

PCA-Based Knowledge Distillation Towards Lightweight and Content-Style Balanced Photorealistic Style Transfer Models

Tai-Yin Chiu
 University of Texas at Austin
 chiu.taiyin@utexas.edu

Danna Gurari
 University of Colorado Boulder
 danna.gurari@colorado.edu

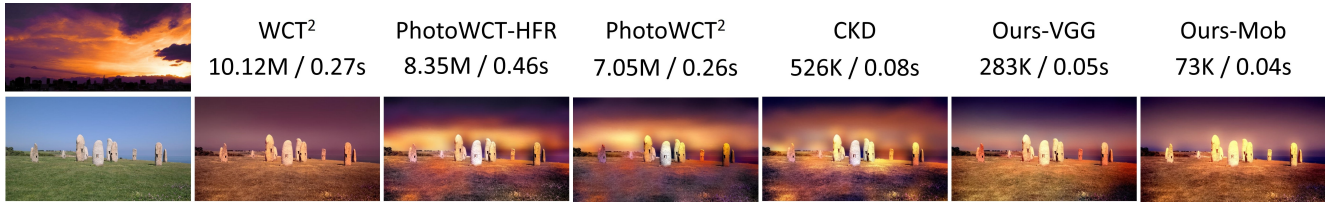


Figure 1. Exemplification of our PCA-based knowledge distillation resulting in models that are more lightweight, faster, and achieve a better content-style balance for photorealistic style transfer than existing models. We demonstrate its versatility by reporting results for when our approach is applied to two backbone architectures: VGG (Ours-VGG) and MobileNet (Ours-Mob). Compared to WCT², our smallest model uses only 0.7% of the parameters (73K vs. 10.12M) while producing results over 6x faster (0.04s vs. 0.27s). Additionally, our distilled models consistently transfer stronger style effects than WCT² [48] and preserve content better than PhotoWCT-HFR (an improved variant of PhotoWCT [24]), PhotoWCT² [11], and a CKD [43] distilled model.

Abstract

Photorealistic style transfer entails transferring the style of a reference image to another image so the result seems like a plausible photo. Our work is inspired by the observation that existing models are slow due to their large sizes. We introduce PCA-based knowledge distillation to distill lightweight models and show it is motivated by theory. To our knowledge, this is the first knowledge distillation method for photorealistic style transfer. Our experiments demonstrate its versatility for use with different backbone architectures, VGG and MobileNet, across six image resolutions. Compared to existing models, our top-performing model runs at speeds 5-20x faster using at most 1% of the parameters. Additionally, our distilled models achieve a better balance between stylization strength and content preservation than existing models. To support reproducing our method and models, we share the code at <https://github.com/chiutaiyin/PCA-Knowledge-Distillation>.

1. Introduction

Photorealistic style transfer is the task of rendering an image (content image) in the style of a reference image (style image) to create a photorealistic result. Examples are shown in Fig. 1. A key challenge the community has focused on tackling since the seminal neural network-based

algorithm for this task [30] has been how to simultaneously achieve a good balance between stylization strength and content preservation while running fast to better support practical applications [11, 24, 46, 48].

The status quo for modern photorealistic style transfer models is to use autoencoders. As illustrated in Fig. 2(a), the basic model uses a pre-trained VGG-19 [40]¹ as the encoder to extract content and style features, then a feature transformation to adapt the content feature with respect to the style feature, and finally a decoder to invert the adapted feature to a stylized image. A limitation of this framework is that the speed of autoencoder-based approaches is limited by the large size of the VGG-19 backbone.

This size limitation is further amplified in state-of-the-art models, as they extend the basic autoencoder framework by using multiple feature transformations to better capture style effects. For instance, PhotoWCT [24] and PhotoWCT² [11] perform coarse-to-fine feature transformations to sequentially adapt the coarse content feature (i.e., *relu4_1* content feature from VGG) to the fine content feature (i.e., *relu1_1* content feature from VGG) with respect to the corresponding style features. This means the fine style feature is added on top of the coarse style feature and so can produce strong style effects. However, as

¹VGG-19 is favored in artistic [7, 15, 16, 19, 20, 22, 23, 39] and photorealistic [4, 11, 24, 30, 46, 48] style transfer research due to its simple architecture with no complex multiple branches and residual modules, making VGG features easier to interpret for content and style.

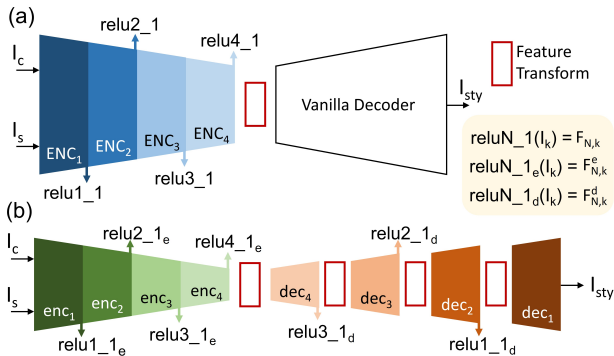


Figure 2. (a) The basic autoencoder framework for photorealistic style transfer. It uses the encoder ENC (VGG-19 from the input to $relu4_1$ layer here) to extract features from the content and style images I_c and I_s , applies a feature transformation (usually ZCA feature transformation [23]) to adapt the content feature $F_{4,c}$ with respect to the style feature $F_{4,s}$, and decodes the adapted feature to a stylized image I_{sty} . (b) The $enc-dec$ autoencoder resulting from our method. enc is distilled from ENC and follows its structure but uses a smaller channel length at each layer. The $reluN_1e$ layer in enc is the counterpart layer of $reluN_1$. If without feature transformations, the feature $F_{N,k}^d$ of an input image I_k from the $reluN_1d$ layer in dec reproduces the feature $F_{N,k}^e$ generated at $reluN_1e$. When performing stylization with feature transformations, our model sequentially adapts the coarse $relu4_1e$ content feature to the fine $relu1_1d$ content feature.

shown in Fig. 1, the strong style effects may in turn introduce artifacts which ruin the content. Other models such as WCT^2 [48] and PhotoNAS [4] perform fine-to-coarse feature transformations to adapt the fine content feature first and the coarse later to transfer weaker style effects and avoid artifacts. However, as shown in Fig. 1, the style effects may be poorly captured since adding the coarse style later partially overshadows the former fine style. Altogether, we observe that the state-of-the-art photorealistic style transfer models not only suffer from large sizes and so slow speeds, but also poor balance between content preservation and stylization strength.

