Neural Style Transfer for Image Stylization

Qifeng Xiang *

School of Big Data & Software Engineering, Chongqing University, Chongqing, China

* Corresponding Author Email: 20215540@cqu.edu.cn

Abstract. Image style migration, exploring the transformation of visual styles from one image to another, has become a focal point in computer vision research. The semantic and stylistic features of images are difficult to express directly through mathematical models, which greatly increases the difficulty of image stylization. Fortunately, approaches based on deep learning have shown promise in extracting deep semantic information from images, facilitating notable advancements in image style transfer. However, achieving a balance between content preservation and style transformation remains a formidable challenge. This paper introduces a neural style transfer network (NSTN) that aims to maintain image semantics while performing style transfer effectively. The NSTN framework comprises a process block, a style block, and an ascension decoder, working in concert to achieve nuanced style shifts while preserving the content integrity. Implementation results on the WikiArt and COCO datasets demonstrate the model's effectiveness in achieving a harmonious balance between content preservation and style integration.

Keywords: Neural Style Transfer, Artistic Creativity, Image Style Transfer, Deep Learning in Art.

1. Introduction

The field of neural style transfer, which bridges artistic creativity with technological innovation, aims to democratize art by applying the styles of historic artworks to contemporary images. This fusion not only seeks to make artistic expression more accessible but also presents significant technical challenges, such as blending different artistic styles while preserving the content's integrity. Gatys et al.'s pioneering work on utilizing Convolutional Neural Networks (CNNs) for this purpose laid the foundation for this area of study, showing the potential for technology to capture and replicate the essence of artistic styles [1].

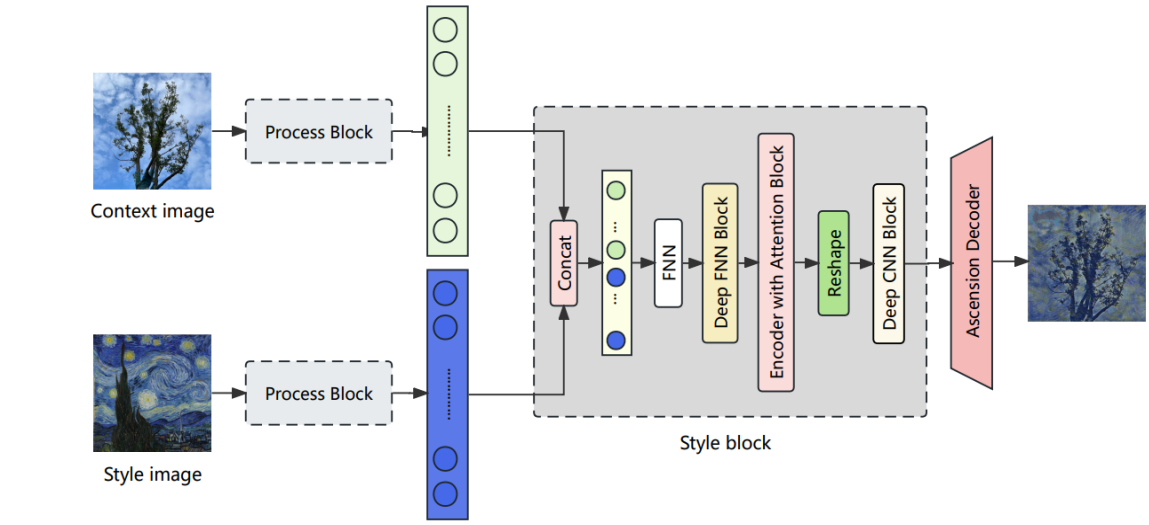
Further research has aimed to refine these techniques for greater efficiency, quality, and accessibility. For instance, Li et al. provided a novel interpretation of neural style transfer, framing it as a domain adaptation problem, which deepened the understanding of how styles are represented and transferred. Jing et al.'s comprehensive review offered a taxonomy of the current algorithms, facilitating a more profound comparative analysis of these methods [2, 3].

Innovations such as Direction-aware Neural Style Transfer by Wu et al. addressed the production of more natural and vivid stylizations by focusing on stroke direction during the style transfer process [4]. Depth-aware Neural Style Transfer, introduced by Liu et al., incorporates depth preservation to ensure the maintenance of image layout and semantic content, enhancing the technique's capability [5].

