MULTIPLE STYLE TRANSFER VIA VARIATIONAL AUTOENCODER

Zhi-Song Liu, Vicky Kalogeiton and Marie-Paule Cani

LIX, École Polytechnique, CNRS, IP Paris

https://www.lix.polytechnique.fr/geovic/project-pages/icip-style-transfer

ABSTRACT

Modern works on style transfer focus on transferring style from a single image. Recently, some approaches study multiple style transfer; these, however, are either too slow or fail to mix multiple styles. We propose ST-VAE, a Variational AutoEncoder for latent space-based style transfer. It performs multiple style transfer by projecting nonlinear styles to a linear latent space, enabling to merge styles via linear interpolation before transferring the new style to the content image. To evaluate ST-VAE, we experiment on COCO for single and multiple style transfer. We also present a case study revealing that ST-VAE outperforms other methods while being faster, flexible, and setting a new path for multiple style transfer.

1. INTRODUCTION

Style transfer is a well-visited topic [1, 2, 3, 4]. Given a target and a reference image, its goal is to synthesize an image with content from the target and style from the reference. Typically, 'style' refers to color, texture, or brushstroke [5, 6].

Most style transfer works are limited as they typically focus either *solely* on single style [1, 2, 4, 7] or on *one set* of styles [5, 8, 9]. In real-life applications, however, a user is rarely satisfied with a single style and instead seeks for multiple ones usually by iteratively applying styles [10]. To address multiple style transfer, the common practice is to interpolate different styles or assign different weights in the feature space [2, 3]. Besides slow, the main limitation of such methods is that features are nonlinear and linear interpolation cannot guarantee style mixture in spatial space.

To tackle these issues, we introduce a Variational AutoEncoder for latent space-based style transfer, coined ST-VAE. It is a flexible framework that adapts to single or multiple style transfer. It consists of (1) an Image AutoEncoder for image reconstruction, where the style manipulation takes place in the feature space instead of the pixel space (Section 3.1); and (2) a Variational autoencoder-based Linear Transformation (VLT) that first learns the feature covariance for style and content images (Section 3.2.1) and then maps the covariance to a latent space via KL divergence (Section 3.2.2). Multiple style transfer is achieved as latent space-based linear interpolation. Experiments on the COCO dataset [11] and comparisons to modern methods show that ST-VAE achieves fine visual quality (Section 4). We also conduct a user study revealing that our results are superior to the state of the art.

Our contributions are: (1) we introduce ST-VAE, a novel method for single and multiple style transfer; (2) it casts style fusion as mixture models setting the path for future study; and (3) it outperforms all methods quantitatively and qualitatively.

2. RELATED WORK

Single image style transfer. Style transfer with DL was introduced in [10]. Since then, it has gained a lot of popularity and several works address it by using features to blend statistics from content and style images. AdaIN [3] proposes to match the mean and variance of intermediate features between content and reference images. Other works use higher-order statistic analysis [7, 12, 13, 14]. For instance, [12] explores the non-local feature correlations, whereas WCT [4] refines style transfer by directly embedding the Whitening and Coloring Transforms into a network. Recent works tackle it with Generative Adversarial Networks (GAN) [15, 16, 17, 18]. A representative work is CycleGAN [5] that uses two sets of GANs to form a mapping loop between domain A and B. [15] proposes to disentangle contents and styles from images so the stylization can be resolved in the style space, while [19] includes segmentation information for conditional editing.

Multiple style transfer is a less explored topic. The goal is to mix multiple styles and add them to the content image. Most approaches [1, 3, 8, 9, 12, 20] cast it as feature interpolation, i.e. interpolate between different styles and transfer the new style to the content images. [3] interpolates the means and variances of the style feature maps, whereas [21] adds the style interpolation into the loss and trains a model for different style mixture. However, these solutions are time-consuming and the range of mean and variance for different style images varies a lot, and hence simple interpolation does not guarantee the desired style mixture. Other works use GANs, e.g. [9] uses 1D codes as a condition for style transfer; however, the network is trained on a fixed number of styles, and hence it cannot transfer styles outside the training data. [8] proposes a GAN to train a spatial transformation matrix; however, the network performs poorly on styles outside the training data, and there are no multiple or convincing style transfer studies.