To address these issues, (1) we propose a *PCA-based knowledge distillation*, which we motivate from PCA theory [3], to distill the most important knowledge for style representation from a source model to a *smaller encoder*. To demonstrate the generalizability of our distillation method, we apply it to two backbone architectures as the source models: VGG-19 and MobileNet [18]. (2) We integrate the blockwise decoder training of PhotoWCT² [11] into our PCA-based knowledge distillation to concurrently implement a *smaller pairing decoder* which reproduces the encoder features to perform coarse-to-fine feature transformations. The resulting model is shown in Fig. 2(b). Experiments demonstrate that our smaller models reflect bet-

ter style than WCT^2 and PhotoNAS due to coarse-to-fine feature transformations, preserve better content than PhotoWCT and PhotoWCT² due to knowledge distillation, and incur faster speeds. These benefits are exemplified in Fig. 1. To the best of our knowledge, our method is the first knowledge distillation algorithm for photorealistic style transfer.

2. Related Works

Modern photorealistic style transfer models. To overcome the slow speed of the pioneering neural network-based algorithm [30], most modern photorealistic style transfer models are based on autoencoders [10, 11, 24, 48].² Yet, as explained in Sec. 1, these models are still relatively slow due to their large sizes. Additionally, such models result in a poor balance between stylization strength and content preservation. For instance, PhotoWCT [24] realizes strong stylization strength by coarse-to-fine feature transformations, but suffers poor content preservation for two reasons: artifacts from strong stylization strength and missing high-frequency details due to its lossy structure. WCT^2 [48] improves the content preservation from PhotoWCT by performing fine-to-coarse feature transformations to transfer weaker style effects in order to reduce artifacts and also introduces wavelet-based skip connections to reinforce the high-frequency component construction. However, sacrificing stylization strength means WCT^2 weakly captures the style. PhotoWCT² [11] combines the merits of PhotoWCT and WCT^2 : it introduces high-frequency residuals to improve the high-frequency detail construction from PhotoWCT and realizes coarse-to-fine feature transformations to maintain strong stylization strength. Yet, PhotoWCT² still can generate artifacts, likely because of the coarse-to-fine feature transformations. Our experiments demonstrate that models distilled with our PCA knowledge distillation simultaneously address the aforementioned limitations of existing models: our models are smaller, faster, and achieve a better content-style balance.

Knowledge distillation for vision tasks. Given a neural network (teacher/source model) for a specific task, knowledge distillation (KD) aims to efficiently train a smaller network (student/target model) for the same task by leveraging the knowledge acquired in the source network.

Most KD algorithms address high-level vision tasks. For example, following the pioneering KD algorithm [17], most follow-up algorithms address image classification [12, 27, 31, 32, 34–36, 42, 44, 47, 49]. Others address object detection [6], semantic segmentation [28] and clothing matching [29]. Yet, these algorithms are not applicable to the

²An exception is a 2020 method [46] which learns to compute a pixel-wise linear mapping to map the content pixel values to the stylized pixel values. However, its code is not available.

low-level photorealistic style transfer task. For example, it is sufficient for prior work that the coarse features of two dog images from a late layer of a distilled image classifier to be similar such that the final classification layer can classify them to the same class (i.e. dog class). In contrast, photorealistic style transfer should preserve image content and so it requires that the features differ so that the different content in the dog images can be reconstructed.

The closest work to ours is collaborative knowledge distillation (CKD) [43] for artistic style transfer.³ However, CKD is different from ours in several respects. First, CKD is empirical without theoretical explanation. Second, with no theory behind, there is no guideline for target model size selection. Third, CKD is poor at preserving content and so is not applicable to photorealistic style transfer. We experimentally demonstrate the advantages of our PCA-based approach over CKD.

3. Method

We now describe our PCA-based algorithm for distilling lightweight, fast, high quality models for photorealistic style transfer. For simplicity, we describe it when used with the most popular backbone architecture for this task: VGG-19. However, to show the generalizability of our approach, we will also apply the algorithm to distill a model from a MobileNet-based source model in the experiments.

3.1. Background: how to represent style

It has been shown that a style representation is valid (i.e., can catch the style of an image) if the distance between style representations of the style image and the stylized image is a Maximum Mean Discrepancy [25]. Under this scheme, the covariance of an image feature \mathbf{F} extracted at an intermediate layer of VGG-19 is one feasible style representation. Formally, by reshaping $\mathbf{F} \in C \times H \times W$ (C, H, W : channel length, height, and width of \mathbf{F}) to a 2D matrix of shape $C \times HW$, the covariance is calculated as $\frac{1}{HW} \bar{\mathbf{F}} \bar{\mathbf{F}}^T$, where $\bar{\mathbf{F}}$ is the centralized feature $\mathbf{F} - \mu(\mathbf{F})$, and $\mu(\mathbf{F})$ is the mean of HW column vectors of \mathbf{F} . We leverage this style representation in our method. For brevity, in what follows, we exclude the coefficient of the covariance matrix.

³There are three prior works indirectly related to ours: (1) optical flow distillation for video artistic style transfer [8], (2) knowledge transfer via PCA and graph neural networks [21], and (3) PCA-based distilled dense neural network [38]. In (1), only the optical flow information is distilled to the target model, not the style information. The target model learns the style knowledge from a given style image and so is only able to transfer that particular style. In (2), PCA is not used for knowledge distillation but rather visualization and reduction of computational complexity in graph neural networks. In (3), the source and the target models can only be networks consisting of dense layers and so not convolutional layers. Unlike our method, which treats a feature of an image as a collection of vectors and applies PCA to the vectors, (3) needs to acquire the dense features of all images in a dataset and then applies PCA to them. Consequently, (3) is limited to small network structures and small datasets.

Often multiple layers of a network (e.g. $reluN-1$, $N = 1, 2, 3, 4$) are simultaneously used in order to extract a style representation that captures both the coarse and fine style of an image [15, 25]. We follow this approach in our work.

3.2. PCA-based knowledge distillation

Our PCA-based knowledge distillation algorithm consists of two parts, which we describe below: *global eigenbasis derivation* and *blockwise PCA knowledge distillation*.

Teacher/Source model. We consider VGG-19 from its input to the $relu4-1$ layer as the source model which we name ENC . From ENC we distill the style knowledge to a smaller target model which we name enc , whose structure follows the source model but uses a smaller channel length at each layer. More specifically, the style knowledge will be distilled from the $reluN-1$ ($N = 1, 2, 3, 4$) layers of ENC to the counterpart layers in enc which we call $reluN-1_e$ layers.

In the following, we let $\mathbf{F}_{N,k} \in C_N \times H_{N,k} W_{N,k}$ be the feature of the image I_k extracted at the $reluN-1$ layer of ENC and $\mathbf{F}_{N,k}^e \in C_N^e \times H_{N,k} W_{N,k}$ ($C_N^e \ll C_N$) be the feature extracted at the counterpart $reluN-1_e$ layer of enc . The above notations are summarized in Fig. 2.

