko et al. [27] introduced a different way to perform style transfer where instead of optimizing pixels, he proposed to optimize parameters of modelled quadratic Bezier curvers.Hong et al. [28] also introduced a new approach where the model transfers the domain aware information along withstyle.

Despite these methods showing impressive results all of them have one issue in common which is the need for a reference style image to perform image style transfer.

2.2 Text Guided Synthesis:

The key break-through required for text-guided image style transfer is given by CLIP [10] which is a powerful modelfor joint image-text representation. It is trained on more than 400 million image-text pairs, both image-text pairs are brought into a latent vector space bycontrastive learning. The representation learned by clip can be used

for a varietyof task including image manipulation by[29][30] and image synthesis [30].
Specifically using CLIP, StyleCLIP[31] made use of the ever-famous generative adversarial network StyleGAN[32] to perform attribute manipulationof portrait images as the domain of Style-GAN [32] specifically lies in generating portrait images. As mentioned, it can only manipulate images based on the trained domain, it becomes a significant limitation of this model as itis not general enough. As an improvement, StyleGAN-NADA[9] proposed a novel method to manipulate attributes based on text-condition using CLIP. But it still suffers with the same limitation that it depends on the domain the pre- trained generative model is trained in.
Very recently, Kwon et al. introduced CLIPStyler [11] which overcomesthe above limitation as instead of usinga pre-trained generative adversarial network, they instead rely on the CNNs to both obtain the higher-level features of an image and also manipulate the imagesimultaneously. Because of this, style transfer can happen regardless of the domain of input images. Hence making themodel more general than the rest.
But the limitation with CLIP- Styler[11] is that although there is a trained model capable of performing style transfer it is not implemented practically as a web application. by which users might be able to perform image style transfer and bring their imagination into reality, this is one thing this paper is trying to fix by introducing a web application where the model is deployed.

3. METHOD

3.1 A General Overflow of Our Application:

The basic workflow of application first starts with by getting input for text condition and source image from the user.

Next, we assign certain fixed a training value for our model such as crop size, loss values, etc. Also, we create and assign our CNN model comprising of U-Net architecture with five convolutional layers, after which we assign adam optimizer to our CNN. We then assign two texts to our CLIP model - style text,which represents the input text obtained from user and source text (eg: a photo) which are then fed to CLIP to obtain contrastive value score.

Next, the actual training of model be-gins where first we invoke our CNN U- Net Architecture with VGG model using sigmoid function. After which we perform random cropping on the given image along with perspective augmentation then we calculate the image featuresusing clip. Then next we start calculating all the loss functions for our model which includes

1. Directional clip loss [9]
2. Patch clip loss function [11]
3. Content loss function
4. Total Variational Regularizational loss

All of this is combined to obtain total loss function. Finally, for each of these iterations, we update our loss function and pass it to a CNN and this is re-peated for N-epochs. The overall architecture of the application is given in figure 1.

3.2 CNN Model:

A Convolutional Neural Network model is helpful in capturing the hierarchical visual features of the content image [2] An encoder-decoder model of CNN is used to both capture features and simultaneously stylize the image [11]. The Convolutional Neural Network is a deeplearning model built using pytorch froma pre-trained vgg-19 model and the CNN model follows U-Net Architecture with 5 convolutional layers, Sigmoid Activation function is used for our model along with Adam optimizer.

3.3 CLIP Loss Functions:

This module consists of CLIP and the rest of the functions required for image style transfer. CLIP (Contrastive Language-Image Pre-Training) is a neural network trained on a variety of (image, text) pairs(400 Million), it was released by OpenAI [10].