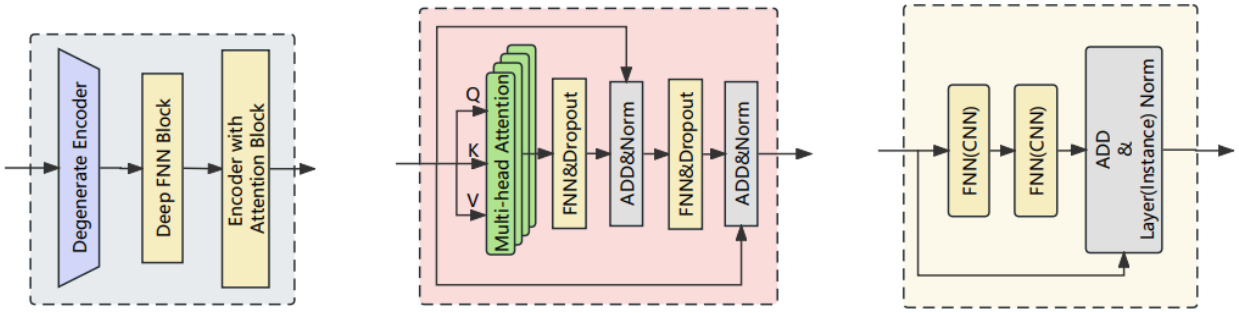
The exploration of neural style transfer has continued to evolve, with research delving into various aspects and challenges of the technique. Gatys et al. extended their work to include controlling factors in neural style transfer, allowing for more precise manipulation of spatial location, color, and scale, thereby improving stylization control and quality [6]. The study on Stereoscopic Image Style Transfer by Gong et al. introduced a method for applying style transfer to stereoscopic images while maintaining view consistency, which is crucial for a comfortable visual experience [7].

The importance of maintaining structural integrity in the style transfer process was highlighted by Cheng et al., who introduced Structure-Preserving Neural Style Transfer, focusing on preserving global structures and local details of the images [8]. Moreover, efforts to enhance texture details and maintain directionality in stylized images were furthered by Wu et al. in their work on Direction-aware Neural Style Transfer with Texture Enhancement [9].

A Review on Neural Style Transfer by Li et al. provides a comprehensive overview of NST methods, shedding light on both image- and model-optimization-based methods and their applications, underscoring the vast potential and ongoing development in NST research [10]. Further extending the application of NST, Shibly et al. proposed an advanced artistic style transfer method that efficiently reduces style and content loss during the transfer process, marking a significant improvement in NST efficiency and speed [11].



(a) Neural Style Transfer Network



(b) Process Block (c) Encoder with Attention Block (d) Deep FNN(CNN) Block

Figure 1. The overall flowchart of the NSTN, which details the structure of the model network.

In the realm of neural style transfer, expressing style characteristics involves capturing the overall information of an image, which necessitates understanding the long-distance dependencies between pixels. Traditional convolutional neural network (CNN)-based methods for feature extraction are hampered by their small receptive fields, due to the limited size of convolution kernels, making it challenging to establish long-range semantic associations. While stacking deep neural networks can increase the receptive field, this approach suffers from low computational efficiency. To address this issue, our paper leverages the transformer architecture as the foundation to propose a neural style transfer network (NSTN). This model integrates key modules: Degenerate Encoder, Deep Feedforward Neural Network Block (DFB), Encoder with Self-Attention Block (ESAB), Style Block, Deep CNN Block (DCB), and Ascension Decoder. Each serves crucial functions, including dimensionality reduction, content analysis, semantic integrity maintenance, feature fusion, refinement, and final reconstruction, ensuring effective style transfer while preserving content's semantic meaning.

The effectiveness of our approach is supported by experimental results. Utilizing a varied dataset that spans a broad spectrum of artistic styles, from the subtle hues of Impressionism to the stark geometric shapes of Cubism, our model exhibits an exceptional capability to replicate these styles accurately on content images. These experiments underscore the model's adeptness at capturing the nuances of different art genres, facilitating transformations that retain the content's core while

embodying the distinctive visual language of the target style. Our experiments, conducted on datasets such as the Architecture dataset [12] and MS-COCO [13], yielded positive results, demonstrating the model's efficiency in preserving semantic information of images while adeptly performing style transformations.