PCA for style knowledge distillation: image-dependent eigenbases. We represent the style of a photograph I_k using the covariance matrices of the features from the source/target model. Formally, in the distillation from $reluN-1$ to $reluN-1_e$, we want the covariance of the distilled feature $\bar{\mathbf{F}}_{N,k}^e (\bar{\mathbf{F}}_{N,k}^e)^T$ to capture the most important information in the covariance of the source feature $\bar{\mathbf{F}}_{N,k} \bar{\mathbf{F}}_{N,k}^T$ needed for stylization. Moreover, we want to do this distillation for multiple layers: distill $reluN-1$ layer to the $reluN-1_e$ layer for $N = 1, 2, 3, 4$.

This kind of problem is classical dimension reduction with PCA. In PCA, we treat $\bar{\mathbf{F}}_{N,k}$ as a collection of $H_{N,k} W_{N,k}$ numbers of C_N -dimensional data points. PCA seeks an orthonormal transformation $\mathbf{W}_{N,k} \in \mathbb{R}^{C_N \times C_N}$ to map these data points to a C_N^e -dimensional space, so that the covariance $\mathbf{W}_{N,k} \bar{\mathbf{F}}_{N,k} \bar{\mathbf{F}}_{N,k}^T \mathbf{W}_{N,k}^T$ of the mapped feature $\mathbf{W}_{N,k} \bar{\mathbf{F}}_{N,k}$ preserves the essential information in the original covariance $\bar{\mathbf{F}}_{N,k} \bar{\mathbf{F}}_{N,k}^T$. Mathematically, PCA solves the following optimization problem [2, 3].

$$\max_{\mathbf{W}_{N,k} \mathbf{W}_{N,k}^T = \mathbb{1}} \text{tr}(\mathbf{W}_{N,k} \bar{\mathbf{F}}_{N,k} \bar{\mathbf{F}}_{N,k}^T \mathbf{W}_{N,k}^T), \quad (1)$$

where the constraint $\mathbf{W}_{N,k} \mathbf{W}_{N,k}^T = \mathbb{1}$ is the orthonormality property of $\mathbf{W}_{N,k}$.

Intuitively, we can think of the trace function as a metric to evaluate the information in a covariance matrix and we search an orthonormal basis to maximize the metric value. According to PCA, the C_N^e row vectors of the solution $\mathbf{W}_{N,k}$ to Eq. (1) are the C_N^e leading eigenvectors of the covariance $\bar{\mathbf{F}}_{N,k} \bar{\mathbf{F}}_{N,k}^T$. With $\mathbf{W}_{N,k}$, we train the target

model *enc* to produce at the *reluN-I_e* layer a feature $\mathbf{F}_{N,k}^e$ whose centralized form $\bar{\mathbf{F}}_{N,k}^e$ is $\mathbf{W}_{N,k}\bar{\mathbf{F}}_{N,k}$.

However, the image-dependent transformations $\mathbf{W}_{N,k}$'s for distillation are problematic, which can be mathematically explained as follows. First, the *reluN-I_e* layer of the distilled model *enc* defines a C_N^e -dimensional space $\mathcal{S}_{C_N^e}$:

$$\mathcal{S}_{C_N^e} = \{f \in \mathbb{R}^{C_N^e} \mid f \text{ is a column vector of } \bar{\mathbf{F}}_{N,k}^e, \forall I_k\}. \quad (2)$$

Therefore, for any two images I_{k_1} and I_{k_2} , the column vectors of $\bar{\mathbf{F}}_{N,k_1}^e$ and $\bar{\mathbf{F}}_{N,k_2}^e$ should be in the same space $\mathcal{S}_{C_N^e}$. However, the orthonormal transformations \mathbf{W}_{N,k_1} and \mathbf{W}_{N,k_2} calculated from the covariances $\bar{\mathbf{F}}_{N,k_1}\bar{\mathbf{F}}_{N,k_1}^T$ and $\bar{\mathbf{F}}_{N,k_2}\bar{\mathbf{F}}_{N,k_2}^T$ may be different, so the column vectors of the mapped feature $\mathbf{W}_{N,k_1}\bar{\mathbf{F}}_{N,k_1}$ may reside in a different C_N^e -dimensional space than the column vectors of $\mathbf{W}_{N,k_2}\bar{\mathbf{F}}_{N,k_2}$, which conflicts with the former argument. Thus, if we distill with the *image-dependent eigenbases* $\mathbf{W}_{N,k}$'s, it results in a *sub-optimal space that does not catch the style of any images well*. We will show this in Sec. 4.3.

PCA for style knowledge distillation: global eigenbases (i.e., image-independent). We instead introduce a global, image-independent eigenbasis $\mathbf{W}_{N,g} \in \mathbb{R}^{C_N^e \times C_N}$. This implementation defines a unique C_N^e -dimensional space that on average catches the style of different images well.

Our solution entails modifying Eq. (1) towards solving the following optimization problem for $\mathbf{W}_{N,g}$:

$$\max_{\mathbf{W}_{N,g}, \mathbf{W}_{N,g}^T = \mathbf{1}} \frac{1}{M} \sum_{k=1}^M \text{tr}(\mathbf{W}_{N,g}\bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T\mathbf{W}_{N,g}^T), \quad (3)$$

where M represents the number of photographs in a virtually infinitely large image dataset, and the solution of $\mathbf{W}_{N,g}$ is the eigenbasis of $\frac{1}{M} \sum_{k=1}^M \bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T$, which is not analytically attainable. While Eq. (3) seems approximately solvable using mini-batch gradient descent to minimize the loss $-\sum_{I_k \in \mathcal{B}_t} \text{tr}(\mathbf{W}_{N,g}\bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T\mathbf{W}_{N,g}^T)/|\mathcal{B}_t|$, where \mathcal{B}_t is a batch of sampled images at the t -th iteration of gradient descent, the minus of a trace as a loss function makes the gradient descent process unstable. This is because such a loss has no lower bound and so a gradient descent algorithm is prone to keep minimizing the loss but ignore the constraint $\mathbf{W}_{N,g}\mathbf{W}_{N,g}^T = \mathbf{1}$.