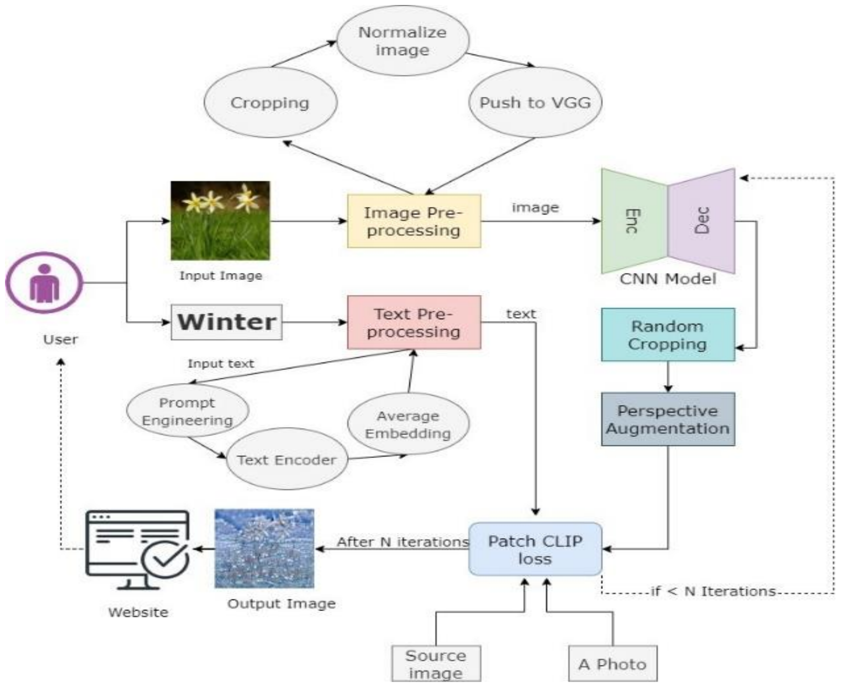

Figure 1: Overall architecture of application

First the output image from our CNN model is patch-wise randomly cropped and perspective augmentation is applied to each of the image to increase diversity. This image is sent as input to our Patchwise CLIP loss function. The totalloss function consists of the following-

3.3.1 Directional CLIP loss to modulate the whole part of the content image (e.g. color tone, global semantics)

$$\Delta T = ET(tsty) - ET(tsrc)$$
$$\Delta I = EI(f(Ic)) - EI(Ic)$$
$$L_{dir} = \frac{\Delta I.\Delta T}{|\Delta I||\Delta T|}$$

where E_I and E_T are CLIP's im- age and text encoders, respectively,and t_{sty}, t_{src} are the semantic text of the style target and the input con-tent. When we utilise natural photos for content, we simply set t_{src} to"Photo."

3.3.2 PatchCLIP loss for local texturestylization. As shown in CLIPStyler[11] solely using directional loss has decreased output quality, hence thisloss function is added to improve the stylization output produced. In this step random cropping of a fixedsize and perspective augmentation is done.

$$\Delta T = ET(tsty) - ET(tsrc)$$
$$\Delta I = EI(aug\left(\hat{I}_{cs}^{i}\right) - EI(Ic)$$
$$l_{patch}^{i} = 1 - \frac{\Delta I.\Delta T}{|\Delta I||\Delta T|}$$
$$L_{patch} = \frac{1}{n}\sum_{i}^{N} R\left(l_{patch}^{i}, \tau\right)$$

Where $\hat{I}$ **is the i – th cropped patch** from the output image, aug is random perspective augmentation, and $R(l_{patch}^{i}, \tau)$is the threshold function.

3.3.3 Content loss As shown in [11] to maintain the content information of input image, we include the content loss L_c by calculating the mean- square error between the features ofthe content and output images extracted from the pre-trained VGG- 19 networks similar to the existing work by Gatys et al[2].

3.3.4 Total Variation Regularization loss to alleviate the side artifacts from irregular pixels L_{tv} as shown in [11].

Finally, now our total loss function be- comes:

$$L_{total} = \lambda_d L_{dir} + \lambda_p L_{patch} + \lambda_c L_c + \lambda_{tv} L_{tv}$$

For each iteration our loss function is updated and passed to CNN to guide thestylization in the right direction.

3.4 Web Application:

The web application is the part which can be used by the user to perform Image style transfer. It can be split into two parts.

3.4.1 Front End

This effort shows the graphics to the end user so that one person may have an easy-to-use experience with the platform. Front-end development is becoming increasingly crucial since a solid userinterface makes it easier to connect with the online platform. Front-end development might be more difficult at times since sophisticated design takes a long time.

