UNIVERSITY OF ZAGREB
**FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING**

**SEMINAR**

# Neural Style Transfer

*Josip Šarić*
Mentor: *prof. dr. sc. Sven Ločarić*

Zagreb, October 2020.

# CONTENTS

# 1. Introduction and related work

Image sharing has become a common online activity with the development of social networks such as Facebook, Instagram, Snapchat etc. Users often apply different kinds of image processing operations to make their image visually more appealing. Those operations are among users often called filters or effects. Some applications offer more complex effects such as style transfer. Style transfer algorithms enable the visual style transfer from one image to another while preserving the semantic content of the target image. Most frequent use case is style transfer from famous artwork to regular photographs. That way user can make his photograph to look like it has been painted by Van Gogh, Munch, Picasso etc.

In style transfer, the main underlying problem is finding an image representation which enables the disentanglement between the style and content information. This is necessary to avoid the loss of semantic content in the target image due to the style transfer. Seminal work from Gatys et al. [3] showed that this is possible with representation provided by ImageNet pretrained convolutional neural network. They showed that the image content is preserved in models hidden representation, while the style information can be expressed as the correlation between different convolutional kernel responses. Correlations are usually aligned in $c \times c$ Gram matrix, where $c$ is number of output channels in a convolutional layer. This method is discussed in detail in the section 2.

Recent work [5, 7, 17, 15, 16, 12] on style transfer almost exclusively utilizes deep learning models in different ways. Gatys et al. [3] based their style transfer on slow optimization process which minimizes combination of style and content loss computed by the deep model. This makes it unsuitable for real-time processing, especially on limited mobile hardware. Further work [15, 7, 5] tried to avoid the optimization process during the inference by training a neural network which performs style transfer with a single forward pass. Ulyanov et al. [15] trained computationally and memory efficient model for style transfer, but limited to single texture or style. This requires separate training and model for each style a user would like to apply. Others [1, 9] presented methods which are able to synthesize multiple textures/styles defined before the training process. Huang and Belongie [5] similarly to [1] used adaptive instance normalization layer to transfer feature statistics from style to content image. However, their method is able to perform transfer for arbitrary style represented with

an image. They train the decoder which has to reconstruct an image from stylized representation of a content image. The usual content and style loss are then calculated w.r.t. the features pooled from the image produced by the decoder. Style transfer has been also studied within the GAN framework [9, 18]. CycleGAN [18] performs image-to-image translation without the requirement for paired images in the training set. However, it requires multiple image instances of same style, while the trained model is bounded to single style.

Complexity of deep models leads to high expressiveness, but also makes them susceptible to adversarial examples [14]. An adversarial example can be defined as input which is close to a natural input, but is misclassified. For example, it can be obtained by small perturbation of a regular image (unnoticeable to human eye), but enough to completely "confuse" the model. This behaviour is undesirable, especially for models in life critical applications. Madry et al. [10] showed that adversarial robustness can be improved by training with adversarial examples. Ilyas et al. [6] show that models prone to adversarial attacks probably over-rely on so called non-robust features which are highly predictive, but brittle. Adversarial training discourages the model to use these non-robust features, and encourages to focus on the robust ones instead. Further details of the research on adversarial examples go behind the scope of this work, but the effect of adversarial robustness on style transfer will be discussed. Nakano [11] was first to show the effectiveness of robust models for style transfer. Further discussion on this topic led to conclusions that robustness is much more important for style component of the loss function. In this work we shed more light on this interesting phenomenon.

# 2. Method

Most deep architectures sequentially build abstract representation through series of convolutional and pooling layers. At the early layers hidden representation has high resolution and poor semantic content. In deep layers some part of spatial information is lost due to the subsampling, but high-level semantic information is present which is necessary for complex visual recognition. Gatys et al. [3] showed that hidden representations from deep convolutional networks trained for recognition tasks are suitable for disentangling the content from style information.