To bypass it, we rewrite Eq. (3) as the equivalent form⁴:

$$\min_{\mathbf{W}_{N,g}, \mathbf{W}_{N,g}^T = \mathbf{1}} \frac{1}{M} \sum_{k=1}^M \|\mathbf{W}_{N,g}^T\mathbf{W}_{N,g}\bar{\mathbf{F}}_{N,k} - \bar{\mathbf{F}}_{N,k}\|_2^2, \quad (4)$$

where $\mathbf{W}_{N,g}^T\mathbf{W}_{N,g}\bar{\mathbf{F}}_{N,k}$ is the reconstructed feature of $\bar{\mathbf{F}}_{N,k}$ from the mapped feature $\mathbf{W}_{N,g}\bar{\mathbf{F}}_{N,k}$. The meaning behind the equivalence is that the orthonormal basis $\mathbf{W}_{N,g}$

⁴Derivation of the equivalence is in the Supplementary Materials.

that on average *maximizes the covariance* information in the C_N^e -dimensional space should also on average *minimize the reconstruction loss* in the original C_N -dimensional space. Unlike the trace function in Eq. (3) which is unbounded, the L2-norm of Eq. (4) has a lower bound of zero and so Eq. (4) is approximately solvable with mini-batch gradient descent. Moreover, since $\mathbf{W}_{N,g}$'s ($N = 1, 2, 3, 4$ for *reluN-I* layers) are independent, they can be solved together with the same batch of sampled images at an iteration of gradient descent.

To summarize, to derive $\mathbf{W}_{N,g}$'s ($N = 1, 2, 3, 4$), we use mini-batch gradient descent to solve Eq. (4) for four $\mathbf{W}_{N,g}$'s at once: At the t -th iteration of gradient descent, we sample a batch \mathcal{B}_t for the following minimization problem:

$$\min_{\substack{\mathbf{W}_{N,g}, \mathbf{W}_{N,g}^T = \mathbf{1} \\ N \in \{1,2,3,4\}}} \frac{1}{|\mathcal{B}_t|} \sum_{N=1}^4 \sum_{I_k \in \mathcal{B}_t} \|\mathbf{W}_{N,g}^T\mathbf{W}_{N,g}\bar{\mathbf{F}}_{N,k} - \bar{\mathbf{F}}_{N,k}\|_2^2 \quad (5)$$

and calculate the gradient of the objective in Eq. (5) to update $\mathbf{W}_{N,g}$'s. In our implementation, we use the MS-COCO dataset with random crop data augmentation, a batch size $|\mathcal{B}_t|$ of 8, and train $\mathbf{W}_{N,g}$'s for five epochs. Fig. 3(a) exemplifies the derivation with a batch of one image.

With global eigenbases $\mathbf{W}_{N,g}$'s ($N = 1, 2, 3, 4$), we can distill style information from *reluN-I* layers of the source model *ENC* to *reluN-I_e* layers of the target model *enc*.

Blockwise PCA knowledge distillation. To realize coarse-to-fine feature transformations in a distilled model, in addition to distilling the style information to the distilled encoder *enc*, our approach also must implement a pairing decoder *dec* to reproduce encoder features. We now describe our distillation approach, which integrates PCA knowledge distillation with the blockwise training strategy for the decoder in PhotoWCT² [11] to achieve this.

A schematic diagram of our *enc-dec* model is shown in Fig. 2. We split the encoder *enc* into a series of blocks $\{enc_1, enc_2, enc_3, enc_4\}$ where the output of *enc_N* is the *reluN-I_e* layer, and the decoder *dec* into a series of blocks $\{dec_4, dec_3, dec_2, dec_1\}$ where the output of *dec_N* is the *relu(N-1)-I_d* layer which reproduces the *relu(N-1)-I_e* feature. That is, the decoder takes as input the *relu4-I_e* feature from *enc* to progressively reproduce *relu3-I_e*, *relu2-I_e*, and *relu1-I_e* features and the reconstructed image. During stylization, ZCA feature transformations [9, 23] are placed at the *relu4-I_e* layer and the *reluN-I_d* layers ($N = 3, 2, 1$) to sequentially adapt the coarse *relu4-I_e* content feature to the fine *relu1-I_d* content feature with respect to the corresponding style features.

To implement our *enc-dec* model, each pair of *enc_N* and *dec_N* are trained together with other pairs fixed by minimizing the encoder distillation loss \mathcal{L}_{enc}^N and the decoder implementation loss \mathcal{L}_{dec}^N . The four pairs are trained sequentially

(a) Global eigenbasis derivation

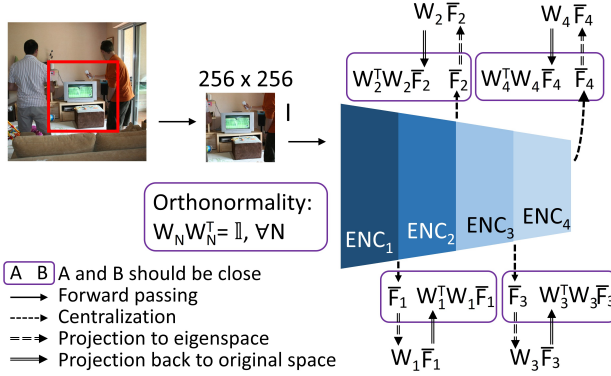
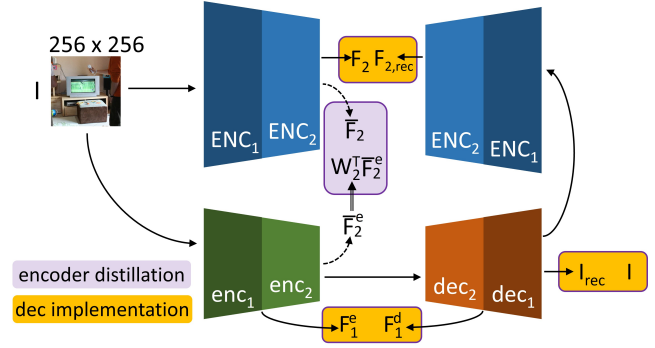
(b) Blockwise PCA knowledge distillation (exemplified with the training of enc_2 and dec_2)

Figure 3. PCA-based knowledge distillation for photorealistic style transfer consists of two steps: (a) global eigenbases (\mathbf{W}_N , $N = 1, 2, 3, 4$) derivation and (b) blockwise PCA knowledge distillation. As explained for Eq. (4), the \mathbf{W}_N that on average maximizes the style information in the distilled space also minimizes the feature reconstruction loss $\|\bar{\mathbf{F}}_N - \mathbf{W}_N^T \mathbf{W}_N \bar{\mathbf{F}}_N\|_2^2$. With \mathbf{W}_N 's, style knowledge is blockwisely distilled from the source model block ENC_N to the target model block enc_N in the order from $N = 1$ to $N = 4$ by minimizing the encoder distillation loss (Eq. (6)) and the decoder implementation loss (Eq. (7)).

in the order from $N = 1$ to $N = 4$. We exemplify training the pair of enc_2 and dec_2 in Fig. 3(b).