Here we use HTML, CSS and JavaScript to build the front end of our application. The web page can be divided into two image elements, where the first element is used to obtain input from the user and the second element displays the output to the user. There's a text element where the user enters the text condition of his desired styling, after which there is a button element named "transfer" upon clicking which, the image is sent to our model and style transfer takes place.

3.4.2 Back End

Since in this application it is required to deploy our trained deep learning model locally, there is a need to use back-end inour web application. This part hosts the trained deep learning model of our web application and is built using the pythonflask framework. It has been seen that it is much easier to host a deep learning via flask than any other framework and hence, it was chosen.

Once we build and deploy our model locally with the flask framework there are other options to deploy our deep learning model such as one of cloud plat-forms eg: AWS,Azure,GCP but due to the limited budget constraints and no free tier offer for the usage of GPU in cloud platforms, we stick with deployingour model locally. It obtains the input image and text condition from user as a POST request, image is saved and text isstored in a variable, both image and text is passed as input to our model, Image is preprocessed and then fed to the modelfor the given inputs and the output image is stored which is then retrieved andshown in the front-end of our application.

4. IMPLEMENTATION

This section describes the implementation details of the project.

4.1 Network Architecture:

We adopt the pre-trained VGG-19 model as the feature extractor. It follows the lightweight U-net [33] architecture which contains three downsample and three upsample layers, the channel sizes for each downsample layer are 16,32 and 64 respectfully. Similar to [2]we use features of layer *conv4_2* and *conv5_2* for content loss. We use sigmoid function for the last layer, so that the pixel value range is between [0,1]. To train the network, we use Adam optimizer [34] with a learning rate of 5×10^{-4}. The total training iterations can be modified and is set to 50 by default (for fast output).

4.2 Training Details:

Considering the resource capacity we use 512 x 512 resolution for all content images. For training we use λ_d, λ_p, λ_c *and* λ_{tv} *as* 5×10^2, 9×10^3, 150 *and* 2×10^{-3}*respectively* and threshold rejection we set $\tau = 0.7$.

For patch cropping, we use patch size 128 x 128 as it shows best perceptualquality. The total number of cropped patches is set as $n = 64$. Perspective augmentation is performed using function given in Pytorch library.

With regards to text condition, we make use of prompt engineering technique as specified by [35]. Specifically, We generate a number of text with the same meaning and feed them into the text encoder. The averaged embedding is then used instead of the original single text conditions. Finally we apply contrast enhancement techniques in-order to deliver more visually pleasable images.

5. RESULTS

We begin by showcasing the results obtained by our model for various different images and input text conditions, as shown in figure 2. In the result image, we can do style transfer for a wide variety of styles including artistic and various other texture changes. It is possible to adjust not only the texture style but also the specific requirements of each style. For example, we can alter the color and also the texture style usingtext conditions such as "gold sandpaper"and "winter season".

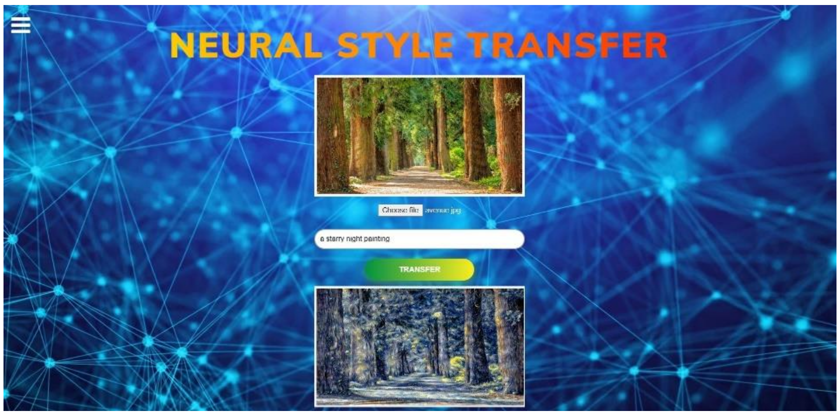

Figure 3: Web application for our model

The model is deployed as a web application and we see the results obtained in figure 3.