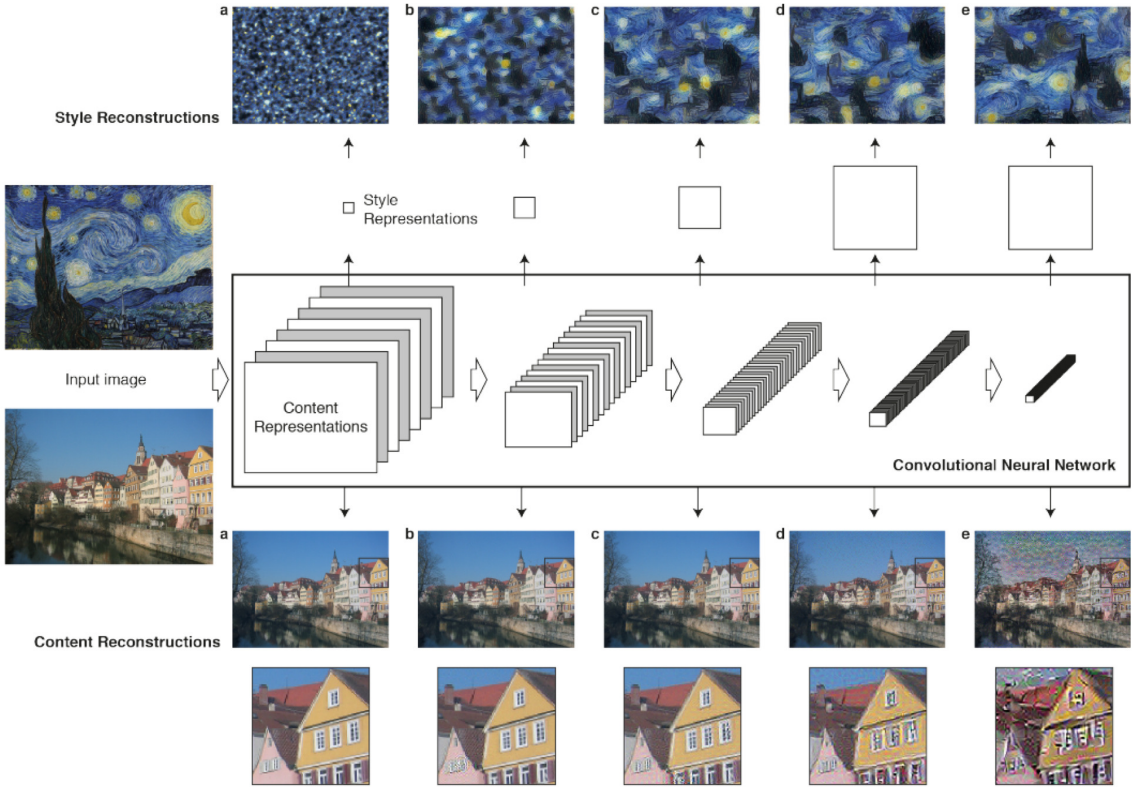


**Figure 2.1:** Style and content reconstructions based on image representations provided by pretrained convolutional neural network as proposed in Gatys et al. [3]. Image downloaded from [3].

Image information in CNNs hidden representation is preserved in surprisingly high manner. Bottom part of the figure 2.1 shows image reconstructions based on the outputs of con-

volutional layers at different depth of VGG-19 architecture. Reconstructions are obtained with optimization process starting from the input noise, requiring that the hidden representation computed with VGG model matches the representation computed from the original image. We observe that the reconstructions from early layers are near perfect, deteriorating towards the deep layers. These representations are used for content reconstruction in neural algorithm of artistic style [3].