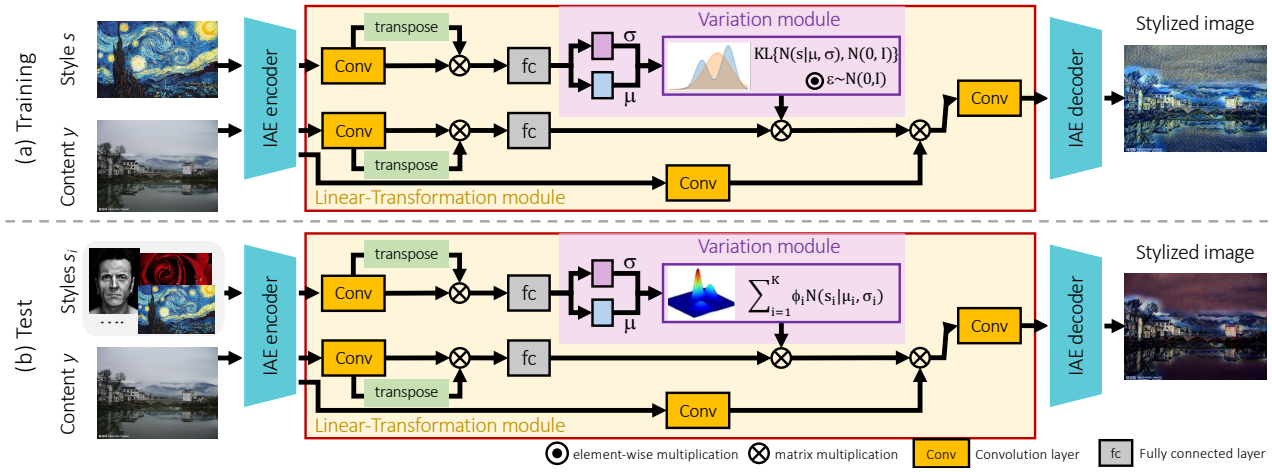

Fig. 1: ST-VAE. (a) At training, IAE is trained and then is kept frozen. After, VLT learns the covariance of style images via **linear-transformation module** and projects them to components of a multi-dimensional Gaussian distribution via KL minimization via the **variation module**. To transfer styles, it computes the covariance between style and content and multiplies it to the content features to obtain the new projected features. (b) At test time, given several input style images, ST-VAE samples styles in the latent space by model mixture (**magenta**).

3. METHOD

Here, we describe the Variational AutoEncoder for latent space-based Style Transfer (ST-VAE) that performs multiple style transfer by projecting nonlinear styles to a linear latent space, and it fuses them by linear interpolation (Figure 1). It consists of an Image AutoEncoder (IAE), i.e. an encoder-decoder performing image reconstruction (Section 3.1); and a Variational autoencoder-based Linear Transformation (VLT), responsible for latent space-based style manipulation (Section 3.2). The training comprises two phases (Figure 1(a)). First, IAE is trained for image reconstruction. Second, IAE is frozen and VLT learns the covariance of different styles via the **linear-transformation module** and then the **variation module** projects them to different components of a multi-dimensional Gaussian distribution via KL minimization. At test time, the style transfer is processed in the latent space via a mixture of Gaussian distributions (**magenta** in Figure 1(b)).

3.1. Image AutoEncoder (IAE)

IAE is a symmetric encoder-decoder that extracts features for image reconstruction. The encoder's structure follows VGG-19 [22] by keeping all conv layers and discarding the fc ones. The decoder is symmetric to the encoder and up-samples the feature abstraction to reconstruct the input image. Unlike [5, 23, 24], IAE has no short connections. IAE performs style transfer in the feature domain. The encoder learns a one-to-one mapping **E** from images **X** to features **M**, while the decoder learns a mapping **D** to reconstruct the images. These mappings ensure that each image corresponds to a unique and compact feature.

3.2. VAE-based Linear Transformation (VLT)

VLT is built upon a Variational AutoEncoder and is responsible for latent space based style manipulation. It consists of a **linear transformation** (Section 3.2.1), which learns the covariance matrices of the content and style features; and a **variation module** (Section 3.2.2), which projects the styles to a normal Gaussian space. After, the content and style covariance matrices are multiplied for stylization.