5.1 Comparison with text-guided manipulation models:

Figure 4 shows the comparison with various other text-guided manipulation models. We make use of StyleNADA [9] and StyleCLIP [31] these are state-of- the-art CLIP models. The models are trained in human faces, hence we perform a comparison using celebrity images.

As seen below the models only work well on the dataset domain they are trained in (Celeb-A dataset). With regards to StyleCLIP it only performs editing features of the given image suchas "bowl haircut" etc. The model is veryweak with regards to input text condition as well. As for StyleGAN-NADA, it can only work on pre-defined input text conditions, all the text conditions used were of the given text conditions the model was already pre-trained with, hence this model is also not capable of arbitrary style transfer with any text condition like our model.

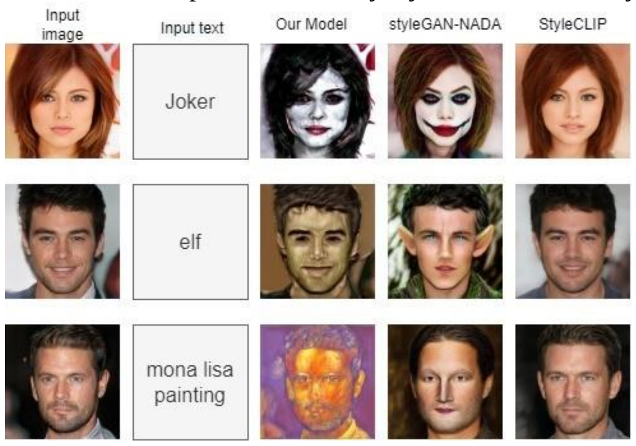

Figure 4. Comparison Of Models

Impact Factor 7.39 ⋇ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11543

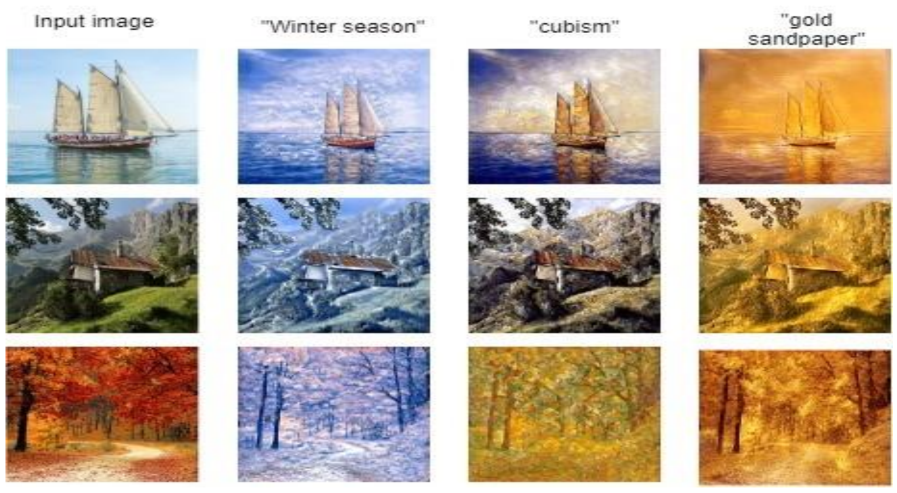

Figure 2: Example images generated by model for various input texts

5.2 Ablation Studies:

Ablation study is used to compare the necessity of each component in a model. Figure 5 shows the results of ablation study. When using all components, we can see that we get the best results. When removing L_{dir} we can see that some artifacts are produced and doesnot go well with the whole image overall. When we remove threshold rejection, we can see that the image is over-stylized in all patches and the result is abysmal. When we remove L_{patch} we can see that there is little effect to the texture of our image and only the color of the image is primarily impacted.