For style representation Gram matrix $G$ is computed capturing correlations between different kernel outputs inside one convolutional layer. Output of the convolutional layer is usually $c \times h \times w$ tensor, where $h$ and $w$ denote height and width, and $c$ the number of output channels. Further, it can be aligned in a matrix $F \in \mathbb{R}^{c \times n}$ where $n = h \cdot w$. Then the Gram matrix can be computed as:

$$G = \frac{1}{n} F \cdot F^T \tag{2.1}$$

In order to synthesize the new image $I_n$ with style of an image $I_s$ and content of image $I_c$ a simple optimization procedure is proposed. The new image $I_n$ is initialized with random noise, and optimized to minimize weighted combination of style and content loss:

$$\mathcal{L}(I_n, I_c, I_s) = w_c \mathcal{L}_c(I_n, I_c) + w_s \mathcal{L}_s(I_n, I_s) \tag{2.2}$$

Content loss $\mathcal{L}_c$ corresponds to the L2 loss computed with respect to feature tensors obtained by applying the CNN to inputs $I_n$ and $I_c$. More specifically, authors used outputs of the layer 'conv4_2' from VGG-19 [13]. Style loss $\mathcal{L}_s$ corresponds to the weighted sum of squared differences between the Gram matrices of feature tensors pooled at different convolutional layers $l$:

$$\mathcal{L}_s(I_n, I_s) = \sum_l w_l [G_l(I_n) - G_l(I_s)]^2 \tag{2.3}$$

Although in the original paper all experiments were performed with VGG-19 model, proposed method should generalize to other deep architectures as well. We test this hypothesis on popular ResNet-50 architecture [4]. Additionally, we compare the regular ImageNet initialization with robust variant [2] trained on ImageNet with adversarial examples.

# 3. Experiments

## 3.1. Influence of model robustness on style transfer

We first compare content and style reconstructions with three variants of ResNet-50 model: randomly initialized, regular ImageNet and robust ImageNet weights. Figure 3.1 shows original content (a) and style (b) image used in reconstruction experiments. Content image shows traditional grass mowers in Kupres, Bosnia and Herzegovina. Style image shows famous artwork called "The Scream" by expressionist Edward Munch.

(a) (b)

**Figure 3.1:** Content (a) and style (b) image for reconstruction experiments.

Content reconstructions are obtained as follows. First, features from the original image are computed and saved to memory. Second, random image is initialized with gaussian noise. Third, we run a process which iteratively optimizes the noise image in order to minimize the L2 distance between features from original image and the features from the noise image. We set maximum number of iterations to 5000, and use ADAM optimizer [8] with learning rate set to 0.05. Figure 3.2 shows reconstructions based on features from 5 different stages of ResNet-50 model: first convolutional layer (also known as stem or conv1), and the last layer of corresponding four residual blocks rb1-rb4. Layer conv1 is the shallowest, and the layer from the last residual block (rb4) is the deepest.


**Figure 3.2:** Content image reconstructions based on features pooled at 5 different stages (rows) of Resnet-50 model: conv1, rb1, rb2, rb3 and rb4. We consider three variants of initialization: random, non-robust and robust ImageNet weights.

We observe that generally reconstruction quality deteriorates as the layer depth increases. Almost perfect reconstructions from the first convolutional layer are somewhat expected. Interestingly, we can recognize the original image in the reconstructions from rb1 and rb2 for randomly initialized model, despite the image being run through series of random non-linear transformations.