Notation. Let $F_c, F_s \in \mathbb{R}^{C\times N}$ be the vectorized feature maps of content and style image, respectively that are obtained at the top-most encoder layer. N is the feature length for content and style image and C is the number of channels.

3.2.1. *Linear Transformation module*

To perform content and style transfer, we deploy a two-fold objective. First, we find a linear transformation $T \in \mathbb{R}^{C\times C}$ to transfer content F_c to the desired feature maps F_d, such that $F_d = TF_c$. Second, we find a nonlinear mapping model ϕ, such that $\phi_s = \phi(F_s)$, where ϕ_s is the transformed style feature maps. This is in line with [4, 13] that cast the stylization as a covariance matching process. Our objective is:

$$L_{style} = \frac{1}{NC}||\bar{F}_d\bar{F}_d^T - \bar{\phi}_s\bar{\phi}_s^T||^l \\ s.t. \bar{F}_c = F_c - mean(F_c), \bar{F}_d = T\bar{F}_c. \quad (1)$$

Equation (1) describes the l-th order minimization, where $\bar{F}_d$ is the desired feature vector. $\bar{F}_d\bar{F}_d^T$ is the covariance of $\bar{F}_d$. Following [4], we find the covariance of $\bar{F}_d$, with the whitening process, i.e. singular value decomposition. We find the eigenvector matrix E and the diagonal eigenvalue matrix D as: $cov(\phi_s) = \bar{\phi}_s\bar{\phi}_s^T = E_sD_sE_s^T$, and hence we estimate

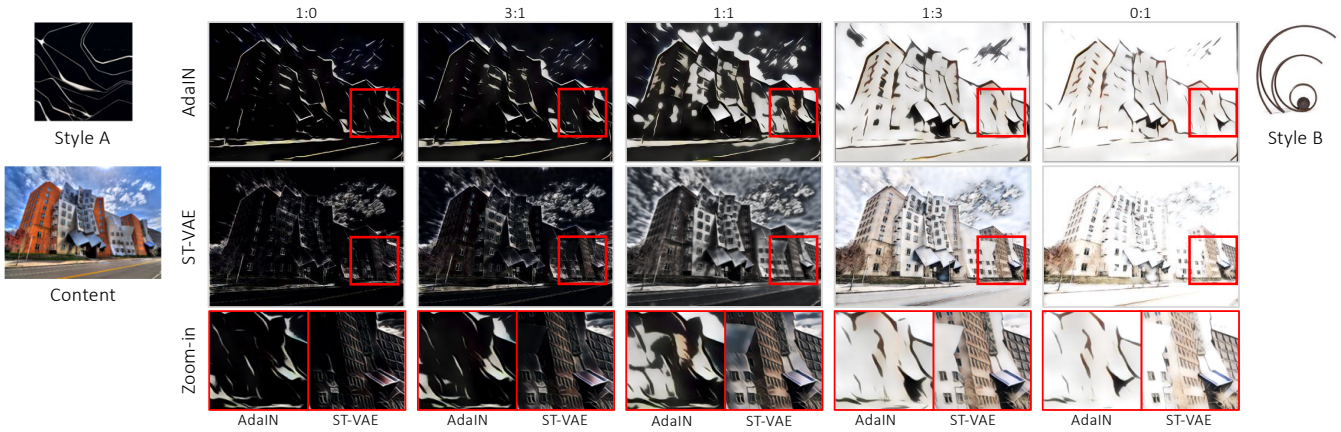

Fig. 2: Image synthesis using two style images for the same content image for AdaIN [3] and ST-VAE (ours). The weights of the styles are shown above each stylized image. ST-VAE transitions between styles smoother than AdaIN, especially on the building edges and windows.

the covariance as: $cov(\bar{F}_d) = T\bar{F}_c\bar{F}_c^T T^T = TE_cD_cE_c^T T^T$. Therefore, the generalized transformation is:

$$\begin{aligned} TE_cD_cE_c^T T^T &= E_sD_sE_s^T \\ T &= (E_sD_sE_s^T)\Sigma(E_cD_cE_c^T) \\ T &= T_1(E_sD_sE_s^T) \times T_2(E_cD_cE_c^T), \end{aligned} \tag{2}$$

where $\Sigma \in \mathbb{R}^{C\times C}$ is C-dimensional orthogonal matrix. Equation (2) shows that the transformation T is determined by the covariance of the content and style image features. Once T is calculated, F_d is obtained by $T\bar{F}_c + mean(F_s)$, which aligns it to the mean and covariance of the style image.