2. Method

To solve the problem of long-distance dependence of the image as well as to maintain the image positional integrity by using the self-attention mechanism, we sample the original image for dimensionality reduction and input it into the model. Inputting a content image $I_c \in \mathbb{R}^{H\times W\times 3}$ and inputting the feature image $I_s \in \mathbb{R}^{H\times W\times 3}$ when creating the model.

The overall flow chart is shown in Figure1. We take the image for input, dimensionality reduction and spreading in Degenerate Encoder. After that, two 1D vectors are obtained by fully extracting the image features through Deep FNN Block and Encoder with Self-Attention Block. The two vectors are directly concatenated into the style block for further style transformation, and then reshaped into the image shape by Encoder with Self-Attention Block and learnt by Deep CNN Block. Finally, the image is reshaped by Ascension Decoder.

2.1. Process Block

2.1.1. Degenerate Encoder

We used 6 CNN layers to convolve the input image, where each block contains 2 layers of 3×3Conv and 1 layer of 3×3Conv with a step size of 2. A total of 2 blocks samples the original image to $I \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 16}$. The last layer selects a smaller convolution kernel to avoid spreading the flat tensor to be too large. After that we flatten it to 1 dimension $I \in \mathbb{R}^{1\times(H\times W)}$ and input it to the network.

2.1.2. Deep FNN Block (DFB)

To deeply understand the image content, we use a deep fully connected feedforward neural network (FNN) and connect them residually. Using this structure allows deeper learning of the information inside the image and alleviates the problem of difficult gradient propagation in deep networks. For the input 2-dimensional image X, we perform this calculation:

$$X' = \mathcal{F}_{FNN}(X'') \tag{1}$$

$$X = \mathcal{F}_{FNN}(X') + X'' \tag{2}$$

Layer normalization (LN) is applied after each block [14].

2.1.3. Encoder with Self-Attention Block (ESAB)

We try to dynamically focus on different parts of the input image using the self-Attention mechanism. And for content image and style image, we used two different encoders to get their features.

For the input data $Z \in \mathbb{R}^{1\times n}(n = H \times W)$, we begin by conducting self-attention following dimensionality reduction accomplished by a fully connected feed-forward neural network. Within the self-attention mechanism, we incorporate a multi-head self-attention module along with a fully connected feed-forward neural network. We create three feed-forward layers $W_q, W_k, W_v \in \mathbb{R}^{n\times(n\times seq)}$, 'seq' represents the length of the input sequence, which is used to derive the query (Q), key (K), and value (V) components:

$$Q = Z \times W_q, K = Z \times W_k, V = Z \times W_v \tag{3}$$

In that case the shape becomes $\mathbb{R}^{1\times(\mathrm{n}\times\mathrm{seq})}$, after that reshape $\mathrm{K},\mathrm{Q},\mathrm{V}\in\mathbb{R}^{\mathrm{seq}\times\mathrm{heads}\times\mathrm{depth}}$, where $\mathrm{depth}=\frac{\mathrm{n}}{\mathrm{heads}}$, the seq dimension and heads dimension are then transposed for the inner product calculation $\mathbb{R}^{\mathrm{heads}\times\mathrm{seq}\times\mathrm{depth}}$. The attention (ATT) is implemented as follows:

$$ATT=\frac{SoftMax(Q\times K^{T})}{\sqrt{depth}}\times V \tag{4}$$

Now, convert back the seq and heads dimensions of ATT, reshape it back to $\mathbb{R}^{1\times(n\times seq)}$, and use 2 layers of FNN to get the output:

$$output=\mathcal{F}_{FNN}(ATT)+Z \tag{5}$$

For each layer dropout and LN are used [14].

2.2. Style Block

For the above block, we got two feature vectors $I\in\mathbb{R}^{1\times n}$ after using Process Block for each input image. Next the following calculations are performed on the two vectors:

$$Z'=Concat\left(Z_{content},Z_{style}\right) \tag{6}$$

$$Z=\mathcal{F}_{FNN}(Z') \tag{7}$$

$\mathcal{F}_{FNN}$ outputs a layer of vectors with the shape $1\times(\frac{H}{4}\times\frac{W}{4}\times 16)$ and then after passing it through a DFB and ESAB reduces it to the shape $I\in\mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times 16}$ as input to the model.