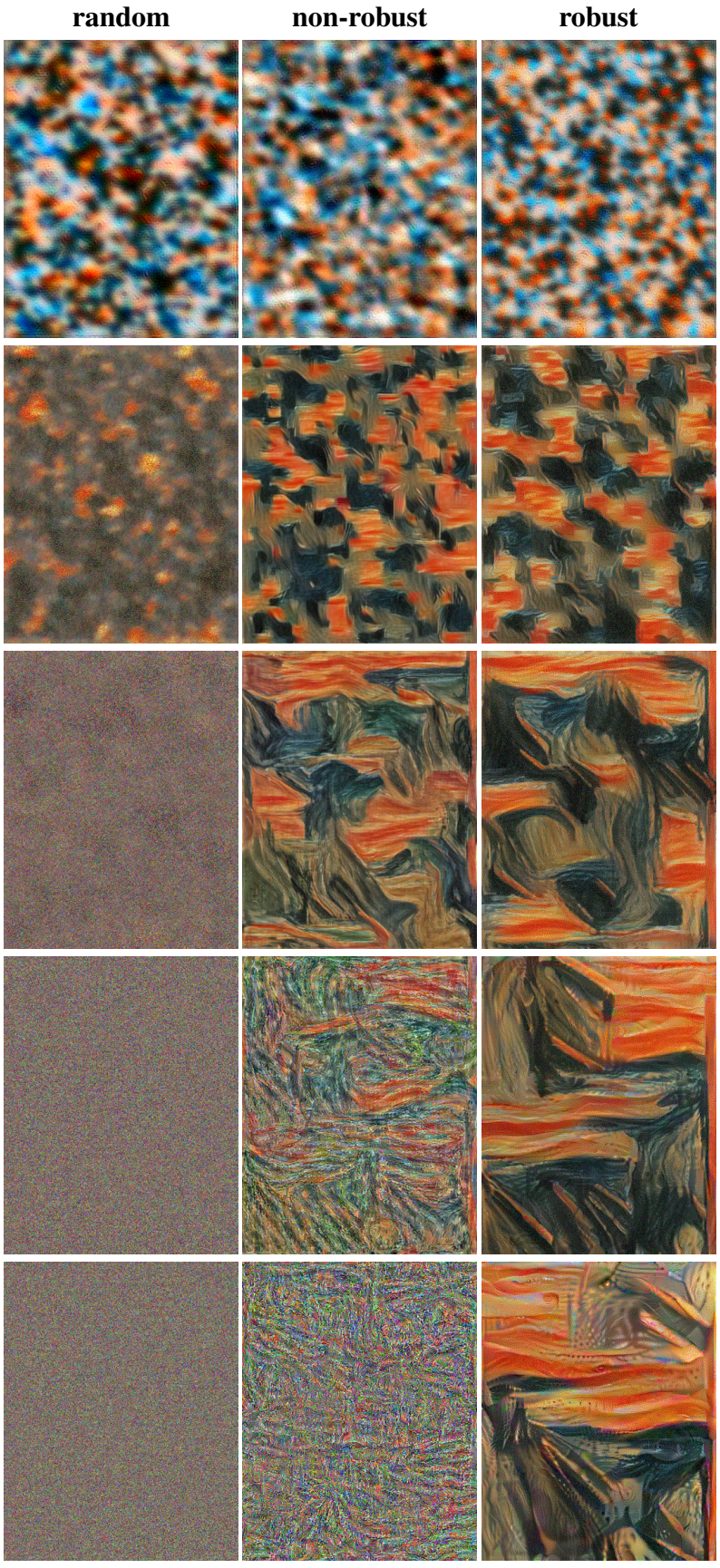


**Figure 3.3:** Style image reconstructions based on gram matrix calculated on top of features pooled at 5 different stages (rows) of Resnet-50 model: conv1, rb1, rb2, rb3 and rb4. We consider three variants of initialization: random, non-robust and robust ImageNet weighs.

We observe that robust model achieves better reconstruction quality consistently, especially for deep layers. Non-robust model relies on non-robust features which are often high frequency, and less on the low frequency features. Because of that, it is easier to "fool" him and find input setup which produces features close enough to the features of the original image. Training with adversarial examples encourages robust model to rely less on the high frequency features and therefore captures more global information.

Style reconstructions are obtained similarly, except that we minimize L2 distance between Gram matrices computed from features, and not between features directly. Figure 3.3 shows style reconstructions. As expected, in style reconstructions there is no spatial information preserved. We observe that early layers capture more local patterns, while in the last layers we can notice some more global structure. Comparing the three weight initialization variants, we can make similar conclusions as in content image reconstruction experiment. Random initialization captures some style information in first layer. First three rows look similar for non-robust and robust model, but after that strong noise appears in non-robust style reconstructions. This is important observation, because if we include these layers in loss function during style transfer it will introduce the noise in whole procedure.