3.2.2. *Variation module*

The **variation module** is a projection model responsible for multiple style transfer. Only a few works explore controllable or weighted style transfer [1, 3, 25] by manually adjusting the weights for linear interpolation, but they cause inconsistent style transition as the features are nonlinear. Instead, we embed the variation module into the transformation model to map features into a linear space spanned by mixture models.

VAE is defined as: $P(X)$=$\int P(\mathbf{X}, z)P(z|\mathbf{X})\, dz$, where $\mathbf{X}$ is the input and z is sampled from the latent space $\mathbf{Z}$ [24, 26, 27]. To regularize the latent space, we use the Kullback–Leibler (KL) divergence that measures the probability close to a normal distribution. The variation module learns parameters θ for maximizing the data log likelihood $P_\theta(\mathbf{X})$:

$$logP_\theta(\mathbf{X}) = E_{Q_\omega(z|X)}[logP_\theta(\mathbf{X}, z)] - KL[Q_\omega(z|\mathbf{X})||P_\theta(z|\mathbf{X})]. \tag{3}$$

Equation (3) shows that the encoder learns parameters ω to approximate posterior $Q_\omega(z|\mathbf{X})$, while the decoder learns θ to represent the likelihood $P_\theta(\mathbf{X}, z)$. The real prior distribution $P_\theta(z|\mathbf{X})$ is a Gaussian distribution and the approximated posterior follows $z \sim Q_\omega(z|x_i) = N(z; \mu_i, \sigma_i^2 I)$. We use the variation module to cast the multiple style transfer problem as a generative sampling process: by projecting arbitrary style images to a hidden distribution, each style image corresponds to one sample on the latent space. Thus, the multiple style transfer becomes data interpolation in the latent space, performed by a multivariate Gaussian mixture model.

Training Loss. We train our model using the style L_{style} and content losses L_{content} and KL divergence as follows:

$$\begin{aligned} L_{\text{content}} &= \mathbb{E}||\psi(T(\mathbf{X})) - \psi(\mathbf{X})||_1, \text{and} \\ L_{\text{VLT}} &= L_{\text{content}} + \lambda L_{\text{style}} + \beta KL[Q_\phi(z|\mathbf{X})||N(0,1)], \end{aligned} \tag{4}$$

where λ and β are the weighting parameters to balance style and KL losses, ψ_i is the i-th feature maps extracted from the pre-trained VGG-16 model [22]. Recall that we use VGG-16 to compute losses, while we modify VGG-19 for IAE.

4. EXPERIMENTS

We report results on single and multiple style transfer, runtime and quantitative evaluations (more examples in sup. material).

Implementation details. For content loss, we use *relu4_1* to compute differences between content and stylized images. For perceptual loss, we combine first- and second-order statistics to measure the similarity between reference and stylized images. We train IAE on COCO [11]. For VLT, we use COCO as content and WikiArt [6] as style. At training, we keep the ratio and crop a region of 256×256 as patches. For augmentation, we randomly flip and rotate the contents. We train with Adam optimizer, a learning rate of 10^{-4} and batch size of 8 for 100k iterations (8h, NVIDIA GTX1080Ti GPU).

Discussion. ST-VAE transfers textures and styles efficiently (appr. 100 fps). For efficiency, the input features are first compressed for transformation learning and then uncompressed to match the original dimension ('conv' in Figure 1). Instead of

