Photorealistic Style Transfer via Wavelet Transforms

Jaejun Yoo*  Youngjung Uh*  Sanghyuk Chun*  Byeongkyu Kang  Jung-Woo Ha
Clova AI Research, NAVER Corp.
{jaejun.yoo,youngjung.uh,sanghyuk.c,bk.kang,jungwoo.ha}@navercorp.com

Abstract

Recent style transfer models have provided promising artistic results. However, given a photograph as a reference style, existing methods are limited by spatial distortions or unrealistic artifacts, which should not happen in real photographs. We introduce a theoretically sound correction to the network architecture that remarkably enhances photorealism and faithfully transfers the style. The key ingredient of our method is wavelet transforms that naturally fits in deep networks. We propose a wavelet corrected transfer based on whitening and coloring transforms (WCT2) that allows features to preserve their structural information and statistical properties of VGG feature space during stylization. This is the first and the only end-to-end model that can stylize 1024 × 1024 resolution image in 4.7 seconds, giving a pleasing and photorealistic quality without any post-processing. Last but not least, our model provides a stable video stylization without temporal constraints. The code, generated images, supplementary materials, and pre-trained models are all available at ClovaAI/WCT2.

1. Introduction

Photorealistic style transfer has contradictory objectives. To be photorealistic, a model should apply the reference style on the scene without hurting the details of an image. In figure 1, for example, the general style (color and tone) of sky and sea should change, while fine structures of ship and bridges remain intact. However, artistic style transfer methods (e.g., whitening and coloring transforms, WCT [18]) generally suffer from severe distortions due to their strong abstraction ability, which is not favored in the photorealistic stylization (figure 1b) (Please refer to our supplementary materials for more failure cases).

Luan et al. [22] introduced a regularizer for photorealism on the traditional optimization-based method [8]. However, solving the optimization problem requires heavy computational costs, which limits their use in practice. To overcome this issue, Li et al. [19] recently proposed a photorealistic variant of WCT (PhotoWCT) that replaced the upsampling components of the VGG decoder with unpooling. By pro-

* indicates equal contribution
progressive stylization and component-wise transformations on the feature space as well, such as decomposed wavelet features provide interesting interpretations on the feature space as well, such as component-wise stylization and why average pooling is known to give better stylization than max-pooling (Section 4.1).

In addition, we propose progressive stylization instead of following the multi-level strategy that is used in the WCT [18] and PhotoWCT [19] (figure 2). To maximize the stylization effect, WCT and PhotoWCT recursively transformed features in a multi-level manner from coarse-to-fine. In contrast, we progressively transform features during a single pass. This allows two significant advantages over the others. First, our model is simple and efficient since we only have a single decoder during training as well as in the inference time. On the other hand, the multi-level strategy requires to train a decoder for each level without sharing parameters, which is inefficient in terms of the number of parameters and training procedure. This overhead remains in the inference time as well because the model requires to pass multiple encoder and decoder pairs to stylize an image. Second, by recursively encoding and decoding the signal with the lossy VGG networks, artifacts are amplified during the multi-level stylization. Because of wavelet operations and progressive stylization, our model does not have such problem, and even more, it shows little error amplification when the multi-level strategy is employed (figure 6).

Our contributions are summarized as follows. By employing the wavelet corrected transfer and progressive stylization, we introduce the first end-to-end photorealistic style transfer model that allows to remove the additional post-processing steps, which is $\sim 25 \times$ faster than the state-of-the-art models (table 1). It is the only model that can process a high resolution image ($1024 \times 1024$) in 4.7 seconds, where PhotoWCT fails due to out-of-memory and Deep Photo Style Transfer (DPST) [22] takes 3887.8 seconds. Our experimental results show quantitatively better visual quality in both SSIM and Gram loss (figure 9), and qualitatively preferred by 62.21% in the user study (table 2). In addition, our model has three times less parameters than PhotoWCT and provides temporally stable stylization enabling video applications without additional constraints, such as optical flow (figure 8).
2. Related Work

Style transfer. Starting from the seminal work of Gatys et al. [8], many artistic style transfer studies have been proposed to synthesize stylized images through either iterative optimization [9], finding dense correspondence [20, 28, 10] or manipulating features in pre-trained networks [13, 18]. Due to their powerful ability to abstract the features, however, they cannot be used in the photorealistic style transfer scenario as they are.

Compared to the artistic style transfer, photorealistic transfer has been overlooked. The classical methods are mostly matching the color and tone [17, 24, 25] of the images, which are restricted to specific usage. Luan et al. [22] proposed deep photo style transfer (DPST), which augments the Neural Style algorithm [8] with an additional photorealism regularization term and a semantic segmentation mask. However, DPST requires heavy computation to solve regularized optimization problem.