Nakano [11] claims that non-robust models are not suitable for style transfer. However, we think that this claim is too strong and can be mitigated with better hyper-parameters. Instead of setting equal weight to all style loss components, we propose to set lower weights for components corresponding to deeper layers. That way, we introduce less noise from style reconstruction. Figure 3.5 compares style transfers performed with non-robust and robust ResNet-50 model. For content loss we use features from last convolutional layer from third residual block, while for the style loss we use features from all four blocks. Style loss weights are set to $[10^5, 10^4, 10^3, 10^2]$ for residual blocks 1-4 respectively. We used same content image as before, while the figure 3.4 shows four style images respectively. There is no exact way to measure the quality of style transfer. It is subjective and depends exclusively on users perception. Nevertheless, we can derive some general conclusions. We show that style transfer works with ResNet architecture also, even with non-robust model.

**Figure 3.4:** Style images used for style transfer comparison between non-robust and robust model.


**Figure 3.5:** Style transfers with non-robust and robust variant of ResNet-50 model.

However, robust model generally produces images which are more visually appealing. For example, in the first row we can notice some high brightness areas which are not present in the robust variant. In the third row, we can notice that regular model fails to apply style to the upper part of the images which corresponds to sky.

## 3.2. Artistic style transfers

Figure 3.6 shows four style transfers with different content and style images. Three columns correspond to original content image, style image and result of style transfer respectively. We run optimization procedure for 500 iterations with LBFGS optimizer as in the original paper. Style transfers are usually successful, however there is room for improvements. During the style transfer some details are lost. This is particularly problematic in scenes where those details are important and add significant value to the scene (e.g. human face). While running these experiments, we also observed that some style images are more suitable for some content images and less for other. For example, when we applied "The Scream" style on content image from first row we considered loss of details, especially on the face of a player. This leads to style transfer which human eye does not find pretty.