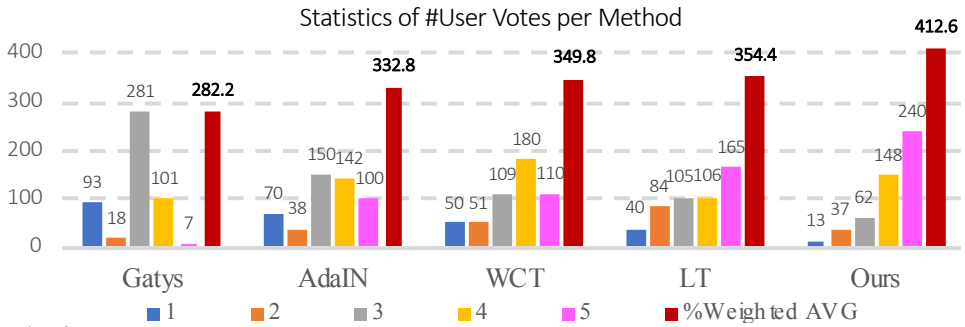

Fig. 3: Statistics of % user scores on style transfer. Different colors represent different scores from 1 to 5 (5 being the highest). 20 users were asked to rate synthesized images. Overall, we collect 500 votes for each method and observe that images from ST-VAE are unanimously rated the best.

Image size	256× 256	512× 512	1024× 1024
Gatys [10]	16.51	59.45	251.44
AdaIN [3]	0.019	0.071	0.288
WCT [4]	0.922	1.080	4.001
LT [13]	0.01	0.036	0.146
Ours	0.01	0.041	0.145

Table 1: Running time comparison on different image resolutions, measured in seconds using the original source code on a GTX 1080Ti GPU. Red indicates the best results, blue indicates the second best results.

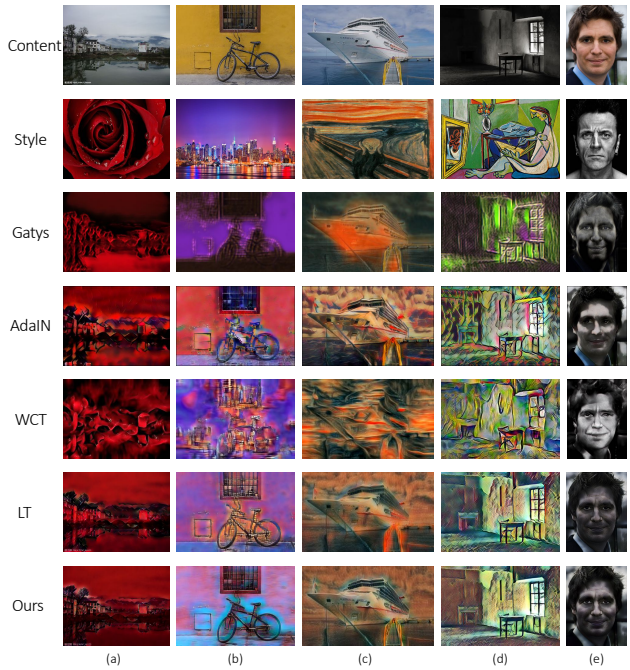

Fig. 4: Results of seven style transfer methods. ST-VAE has more similar affinity with the content than other methods, e.g. it preserves the objects in (b,d), while others fail to produce detailed structures.

fixed-size content and style [10, 12], ST-VAE does not depend on the image resolution, thus handling arbitrary style transfer.

General style transfer. ST-VAE performs robust style transfer without affecting the structure of the content images. To show its effectiveness, we compare it to the state of the art: Gatys [10], AdaIN [3], WCT [4], and LT [13]. Figure 4 shows five content and style images and the results with all methods. For a fair evaluation, we choose content images from [12] and [28] that are not part of our training set. ST-VAE successfully transfers the desired styles and textures and preserves the details of the content better. For instance, in Figure 4(a) ST-VAE clearly reconstructs the windows and doors.

Multiple style transfer. Figure 2 shows the style mixture results when using two style images for ST-VAE and AdaIN [3]. Note, the style interpolation is done by assigning different weights to the style images. We focus on the texture transition. We observe that ST-VAE successfully preserves the content information better than AdaIN while transitioning between styles in a smoother way. For instance, it successfully preserves the clouds and windows while transferring styles, while AdaIN losses these details. Furthermore, ST-VAE results in smooth changes between foreground and background without any boundary effects, while AdaIN fails.

Quantitative evaluation. To evaluate ST-VAE, we conduct a user study, where users are presented with 5 synthetic images in random order and are asked to rate their quality from 1 to 5 (5 the highest). We use five methods: Gatys, AdaIN, WCT, LT and ours. For each method, we synthesize 400 images (10 content, 40 styles) and randomly select 20. We collect 500 votes from 20 users and report the results in Figure 3, where we observe that ST-VAE is favoured amongst all.