In the encoder distillation, given the image I_k we want to train the encoder block enc_N to make its centralized output $\bar{\mathbf{F}}_{N,k}^e$ close to the feature $\mathbf{W}_{N,g} \bar{\mathbf{F}}_{N,k}$, which is mapped from the centralized output $\bar{\mathbf{F}}_{N,k}$ of ENC_N by the global eigenbasis $\mathbf{W}_{N,g}$. Instead of taking $\|\bar{\mathbf{F}}_{N,k}^e - \mathbf{W}_{N,g} \bar{\mathbf{F}}_{N,k}\|_2^2$ as the loss function, we observe the following feature reconstruction loss for encoder distillation converges better:

$$\mathcal{L}_{enc}^N(I_k) = \|\mathbf{W}_{N,g}^T \bar{\mathbf{F}}_{N,k}^e - \bar{\mathbf{F}}_{N,k}\|_2^2. \quad (6)$$

This loss also effectively makes enc_N produce $\bar{\mathbf{F}}_{N,k}^e$ equal to $\mathbf{W}_{N,g} \bar{\mathbf{F}}_{N,k}$.⁵ We suspect the better convergence is due to its stronger constraint: $\mathcal{L}_{enc}^N(I_k)$ imposes $C_N H_{N,k} W_{N,k}$ scalar constraints, while $\|\bar{\mathbf{F}}_{N,k}^e - \mathbf{W}_{N,g} \bar{\mathbf{F}}_{N,k}\|_2^2$ imposes $C_N' H_{N,k} W_{N,k}$ scalar constraints, which are C_N/C_N' times ($\sim 10x$) looser and might be casually fulfilled.

In the decoder implementation, given the image I_k we want to make the output $\mathbf{F}_{N-1,k}^d$ of dec_N to reproduce the input $\mathbf{F}_{N-1,k}^e$ of enc_N and the reconstructed image $I_{k_{rec}}$ from dec_1 close to the input I_k . Collectively, we minimize \mathcal{L}_{dec}^N consisting of three terms:

$$\mathcal{L}_{dec}^N(I_k) = \|\mathbf{F}_{N-1,k}^d - \mathbf{F}_{N-1,k}^e\|_2^2 + \|I_{k_{rec}} - I_k\|_2^2 + \|\mathbf{F}_{N,k_{rec}} - \mathbf{F}_{N,k}\|_2^2, \quad (7)$$

where the third term is perceptual loss used to boost the image reconstruction. Note that when $N = 1$, there is no first term for feature reproduction.

To summarize, when training the pair of enc_N and dec_N ,

⁵We prove this in the Supplementary Materials.

we solve the following optimization problem:

$$\min_{enc_N, dec_N} \mathcal{L}_{enc}^N(I_k) + \mathcal{L}_{dec}^N(I_k). \quad (8)$$

3.3. Channel lengths of the target model

We now show how we select the channel lengths C_N^e 's of the target model enc . Following the rule of thumb [5, 13, 14, 33, 37] in PCA dimension reduction to keep the most important information, the target layer $reluN_I$ of the channel length C_N^e should preserve 85% of the variance information from the source layer $reluN_I$. The channel length that preserves this amount of information can be determined with a data-driven approach.

For each image I_k in our dataset, we compute the covariance of its feature $\mathbf{F}_{N,k}$ extracted at the $reluN_I$ layer. Let $\sigma_{N,k}^j$ be the j -th largest eigenvalue of the covariance associated with the j -th principal eigenvector $\mathbf{e}_{N,k}^j$. The j -th explained variance (EV) $\sigma_{N,k}^j / \sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha$ reflects the portion of the feature variance captured by $\mathbf{e}_{N,k}^j$. Then the cumulative EV (CEV) $\sum_{\beta=1}^{C_N'} \sigma_{N,k}^\beta / \sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha$ reflects the feature variance captured by the top C_N' eigenvectors. We use mean cumulative explained variance (mCEV) to determine the value of C_N^e . The mean CEV (mCEV) is the average of the CEVs across all images:

$$\text{mCEV}(C_N') = \frac{1}{M} \sum_{k=1}^M \frac{\sum_{\beta=1}^{C_N'} \sigma_{N,k}^\beta}{\sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha} = \sum_{\beta=1}^{C_N'} \text{mEV}(\beta), \quad (9)$$

where M is the number of images in the dataset and $\text{mEV}(\beta) = \frac{1}{M} \sum_{k=1}^M (\sigma_{N,k}^\beta / \sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha)$ is the mean β -th

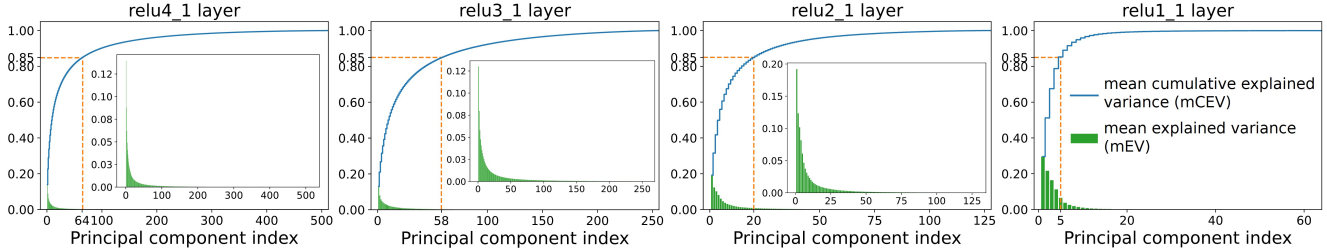


Figure 4. Mean explained variance (green histogram) and mean cumulative explained variance (blue curve) of the $reluN_1$ features of MS-COCO images. It is observed that on average 85% of the variance of a $relu4_1$, $relu3_1$, $relu2_1$, or $relu1_1$ feature can be explained by 64, 58, 20, or 5 eigenvectors of the feature covariance, respectively.

EV. We look for a C_N^e that satisfies $mCEV(C_N^e) \approx 85\%$. We use the MS-COCO [26] training set of 118,287 images because of its large number of diversified images.

The mCEV and mEV for each $reluN_1$ layer of VGG-19 are shown in Fig. 4. We observe that 85% of the variance information in the original 512 (256, 128, 64)-dimensional space of the $relu4_1$ ($relu3_1$, $relu2_1$, $relu1_1$) layer can be explained by, on average, 64 (58, 20, 5) principal components. In other words, across the four layers, a small percentage of the original number of channels are needed to preserve 85% of the variance information; i.e., 12.5% (22.7%, 15.6%, 7.8%).