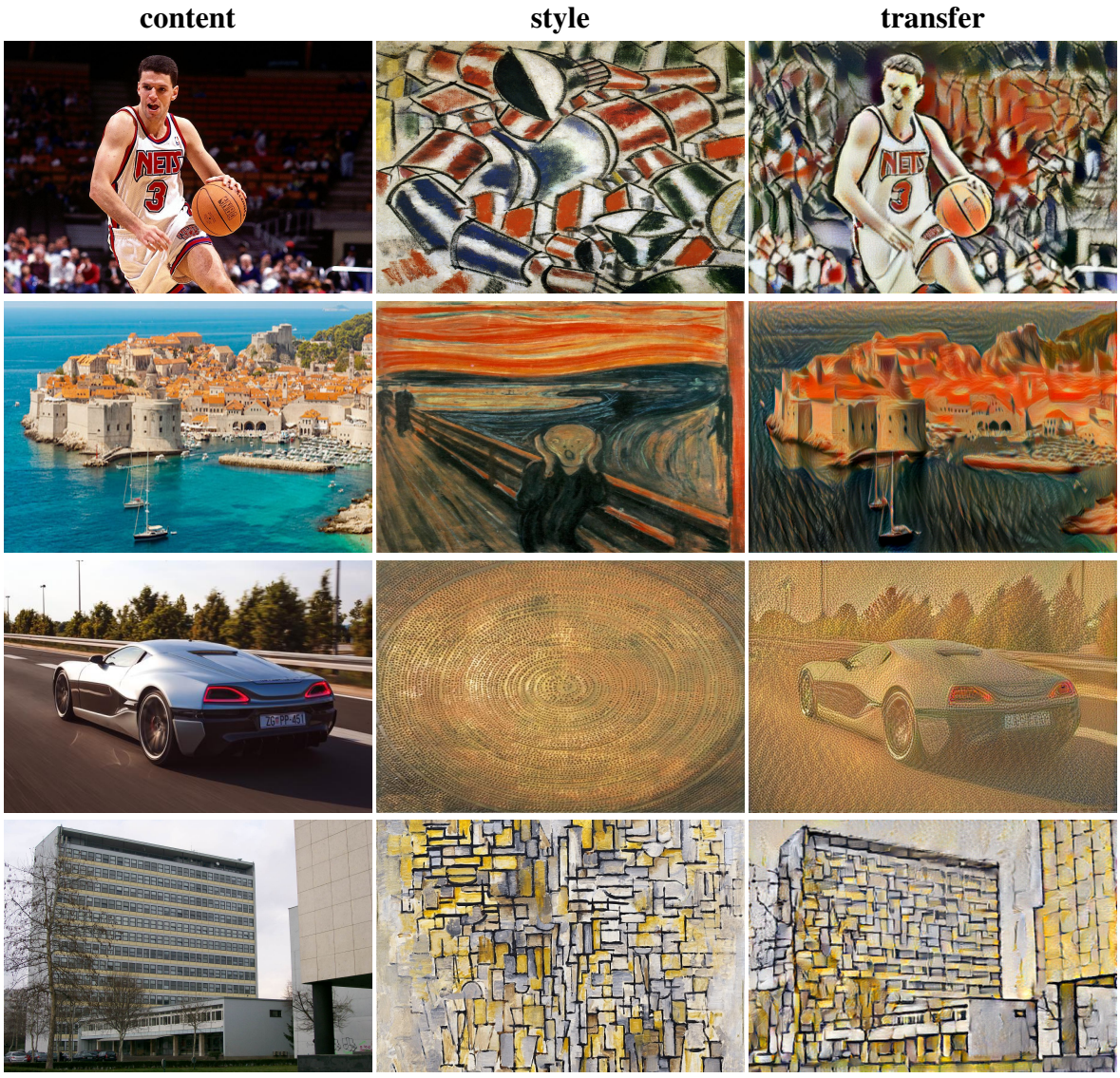


**Figure 3.6:** Style transfers with different content and style images.

## 3.3. Hyper-dependency on hyper-parameters

Original style transfer procedure proved to be extremely dependant on the choice of the hyper-parameters. For example, it is known that using ADAM instead of LBFGS optimizer leads to much slower convergence and significantly worse style transfers. Another important set of hyper-parameters are weights in the style component of loss function. Figure 3.7 shows style transfers with six different sets of style loss weights. We set weights to $[x, 10^{-1}x, 10^{-2}x, 10^{-3}x]$ for features from rb1-4 respectively. We start from $x = 10^2$ and increase it each time $10\times$ obtaining that way totally six style transfers with different sets of hyper-parameters. We observe that setting the style weights too low leads to no style transfer whatsoever. However, at next iteration $x = 10^3$ we already achieve satisfying transfer. Best results are obtained for $x = 10^4$. Increasing it further guides the process to include too much style information and the final result is not visually appealing.

**Figure 3.7:** Style transfer with six different sets of hyper-parameters.

# 4. Conclusion

Development of deep learning led to interesting findings in the field of style transfer also. Convolutional neural networks can be used to disentangle content and style information of an image. In this work we extend the original style transfer procedure for VGG-19 model to new ResNet-50 model for image classification. We showed that style transfer procedure is not architecture dependant. We also compared robust and non-robust variants of ResNet-50 model. We showed that robust model leads to better and more stable style transfer procedure. With careful choice of hyper-parameters style transfer is possible with non-robust model also. We demonstrate hyper-dependency of original style transfer procedure on the choice of weights in the style component of loss function. Setting the weights too low leads to weak style transfer, and setting them too high leads to loss of content image details. In this work we performed experiments which indicated some connections between the robustness and style transfer at the empirical level. However, there is more work to be done especially on developing and looking further into this phenomena from the theoretical perspective.

# 5. Bibliography


[1] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[2] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019.

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[6] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

[7] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Diversified texture synthesis with feed-forward networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3920–3928, 2017.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[11] Reiichiro Nakano. A discussion of 'adversarial examples are not bugs, they are features': Adversarially robust neural style transfer. *Distill*, 2019. https://distill.pub/2019/advex-bugs-discussion/response-4.

[12] Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, pages 26–36. Springer, 2016.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[15] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *ICML*, volume 1, page 4, 2016.

[16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.

[17] Xin Wang, Geoffrey Oxholm, Da Zhang, and Yuan-Fang Wang. Multimodal transfer: A hierarchical deep convolutional neural network for fast artistic style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5239–5247, 2017.

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.