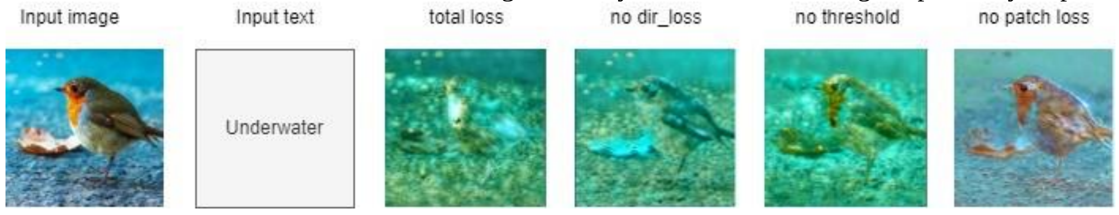

Figure 5: Ablation study comparison

6. CONCLUSION

We introduce a deep-learning approach that faithfully transfers style from a given text condition, without the use of a style reference image for a wide variety of image content. We make use of CLIPand various other loss functions to produce state-of-the-art image style transfer results with only text condition. The model is then deployed as a web application using flask framework, by which users can make use of state-of-the-art deep learning models with ease.

7. REFERENCES

[1] Neural style transfer | TensorFlowCore. en. url: https://www.tensorflow.org/tutorials/generative / style _ transfer (visited on 02/17/2022).

[2] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge.“Image Style Transfer Using Convolutional Neural Networks”. en. In: 2016 IEEE Confer- ence on Computer Vision and Pattern Recognition (CVPR).Las Vegas, NV, USA: IEEE, June 2016, pp. 2414–2423. ISBn: 978-1-4673-8851-1. doI: 10.1109/CVPR.2016.265. url: http://ieeexplore.ieee.org/ document/7780634/ (visited on12/22/2021).

[3] Yongcheng Jing, Yezhou Yang,Zunlei Feng, et al. “Neural StyleTransfer: A Review”. en. In:arXiv:1705.04058 [cs, eess, stat] (Oct. 2018). arXiv: 1705.04058.url: http : / / arxiv . org /abs / 1705 . 04058 (visited on12/22/2021).

[4] Alex J Champandard. “Semantic style transfer and turning two-bit doodles into fine artworks”. In: arXiv preprint arXiv:1603.01768 (2016).

[5] Chuan Li and Michael Wand. “Combining markov random fields and convolutional neural networks for image synthesis”. In: Proceedings of the IEEE conference on computer vision and pattern recog-nition. 2016, pp. 2479–2486.

[6] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. “The contex- tual loss for image transformation with non-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⋇ Vol. 11, Issue 5, May 2022

DOI: 10.17148/IJARCCE.2022.11543

aligned data". In: Pro- ceedings of the European conference on computer vision (ECCV).2018, pp. 768–783.

[7] Carlos Castillo, Soham De, Xin- tong Han, et al. "Son of zorn'slemma: Targeted style transfer us-ing instance-aware semantic seg mentation". In: 2017 IEEE In- ternational Conference on Acous- tics, Speech and Signal Processing(ICASSP). IEEE. 2017, pp. 1348–1352.

[8] Ahmed Selim, Mohamed Elgharib, and Linda Doyle. "Painting style transfer for head portraits using convolutional neural networks". In: ACM Transactions on Graph-ics (ToG) 35.4 (2016), pp. 1–18.

[9] Rinon Gal, Or Patashnik, Haggai Maron, et al. "StyleGAN-NADA: CLIP-Guided Domain Adapta- tion of Image Generators". en. In: arXiv:2108.00946 [cs] (Dec. 2021). arXiv: 2108.00946. url: http://arxiv.org/abs/2108.00946(visited on 12/22/2021).

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning Transferable Visual Models From Natural Language Supervision. 2021. arXiv: 2103 . 00020[cs.CV].

[11] Gihyun Kwon and Jong Chul Ye. "CLIPstyler: Image Style Trans- fer with a Single Text Condition".en. In: arXiv:2112.00374 [cs, eess](Dec. 2021). arXiv: 2112.00374. url: http : / / arxiv . org /abs / 2112 . 00374 (visited on12/22/2021).

[12] Leon A. Gatys, Alexander S.Ecker, and Matthias Bethge."A Neural Algorithm of Artistic Style". In: arXiv:1508.06576 [cs, q-bio] (Sept. 2015). arXiv: 1508.06576. url: http://arxiv.

[13] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Improvedtexture networks: Maximizing quality and diversity in feed- forward stylization and texture synthesis". In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, pp. 6924–6932.