Experimentally, we found that setting C_1^e to 5 with C_2^e set to 20 hinders the distillation from the $relu2_1$ layer to the $relu2_1$ layer. We suspect it is due to the low reduction rate of 7.8% for the $relu1_1$ layer. We found experimentally that doubling the reduction rate and setting C_1^e to 10 overcomes this issue. For our final model, we set the four channel lengths ($C_1^e, C_2^e, C_3^e, C_4^e$) to (10, 20, 58, 64).

4. Experiments

We evaluate our models with respect to model size (Sec. 4.1), inference time at different image resolutions (Sec. 4.2), and quality of rendered images (Sec. 4.3).

4.1. Model size

Our models. We apply our PCA knowledge distillation to distill two models from VGG and MobileNet. We refer to these models as Ours-VGG and Ours-Mob, respectively.

To assess the benefit of using PCA to derive the channel lengths for a target model, we also evaluate against a PCA-distilled model using the channel lengths empirically selected for CKD [43]: $C_1^e = 16$, $C_2^e = 32$, $C_3^e = 64$, and $C_4^e = 128$. We call this variant Ours-VGG-CKD.

Following prior work [11], we apply high-frequency residuals (HFR) to all of our models to support good high-frequency detail construction.

Baselines. We compare to four models.⁶ One model is distilled with the only prior distillation method for style transfer: CKD [43]. We also evaluate three state-of-the-art non-distilled models: WCT² [48], PhotoWCT² [11], and PhotoWCT [24]. To support fair comparison, we apply HFR to reinforce the high-frequency detail construction for those models that lack this feature and so suffer lossy structure for high-frequency details (as discussed in Sec. 2): PhotoWCT [24] and the CKD model. We show in the Supplementary Materials that both variants achieve better content preservation than the original versions, with little impact to model size and speed.

Results. Results are reported in Tab. 1(a).

Compared to the existing models, our models have far fewer model parameters. For example, Ours-VGG uses only 2.8%, 3.4%, and 4.0% of the parameters of WCT², PhotoWCT-HFR, and PhotoWCT², respectively. Given that MobileNet is a lightweight alternative to VGG-19, we observe that Ours-Mob uses fewer parameters than Ours-VGG: it uses only 0.7%, 0.9%, and 1.0% of the parameters of WCT², PhotoWCT-HFR, and PhotoWCT², respectively.

Compared to the baseline distilled model, CKD, Ours-VGG-CKD uses fewer parameters. We attribute that to the fact that Ours-VGG-CKD uses a single autoencoder while CKD uses a cascade of four autoencoders.

When comparing the sizes of models distilled using the channel lengths derived with PCA and those empirically selected for CKD [43], Ours-VGG uses 64.3% parameters of Ours-VGG-CKD. In other words, our theoretically motivated approach results in a more compact model.⁷

⁶We exclude a couple recent models from this experiments section for the following reasons. Due to the huge size, PhotoNAS [4] cannot handle the smallest considered image resolution (HD) in this paper and so is not used for comparison. LST [22] is another autoencoder-based method, which is designed for artistic style transfer and is insufficient for photorealistic style transfer, as we show in the Supplementary Materials.

⁷Note that a certain amount of variance needs to be preserved so that the model compression does not result in an architecture with too little capacity to effectively learn. Through experimentation, we found the lower bound for the variance information percentage is around 75%, which results in 59K parameters with VGG as the source model. While the distilled model

| Model | (a) Size | (b) Inference time | | | | | |
|-----------------------|-------------|--------------------|-------------|-------------|-------------|-------------|-------------|
| | | HD | FHD | QHD | 4K | 5K | 8K |
| WCT ² | 10.12M | 0.37 | 0.80 | 1.20 | OOM | OOM | OOM |
| PhotoWCT-HFR | 8.35M | 0.56 | 0.84 | 1.27 | 2.53 | 4.36 | OOM |
| PhotoWCT ² | 7.05M | 0.32 | 0.45 | 0.69 | 1.26 | 2.14 | OOM |
| CKD | 526K | 0.09 | 0.16 | 0.25 | 0.53 | 0.93 | 2.09 |
| Ours-VGG-CKD | 440K | 0.06 | 0.09 | 0.13 | 0.25 | 0.42 | 0.92 |
| Ours-VGG | 283K | 0.05 | 0.07 | 0.11 | 0.22 | 0.38 | 0.82 |
| Ours-Mob | 73K | 0.04 | 0.05 | 0.06 | 0.10 | 0.17 | 0.38 |

Table 1. Size and inference time of different models, with all distilled models highlighted in light gray. Note that all models except the separated Ours-Mob are distilled from VGG-19. Compared to existing models, our models are smaller and achieve higher speeds. OOM: Out of memory. Unit: Second/Image.

4.2. Inference time

Intuitively, smaller models should enable faster stylization and support larger image resolutions. We demonstrate these benefits here. We conduct all experiments on an Nvidia RTX8000 GPU with 48GB memory. For completeness, we report CPU times in the Supplementary Materials.

Our models and baselines. We test the same models and baselines as used in Sec. 4.1.

Dataset. We test all models on six resolutions: 1280×720 (HD), 1920×1080 (Full HD), 2560×1440 (Quad HD), 3840×2160 (4K UHD), 5120×2880 (5K), and 7680×4320 (8K). We collect images for testing by downloading an 8K video [1] from YouTube, sampling a frame per second to collect 100 frames, and then downsampling each frame to the other lower resolutions.

Results. We report the speed of different models for stylizing images of different resolutions in Tab. 1.

Both our PCA-based models, Ours-VGG and Ours-Mob, run the fastest for all considered resolutions of the VGG-based models and of all models, respectively. Compared to the fastest non-distilled model PhotoWCT², Ours-VGG is consistently 5-6x faster in the resolutions supported by PhotoWCT², while Ours-Mob is 8x faster in HD image stylization and 12.6x faster in 4K and 5K image stylization. Compared to the slowest non-distilled model WCT², Ours-VGG is 7x faster in HD image stylization and 11x faster in FHD and QHD image stylization, while Ours-Mob is 9x, 16x, and 20x faster in HD, FHD, and QHD image stylization, respectively. Compared to the only baseline distilled

with 75% of the variance information marginally decreases the content loss from Ours-VGG (2.20e⁶ vs. 2.25e⁶) and slightly increases the style loss (4.90e⁴ vs. 4.77e⁴), the distillation of this model is unstable and relies on good initialization of parameter values.

model, CKD, Ours-VGG is consistently 2-2.5x faster in all resolutions while Ours-Mob can even run 5.3-5.5x faster in 4K, 5K and 8K resolutions.

Regarding supported resolutions, only distilled models can support the highest resolution tested: 8K. Moreover, WCT² is not even able to support 5K resolutions. This finding highlights that distilling models to smaller sizes is critical for supporting higher resolution visual data.