Computational cost. Table 1 reports the run-times on style transfer with different resolutions. In most cases, ST-VAE leads to the lowest run-time; for instance, for low-resolution images it generates images within 0.01s.

5. CONCLUSION

We introduced ST-VAE, a Variational AutoEncoder based style transfer method that maps features into a multivariate Gaussian distribution for both single and multiple style transfer with more consistent style transitions. The linear transformation enables feed-forward training and testing, thus making ST-VAE very efficient. Our experiments show that ST-VAE performs favourably against the state of the art, both quantitatively and qualitatively. Future work involves extending it to videos by exploiting the temporal continuity of frames [29].

Acknowledgements. This work was partly funded by the Google chair at École Polytechnique.

6. REFERENCES

[1] Aysegul Dundar, Ming-Yu Liu, Ting-Chun Wang, John Zedlewski, and Jan Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," *CoRR*, vol. abs/1807.09384, 2018. 1, 3

[2] Leon A Gatys, Alexander S Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman, "Controlling perceptual factors in neural style transfer," in *Proc. CVPR*, 2017. 1

[3] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. ICCV*, 2017. 1, 3, 4

[4] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang, "Universal style transfer via feature transforms," in *NeurIPS*, 2017. 1, 2, 4

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. ICCV*, 2017. 1, 2

[6] Kiri Nichol, "Painter by numbers, wikiart," https://www.kaggle.com/c/painter-by-numbers, 2016. 1, 3

[7] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz, "A closed-form solution to photorealistic image stylization," in *Proc. ECCV*, 2018. 1

[8] Hang Zhang and Kristin Dana, "Multi-style generative network for real-time transfer," in *Proc. ECCV-Workshops*, 2018. 1

[9] Keiji Yanai and Ryosuke Tanno, "Conditional fast style transfer network," in *ACM ICMR*, 2017. 1

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in *Proc. CVPR*, 2016. 1, 4

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *Proc. ECCV*, 2014. 1, 3

[12] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala, "Deep photo style transfer," in *Proc. CVPR*, 2017. 1, 4

[13] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang, "Learning linear transformations for fast arbitrary style transfer," in *Proc. CVPR*, 2019. 1, 2, 4

[14] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang, "Diversified texture synthesis with feed-forward networks," in *Proc. CVPR*, 2017. 1

[15] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. ECCV*, 2018. 1

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila, "Analyzing and improving the image quality of stylegan," in *Proc. CVPR*, 2020. 1

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Proc. ICLR*, 2018. 1

[18] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto, "Adversarial latent autoencoders," in *Proc. CVPR*, 2020. 1

[19] Ziad Al-Halah and Kristen Grauman, "From paris to berlin: Discovering fashion style influences around the world," in *Proc. CVPR*, 2020. 1

[20] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, ICML'16, p. 1349–1357. 1

[21] Michael C. Maring and Kaustav Chakraborty, "Multiple style-transfer in real-time," *ArXiv*, 2019. 1

[22] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015. 2, 3

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015. 2

[24] Zhi-Song Liu, Wan-Chi Siu, and Yui-Lam Chan, "Reference based face super-resolution," *IEEE Access*, 2019. 2, 3

[25] Minxuan Lin, Fan Tang, Weiming Dong, Xiao Li, Chongyang Ma, and Changsheng Xu, "Distribution aligned multimodal and multi-domain image stylization," *arXiv*, 2020. 3

[26] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," in *Proc. ICLR*, 2014. 3

[27] Zhi-Song Liu, Wan-Chi Siu, Li-Wen Wang, Chu-Tak Li, and Marie-Paule Cani, "Unsupervised real image super-resolution via generative variational autoencoder," in *Proc. CVPR-Workshops*, 2020. 3

[28] Zhi-Song Liu, Li-Wen Wang, Chu-Tak Li, Wan-Chi Siu, and Yui-Lam Chan, "Image super-resolution via attention based back projection networks," in *Proc. ICCV-Workshops*, 2019. 4

[29] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid, "Action tubelet detector for spatio-temporal action localization," in *Proc. ICCV*, 2017. 4