Recently, Li et al. [19] proposed a photorealistic variant of WCT (PhotoWCT). It replaces the upsampling of the VGG decoder with unpooling. PhotoWCT showed that the spatial distortion could be relaxed by providing max-pooling masks to the decoder. However, because the visual quality of the raw outputs of PhotoWCT were not satisfactory, the authors had to employ additional post-processing, such as smoothing and filtering. However, not only do these increase runtime exponentially to the image resolution, but blur final outputs.

Different from the existing methods, our method can preserve the fine structures of an image with little spatial distortion in an end-to-end manner, and thus removes the necessity of additional post-processing steps.

Signal reconstruction using wavelets. Signal reconstruction using wavelets has been an extensive research topic in applied mathematics community due to its favorable characteristics such as proven convergence and compact representation of an arbitrary signal [6, 17]. There have been several attempts to incorporate both classical signal processing and deep learning approaches, including feature reduction [16], network compression [11, 16], super-resolution [2], classifications [3, 7, 23, 27, 30] and image denoising [14]. Similarly, our approach augments wavelets as a component of network architecture and provides an interpretable module that can enhance the photorealism of a style transfer model.

One closest related work [30] recently proposed to use wavelets as an alternative to traditional neighborhood pooling. However, their goal is to reduce feature dimensions by discarding the first-level sub-bands, which is in the opposite direction of ours. In addition, we utilize both wavelet decomposition and reconstruction together to exactly recover the spatial information with minimal noise amplification.

3. WCT^2

To achieve photorealism, a model should recover the structural information of a given content image while it stylizes the image faithfully at the same time. To address this issue, we propose a wavelet corrected transfer based on whitening and coloring transforms, dubbed WCT^2. More specifically, we handle the first objective by employing wavelet pooling and unpooling which could preserve information of the content to the transfer network. We use a progressive stylization within a single forward pass to tackle the second issue.

3.1. Wavelet corrected transfer

Haar wavelet pooling and unpooling. We first explain the main components of our model using Haar wavelets, which we call wavelet pooling and unpooling. Haar wavelet pooling has four kernels, \( \{L^\top L, H^\top L, H^\top H, H^\top H\} \), where the low (L) and high (H) pass filters are

\[
L^\top = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad H^\top = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \end{bmatrix}.
\]

Thus, unlike common pooling operations, the output of the Haar wavelet pooling has four channels. Here, the low-pass filter captures smooth surface and texture while the high-pass filters extract vertical, horizontal, and diagonal edge-like information. For simplicity, we denote the output of each kernel as LL, LH, HL and HH, respectively.

One important property of our wavelet pooling is that the original signal can be exactly reconstructed by mirroring its operation; i.e., wavelet unpooling. In detail, wavelet unpooling fully recovers the original signal by performing a component-wise transposed-convolution and followed by a summation (please see our supplementary materials for more details). Thanks to this favorable property, our proposed model can stylize the image with minimal information loss and noise amplification. On the other hand, max-pooling does not have its exact inverse so that the encoder-decoder structured networks used in the WCT [18] and PhotoWCT [19] cannot fully restore the signal.

Note that Haar wavelet pooling and unpooling is not the only operation which can fully recover the original signal. However, we choose Haar wavelet because it splits the original signal into channels that capture different components, which leads to better stylization.

Model architecture. To fully utilize the encoded information, we replace every max-pooling and unpooling of PhotoWCT with the wavelet pooling and unpooling (figure 2). Specifically, we use the ImageNet [5] pre-trained VGG-19 network [29] from conv1_1 layer to conv4_1 layer as the encoder. The max-pooling layers are replaced with wavelet pooling where high frequency components (LH,
HL, HH) are skipped to the decoder directly. Thus, only the low frequency component (LL) is passed to the next encoding layer. The decoder has a mirror structure of the encoder, and the wavelet unpooling aggregates the components (please refer to our supplementary materials for more details about the proposed network architecture).

3.2. Stylization

Whitening and coloring transforms (WCT). Because our method is built upon WCT [18], we first recap WCT briefly. WCT can perform style transfer with arbitrary styles by directly matching correlation between content and style in the VGG feature domain. It projects the content features to the eigenspace of style features by calculating singular value decomposition (SVD). The final stylized image is obtained by feeding the transferred features into the decoder. To provide better artistic style transfer, the authors employed a multi-level stylization framework by applying WCT to multiple encoder-decoder pairs (figure 2b).