2.2.1. Deep CNN Block

Like Deep Embedding Block, here the FNN layer is replaced with a Conv layer, which is calculated as follows:

$$X'=\mathcal{F}_{Conv}(X'') \tag{8}$$

$$X=\mathcal{F}_{Conv}(X')+X'' \tag{9}$$

After each block there is an Instance Normalization, which is normalized for each pixel of the channel. We have used 5 layers of Deep CNN Block for deep mining the features of synthetic images.

2.3. Ascension Decoder

In contrast to Degenerate Encoder, each block of this decoder uses 1 layer of 2×2UpSampling2D and 2 layers of 3×3Conv, for a total of 2 blocks, and the last layer reduces the image to the original image size $I_o\in\mathbb{R}^{H\times W\times 3}$ using 3 convolution kernels.

2.4. Loss Function

For the image stylistic migration model, considering the need to maintain the overall style of the original image, but also the need to learn the color stroke style of the stylistic image and so on. So, we used two loss functions corresponding to content loss $\mathcal{L}_c$ and style loss $\mathcal{L}_s$, and multiplied by different weights to adjust whether the overall image is closer to the content image or the style image.

For $\mathcal{L}_c$, we created the loss function based on the shallow {*block1_conv1, block2_conv1*} of VGG19, which is calculated as follows:

$$MSE\left(I_{c_i},I_{o_i}\right)=\frac{1}{M_i}\sum_{j=0}^{M_i}(I_{c_{ij}}-I_{o_{ij}})^2 \tag{10}$$

$$\mathcal{L}_c=\frac{1}{N_l}\sum_{i=0}^{N_l}MSE\left(\phi_i(I_c),\phi_i(I_o)\right) \tag{11}$$

Where $\phi_i(\cdot)$ denotes the features extracted from the i-th layer of the pre-trained VGG19, N_l is the number of layers, M_i is the number of elements of the features in the i-th layer, and $I_{c_{ij}}$ as well as $I_{o_{ij}}$ is the j-th element in the i-th layer.

For $\mathcal{L}_s$, we chose the backward level {*block1_conv2*, *block2_conv2*, *block3_conv3*, *block4_conv3*} in each block of VGG19 and included all the blocks, to ensure that we can learn the painting style of the style image while also learning the more abstract part of the style image, and the following is the computation method:

$$\mathcal{F}_i(I) = Flatten\big(\phi_i(I)\big) \tag{12}$$

$$Gram(I) = \frac{I^T \times I}{W \times H} \tag{13}$$

$$\mathcal{L}_s = \frac{1}{N_l}\sum_{i=0}^{N_l} MSE\left(Gram\big(\mathcal{F}_i(I_c)\big), Gram\big(\mathcal{F}_i(I_o)\big)\right) \tag{14}$$

In this context, $\mathcal{F}_i(\cdot)$ denotes flattening the features extracted from the i-th layer of the pre-trained VGG19 into one-dimensional space. $Gram(\cdot)$ denotes computing the inner product of the vectors, which is the Gram matrix, and then normalize.

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s \tag{15}$$

Where λ_c and λ_s are custom hyperparameter weights.

3. Experiments

3.1. Experimental condition

3.1.1. Dataset

In our experiments, we used two main datasets: the content dataset and the style dataset. The content dataset consists of the Architecture dataset [12] as well as some of the images in MS-COCO [13], which is a dataset containing a variety of architectural photographs. This dataset was chosen for its diversity of scenes and objects, which provides a comprehensive basis for assessing content preservation. The Style dataset consists of a selection of paintings from the WikiArt dataset [15], representing a variety of art styles, including Impressionism, Cubism, and Abstract Art.

3.1.2. Weights and hyperparameters setting

We employed a batch size of 8 and configured both DFB and DCB with 5 layers to facilitate complex feature interactions. Conversely, ESAB was designed with a more streamlined 2-layer architecture to emphasize efficient attention-driven feature synthesis. Training utilized the Adam optimizer [16], iterating 18,400 times per style map to refine the nuances of each artistic influence. Each CNN layer was outfitted with 128 filters, ensuring a detailed and robust feature extraction process. The loss function weights were carefully chosen, with a content loss weight of 4 and a style loss weight of 0.4, to ensure a balanced representation of content and style.

3.1.3. Training time

We trained each of our different style map models for 2 hours on NVIDIA GeForce RTX 4090 GPUs.