4.3. Content preservation and stylization strength

We assess the quality of our models’ rendered images.

Our models. We again evaluate Ours-VGG, Ours-Mob, and Ours-VGG-CKD, which are described in Sec. 4.1. Additionally, to assess the benefit of using global eigenbases to distill our models, we also evaluate models distilled using image-dependent eigenbases: Ours-VGG-Local and Ours-Mob-Local, where *Local* denotes local eigenbases.

Baselines. We test the same baselines as used in Sec. 4.1.⁸

Metrics. We adopt the de facto standard metrics to measure content preservation and stylization strength, which were established in [15]. Specifically, given a stylized image I_t , we calculate its content loss $\|\bar{\mathbf{F}}_{4,c} - \bar{\mathbf{F}}_{4,t}\|_2^2$ from the content image I_c and its style loss $\sum_{N=1}^4 \|\text{Cov}(\mathbf{F}_{N,s}) - \text{Cov}(\mathbf{F}_{N,t})\|_2^2$ from the style image I_s .

We also assess the quality of an image I_t , using standard metrics employed in the image quality assessment community: SSIM [45], FSIM [50], and NIMA [41]. While the first two metrics measure the structure similarity between I_t and I_c , the third metric evaluates I_t as a standalone image.

Dataset. We use PST dataset [46], the largest publicly available dataset for evaluating photorealistic style transfer models. It consists of 786 pairs of content and style images, where the resolutions of images are between HD and FHD.

Results. Qualitative results are shown in Fig. 5 and quantitative results are shown in Fig. 6.

Overall, our models distilled with global eigenbases achieve the best balance of content preservation and stylization strength (Fig. 6(a)). This is evident when examining the two extremes for the content-style trade-off: WCT² sets the upper bound for style loss since it achieves the weakest stylization strength while PhotoWCT-HFR sets the upper bound for content loss since it achieves the worst content preservation. We denote the region set by these two extremes in gray. Lighter gray denotes the region where the trade-off is a smaller amount of stylization strength/content from PhotoWCT-HFR/WCT² for a larger gain in content preservation/stylization strength. All our distilled models

⁸For completeness, we show in the Supplementary Materials that CKD’s image quality is not better when using our smaller channel lengths.

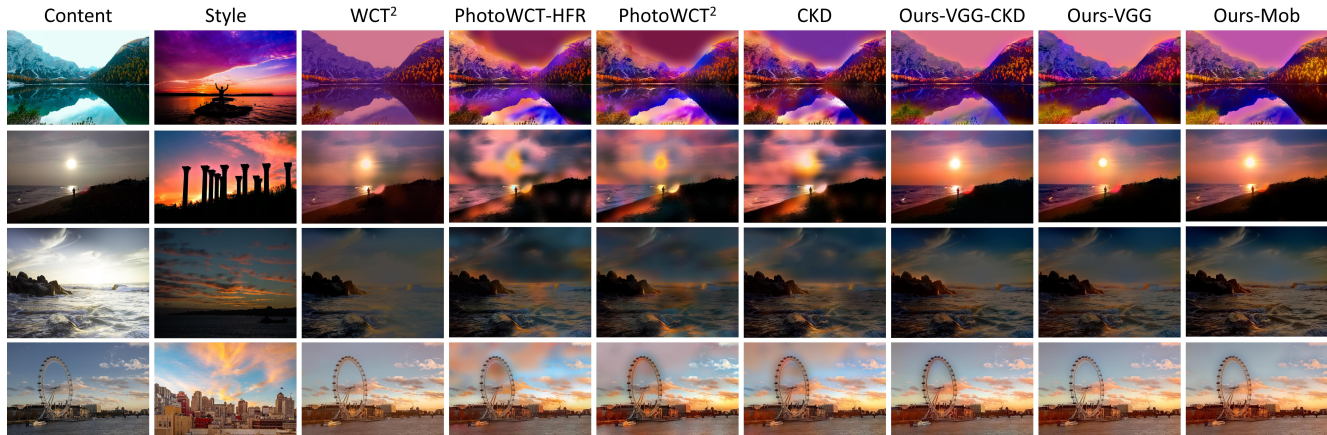


Figure 5. Our PCA-distilled models result in stronger style effects than WCT^2 and better photorealism than the other methods, achieving a better content-style balance. Please see more stylized images in Supplementary Materials.

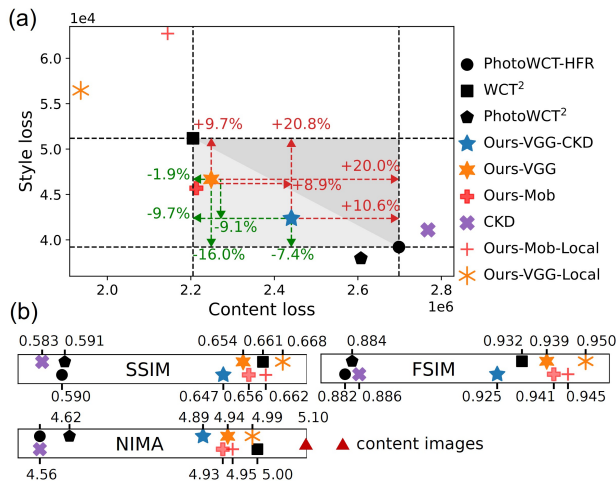


Figure 6. (a) Content vs. style losses resulting from different models. Our models distilled with global eigenbases (Our-VGG-CKD, Our-VGG, Our-Mob) can trade-off a smaller amount of stylization strength/content from PhotoWCT/ WCT^2 for a larger gain in content preservation/stylization strength. (b) Quality scores of stylized images resulting from different models. Our models consistently achieve higher scores.

(Ours-VGG, Ours-Mob, Ours-CKD) achieve a better balance between stylization strength and content preservation, with our most compact model (i.e., Ours-Mob) achieving the best balance. We suspect this is because PCA filters out marginal pattern information (e.g. cloud contours, ripples) that can cause artifacts while preserving the dominant color information. In contrast, we suspect that the high content loss of CKD reflects that CKD poorly filters out the non-essential pattern information, as shown in Fig. 5.

Compared to Ours-VGG-CKD, Ours-VGG further improves the content preservation by trimming extra style information that results in slight artifacts⁹, which the 9.1%

⁹The slight artifacts are not clear in the small resolutions in Fig. 5.

lower style loss of Ours-VGG-CKD mostly contributes to. This highlights that, with the guidance from the PCA theory, we can properly select the channel lengths for our models to achieve a better content-style balance than when empirically selecting from trials.