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: Europeanconference on computer vision. Springer. 2016, pp. 694–711.

[15] Chuan Li and Michael Wand. "Precomputed real-time texture synthesis with markovian genera- tive adversarial networks". In: Eu-ropean conference on computer vision. Springer. 2016, pp. 702–716.

[16] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style". In: arXiv preprint arXiv:1610.07629 (2016).

[17] Dongdong Chen, Lu Yuan, Jing Liao, et al. "Stereoscopic neural style transfer". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018, pp. 6654–6663.

[18] Yijun Li, Chen Fang, Jimei Yang,et al. "Diversified texture synthesis with feed-forward networks". In: Proceedings of the IEEE Conference on Computer Vision

[19] Hang Zhang and Kristin Dana."Multi-style generative networkfor real-time transfer". In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018, pp. 0–0.

[20] Tian Qi Chen and Mark Schmidt. "Fast patch-based style transfer of arbitrary style". In: arXiv preprint arXiv:1612.04337 (2016).

[21] Xun Huang and Serge Belongie. "Arbitrary style transfer in real- time with adaptive instance nor- malization". In: Proceedings of theIEEE international conference oncomputer vision. 2017, pp. 1501–1510.

[22] Golnaz Ghiasi, Honglak Lee, Man- junath Kudlur, et al. "Exploring the structure of a real-time,arbitrary neural artistic stylization network". In: arXiv preprint arXiv:1705.06830 (2017).

[23] Yijun Li, Chen Fang, Jimei Yang,et al. "Universal style transfer via feature transforms". In: Advances in neural information processing systems 30 (2017).

[24] Manuel Ruder, Alexey Dosovit-skiy, and Thomas Brox. "Artisticstyle transfer for videos". In: German conference on pattern recognition. Springer. 2016, pp. 26–36.

[25] Songhua Liu, Tianwei Lin, Dongliang He, et al. "Adaattn: Revisit attention mechanism in arbitrary neural style transfer". In: Proceedings of the IEEE/CVFInternational Conference on Computer Vision. 2021, pp. 6649–6658.

[26] Dae Young Park and Kwang Hee Lee. "Arbitrary style transfer with style-attentional networks". In: proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, pp. 5880–5888.

[27] Dmytro Kotovenko, Matthias Wright, Arthur Heimbrecht, et al. "Rethinking style transfer: From pixels to parameterized brushstrokes". In: Proceedings of the IEEE/CVF Conference onComputer Vision and Pattern Recognition. 2021, pp. 12196–12205.

[28] Kibeom Hong, Seogkyu Jeon,Huan Yang, et al. "Domain-Aware Universal Style Transfer". In: Pro- ceedings of the IEEE/CVF Inter-national Conference on ComputerVision. 2021, pp. 14609–14617.

[29] David Bau, Alex Andonian, Audrey Cui, et al."Paint by word". In: arXiv preprint arXiv:2103.10951 (2021).

[30] Adverb. The Big Sleep Here's the notebook for generating images by using CLIP to guide BigGAN. en.Tweet. Jan. 2021. url: https://twitter.com/advadnoun/status / 1351038053033406468 (visited on 02/18/2022).

DOI: 10.17148/IJARCCE.2022.11543

[31] Or Patashnik, Zongze Wu, Eli Shechtman, et al. "Styleclip: Text- driven manipulation of styleganimagery". In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021, pp. 2085–2094.

[32] Tero Karras, Samuli Laine, and Timo Aila. "A style-based genera- tor architecture for generative ad- versarial networks". In: Proceed-ings of the IEEE/CVF conferenceon computer vision and patternrecognition. 2019, pp. 4401–4410.

[33] Wenju Xu, Chengjiang Long, Ruisheng Wang, et al. "DRB- GAN: A dynamic resblock generative adversarial network for artis-tic style transfer". In: Proceedings of the IEEE/CVF International Conference on Computer Vision.2021, pp. 6383–6392.

[34] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: arXiv preprint arXiv:1412.6980 (2014).

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, et al. "Learningtransferable visual models from natural language supervision". In:International Conference on Ma- chine Learning. PMLR. 2021,pp. 8748–8763.