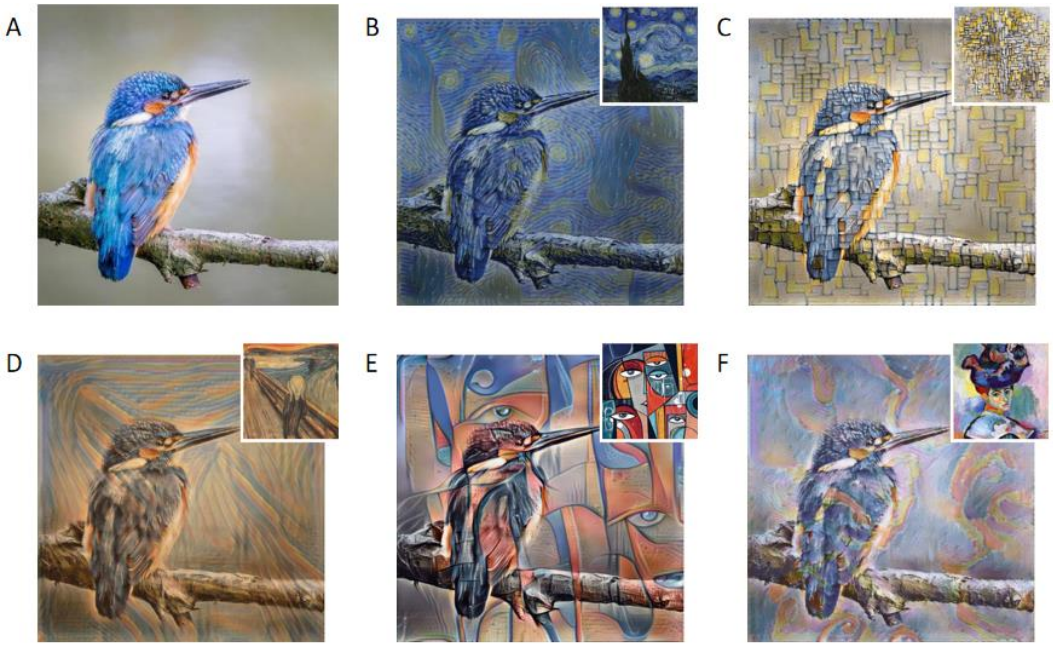


Figure 2. An image that combines the contents of a photograph with the styles of several famous works of art. A Blue Kingfisher standing on a branch is shown in A. For each image, the style map is in the upper right corner of the converted image. B "The Starry Night" by Vincent van Gogh, 1889. C "Composition 10 in Black and White" by Piet Mondrian, 1915. D "The Scream" by Edvard Munch, 1893. E "Woman in Hat and Fur Collar" by Pablo Picasso, 1937. F "Woman with a Hat" by Henri Matisse, 1905.

3.2. Results

Figure 2 presents the results of our experiments through a series of striking images that juxtapose the original content images and the adapted images of six different styles extracted from our curated stylistic dataset. The contrasting visualization highlights the model's ability to retain the essence of the original content while seamlessly integrating the unique stylistic elements of each of the chosen art forms. The transformations vividly demonstrate the model's ability to navigate and embody the intricate nuances of art, ranging from the subtle interplay of light and color characteristic of Impressionism to the obvious geometric distortion’s characteristic of Cubism. The transformation of each image clearly demonstrates the model's ability to master and translate complex artistic conventions in the process of stylistic transitions.

3.3. Limitations

3.3.1. Large number of participants

The experimental model architecture is characterized by a fairly large number of parameters, which in turn requires the use of large amounts of graphics processor (GPU) memory, particularly video memory (VRAM), during the training and inference phases. This VRAM density increases the demand for computational resources and inherently limits the availability of the model with advanced, high-end GPU capabilities. As a result, the availability of the model may be reduced for users without such resources, potentially narrowing its utility and application scope. In addition, this limitation may also pose a significant challenge in scaling up the model for batch image processing or real-time applications, which require not only high-speed processing but also optimized memory management to ensure responsiveness and efficiency.