The benefit of our PCA distillation strategy is validated by comparing models based on local eigenbases (i.e., Ours-VGG-Local and Ours-Mob-Local) to our models based on global eigenbases. We observe a stronger stylization strength in our models based on global eigenbases (qualitative comparisons are shown in Supplementary Materials).

Overall, we observe similar trends to those noted above with the quality metrics (Fig. 6(b)). Our models outperform existing distilled and non-distilled models.

4.4. Limitation

While the primary benefit of our work is to demonstrate that we can create ultra-compact models that run very fast, our work also highlight that stylizing images of higher resolutions is now in reach (i.e., 4K resolutions and beyond). However, we show in the Supplementary Materials that our models occasionally result in artifacts for the larger resolution images, since they contain more high-frequency details which are hotbeds where artifacts form. Future work should establish a new benchmark dataset for photorealistic stylization that includes larger resolution images.

5. Conclusion

We propose the first knowledge distillation for photorealistic style transfer, which is motivated by PCA theory. Compared to existing distilled and non-distilled models, our models are smaller, faster, and achieve a better balance between content preservation and stylization strength.

We show stylized images of 4K+ resolutions in Supplementary Materials where the slight artifacts become apparent.

References

- [1] 8k Taiwan. <https://www.youtube.com/watch?v=PB4gId2mPNc>. Accessed: 2021-10-31. 7
- [2] L1-norm principal component analysis. https://en.wikipedia.org/wiki/L1-norm_principal_component_analysis. Accessed: 2021-10-31. 3
- [3] Principal component analysis. https://en.wikipedia.org/wiki/Principal_component_analysis. Accessed: 2021-10-31. 2, 3
- [4] Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo. Ultrafast photorealistic style transfer via neural architecture search. In *AAAI*, pages 10443–10450, 2020. 1, 2, 6
- [5] Fiorenzo Artoni, Arnaud Delorme, and Scott Makeig. Applying dimension reduction to eeg data by principal component analysis reduces the quality of its subsequent independent component decomposition. *NeuroImage*, 175:176–187, 2018. 5
- [6] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017. 2
- [7] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016. 1
- [8] Xinghao Chen, Yiman Zhang, Yunhe Wang, Han Shu, Chunjing Xu, and Chang Xu. Optical flow distillation: Towards efficient and stable video style transfer. In *European Conference on Computer Vision*, pages 614–630. Springer, 2020. 3
- [9] Tai-Yin Chiu. Understanding generalized whitening and coloring transform for universal style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4452–4460, 2019. 4
- [10] Tai-Yin Chiu and Danna Gurari. Iterative feature transformation for fast and versatile universal style transfer. In *European Conference on Computer Vision*, pages 169–184. Springer, 2020. 2
- [11] Tai-Yin Chiu and Danna Gurari. Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. *arXiv preprint arXiv:2110.11995*, 2021. 1, 2, 4, 6
- [12] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802, 2019. 2
- [13] Francesca Corvucci, Lara Nobili, Dora Melucci, and Francesca-Vittoria Grillenzoni. The discrimination of honey origin using melissopalynology and raman spectroscopy techniques coupled with multivariate analysis. *Food chemistry*, 169:297–304, 2015. 5
- [14] Shriram Gajjar, Murat Kulahci, and Ahmet Palazoglu. Real-time fault detection and diagnosis using sparse principal component analysis. *Journal of Process Control*, 67:112–128, 2018. 5
- [15] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 1, 3, 7
- [16] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, and Jonathon Shlens. Exploring the structure of a real-time, arbitrary neural artistic stylization network. *arXiv preprint arXiv:1705.06830*, 2017. 1
- [17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2
- [18] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2
- [19] 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. 1
- [20] 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. 1
- [21] Seunghyun Lee and Byung Cheol Song. Interpretable embedding procedure knowledge transfer via stacked principal component analysis and graph neural network. *arXiv preprint arXiv:2104.13561*, 2021. 3
- [22] Xueting Li, Sifei Liu, Jan Kautz, and Ming-Hsuan Yang. Learning linear transformations for fast image and video style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3809–3817, 2019. 1, 6
- [23] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. In *Advances in neural information processing systems*, pages 386–396, 2017. 1, 2, 4
- [24] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 453–468, 2018. 1, 2, 6
- [25] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 3
- [26] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 6
- [27] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019. 2
- [28] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2019. 2
- [29] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. Structured knowledge distillation for dense predic-

- tion. *IEEE transactions on pattern analysis and machine intelligence*, 2020. [2](#)
- [30] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4990–4998, 2017. [1](#), [2](#)
- [31] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020. [2](#)
- [32] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852*, 2017. [2](#)
- [33] Monica Odlare, Kalle Svensson, and Mikael Pell. Near infrared reflectance spectroscopy for assessment of spatial soil variation in an agricultural field. *Geoderma*, 126(3-4):193–202, 2005. [5](#)
- [34] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. [2](#)
- [35] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016, 2019. [2](#)
- [36] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *International Conference on Machine Learning*, pages 5142–5151. PMLR, 2019. [2](#)
- [37] Homa Razmkhah, Ahmad Abrishamchi, and Ayoob Torkian. Evaluation of spatial and temporal variation in water quality by pattern recognition techniques: A case study on jajrood river (tehran, iran). *Journal of environmental management*, 91(4):852–860, 2010. [5](#)
- [38] Mohamed El Amine Seddik, Hassane Essafi, Abdallah Benzine, and Mohamed Tamaazousti. Lightweight neural networks from pca & lda based distilled dense neural networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3060–3064. IEEE, 2020. [3](#)
- [39] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8242–8250, 2018. [1](#)
- [40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [1](#)
- [41] Hossein Talebi and Peyman Milanfar. Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011, 2018. [7](#)
- [42] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019. [2](#)
- [43] Huan Wang, Yijun Li, Yuehai Wang, Haoji Hu, and Ming-Hsuan Yang. Collaborative distillation for ultra-resolution universal style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1860–1869, 2020. [1](#), [3](#), [6](#)
- [44] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Kdgan: Knowledge distillation with generative adversarial networks. In *NeurIPS*, pages 783–794, 2018. [2](#)
- [45] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. [7](#)
- [46] Xide Xia, Meng Zhang, Tianfan Xue, Zheng Sun, Hui Fang, Brian Kulis, and Jiawen Chen. Joint bilateral learning for real-time universal photorealistic style transfer. *arXiv preprint arXiv:2004.10955*, 2020. [1](#), [2](#), [7](#)
- [47] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017. [2](#)
- [48] Jaejun Yoo, Youngjung Uh, Sanghyuk Chun, Byeongkyu Kang, and Jung-Woo Ha. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9036–9045, 2019. [1](#), [2](#), [6](#)
- [49] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020. [2](#)
- [50] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. [7](#)