3.3.2. Insufficient detail textures

The model exhibits significant deficiencies in replicating detailed textures, which is particularly noticeable when complex patterns and nuances are important aspects of the content image. The stylization process appears to ignore these subtle details, resulting in a translated image that lacks the

depth and clarity of the original image. This limitation suggests that the feature extraction capabilities of the neural network may overgeneralize, in which case the emphasis on broader stylistic strokes may mask the need for finer texture fidelity. As a result, this may reduce the applicability of the model to detail-demanding tasks such as digital art creation or detailed visual content generation.

4. Conclusion

In this work, we explored the application of deep learning in the domain of image style transfer, particularly focusing on the impacts of simple network aggregation and self-attention mechanisms on this process. We carried out a thorough analysis of the style transformation process by combining deep stacked fully connected networks (FNNs), convolutional neural networks (CNNs), and self-attention mechanisms, with a focus on striking a balance between content preservation and style integration. Each module is designed to serve crucial functions within the style transfer process dimensionality reduction, deep content analysis, maintaining semantic integrity, feature fusion, refinement, and final reconstruction. Through experimentation, our model was able to replicate a wide range of artistic styles, from the subtle tones of Impressionism to the bold geometric shapes of Cubism, proving its ability to maintain the essence of the content while reflecting the unique visual language of the chosen style.

Despite limitations related to the requirement for substantial GPU memory and insufficient detail texture replication, this study contributes a novel approach to the development of neural style transfer technology. By combining deep networks and self-attention mechanisms, our method has shown potential in achieving style transformations while preserving image content, also highlighting directions for future work, including reducing the model's dependency on high-end GPU resources, improving the precision of texture replication, and further enhancing the accuracy and naturalness of the style transfer process.

References

[1] Gatys, L.A., Bethge, M., Hertzmann, A., & Shechtman, E., Preserving Color in Neural Artistic Style Transfer, ArXiv, abs/1606.05897, (2016).

[2] Li, Y., Wang, N., Liu, J., & Hou, X., Demystifying Neural Style Transfer, International Joint Conference on Artificial Intelligence, (2017).

[3] Jing, Y., Yang, Y., Feng, Z., Ye, J., & Song, M., Neural Style Transfer: A Review, IEEE Transactions on Visualization and Computer Graphics, 26, 3365 - 3385 (2017).

[4] Wu, H., Sun, Z., & Yuan, W., Direction-aware Neural Style Transfer, Proceedings of the 26th ACM international conference on Multimedia, (2018).

[5] Liu, X., Cheng, M., Lai, Y., & Rosin, P.L., Depth-aware neural style transfer, International Symposium on Non-Photorealistic Animation and Rendering, (2017).

[6] Gatys, L.A., Ecker, A.S., Bethge, M., Hertzmann, A., & Shechtman, E., Controlling Perceptual Factors in Neural Style Transfer, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3730 - 3738 (2016).

[7] Gong, X., Huang, H., Ma, L., Shen, F., Liu, W., & Zhang, T., Neural Stereoscopic Image Style Transfer, European Conference on Computer Vision, (2018).

[8] Cheng, M., Liu, X., Wang, J., Lu, S., Lai, Y., & Rosin, P.L., Structure-Preserving Neural Style Transfer, IEEE Transactions on Image Processing, 29, 909 - 920 (2020).

[9] Wu, H., Sun, Z., Zhang, Y., & Li, Q., Direction-aware neural style transfer with texture enhancement, Neurocomputing, 370, 39 - 55 (2019).

[10] Li, J., Wang, Q., Chen, H., An, J., & Li, S., A Review on Neural Style Transfer, Journal of Physics: Conference Series, 1651 (2020).

[11] Shibly, K.H., Rahman, S., Dey, S.K., & Shamim, S.H., Advanced Artistic Style Transfer Using Deep Neural Network, (2020).

[12] Xu, Z., Tao, D., Zhang, Y., Wu, J., & Tsoi, A.C., Architectural Style Classification Using Multinomial Latent Logistic Regression, European Conference on Computer Vision, (2014).

[13] Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L., Microsoft COCO: Common Objects in Context, European Conference on Computer Vision, (2014).

[14] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I., Attention is All you Need, Neural Information Processing Systems, (2017).

[15] Phillips, F.Y., & Mackintosh, B., Wiki Art Gallery, Inc.: A Case for Critical Thinking, Issues in Accounting Education, 26, 593 - 608 (2011).

[16] Kingma, D.P., & Ba, J., Adam: A Method for Stochastic Optimization, CoRR, abs/1412.6980 (2014).