Progressive stylization. Instead of using the multi-level stylization used in WCT and PhotoWCT, we progressively transform features within a single forward-pass as illustrated in figure 2. We sequentially apply WCT at each scale (conv1_X, conv2_X, conv3_X and conv4_X) within a single encoder-decoder network. Note that the number of SVD computations of our model remains the same. We can add more WCTs on skip-connections and decoding layers to further strengthen the stylizing effect at the cost of time consumption. This will be covered in more detail in Section 4.4. There are several advantages in the proposed progressive stylization against the multi-level one. First, the multi-level strategy requires to train a decoder for each level without sharing parameters, which is inefficient. On the other hand, our training procedure is simple because we only have a single pair of encoder and decoder, which is advantageous in the inference time as well. Second, recursively encoding and decoding the signal with VGG network architecture amplifies errors so that it causes unrealistic artifacts in the output. In the later section, we show that our proposed progressive stylization technique suffers less from the error amplification than the multi-level strategy.

4. Analysis

4.1. Wavelet pooling

We first examine the effects of using the wavelet pooling instead of max-pooling. As shown in figure 3b and 3c, PhotoWCT suffers from the loss of spatial information by max-pooling while ours preserves fine details. We recall that low frequency component captures smooth surface and texture while high frequency components detect edges. This enables our model to separately control the stylization effect by choosing a component. More specifically, it implies that applying WCT to LL of the encoder affects overall texture or surface while applying WCT to high frequency components (i.e., LH, HL, HH) stylize edges. Indeed, when we stylize all components (figure 3c), our model transfers the given style to the entire building. In contrast, if we do not perform WCT on the high frequency components, the boundaries of windows remain unchanged (figure 3d).

Note that using only the LL component of our wavelet pooling is equivalent to using the average pooling. Interestingly, since Gatys et al. [8], many studies have consistently reported that replacing the max-pooling operation with average pooling yields slightly more appealing results. This can be explained in our framework that the model is using the partial information (LL) of the wavelet decomposed feature domain. In addition, because each frequency component of the content feature is transformed into its corresponding component of style feature, we can obtain similar advantage as we do by using spatial correspondences. Recall that DPST and PhotoWCT employ semantic label maps to transform the features within corresponding regions.

4.2. Ablation study

To show that our model indeed benefits from the wavelet pooling, we compare the stylization results using the other pooling variants. We unpool the features similarly to the way we do for the wavelet unpooling; i.e., transposed-convolution and summation.

Figure 3: Comparison between max-pooling and wavelet pooling. Given (a) an input pair (inset: style), we compare the results of (b) PhotoWCT without post-processing, (c) ours and (d) ours but stylize only the LL component. Note that the edges are left unstylized (inside the red box).
Figure 4: Ablation study on pooling methods. While split and learnable poolings suffer from the lack of representation power or altered feature statistics, wavelet pooling benefits from the compact representation of wavelets and retains the original VGG feature property intact.

**Split pooling.** Split pooling has four $2 \times 2$ filters with fixed weights, $[1 \ 0 \ 0 \ 0]$, $[0 \ 1 \ 0 \ 0]$, $[0 \ 0 \ 1 \ 0]$, and $[0 \ 0 \ 0 \ 1]$. Split pooling has a similar property to wavelet pooling in that it can carry whole information and preserve characteristics of the pre-trained network. Here, we can see similar effect but a slight degradation in fine details. We suppose that the reason comes from the lack of representation power so that we can often find failure cases, such as the grass (figure 4b).

**Learnable pooling.** Learnable pooling is a trainable conv layer with stride two. As shown in figure 4c, it does not preserve the content nor faithfully transfer the style. We suppose that this happens because the learnable pooling brings too much flexibility to the network. This ruins the original feature properties of VGG networks [29], which is known to be good at extracting styles [8].

### 4.3. Unpooling options

To achieve better reconstruction, we adopted concatenation instead of summation for unpooling, similar to U-Net structure [12, 26, 34]. This enables the network to learn the weighted sum of components at the expense of interpretability and theoretical correctness. Specifically, our wavelet unpooling now performs channel-wise concatenation of four feature components from the corresponding scale plus feature output before the wavelet pooling. Therefore, the number of parameters increases at the conv layer that comes right after the wavelet unpooling. This increases the total number of parameters to be $1.80 \times$ of the sum-version of WCT$^2$ while PhotoWCT has $3.06 \times$ parameters. As shown in figure 5, the spatial details are further improved. The sum-version generally gives more stylized output while the concatenated-version provides clearer image (please see our supplementary materials for more results).

### 4.4. Progressive vs. multi-level strategy

Owing to the exact reconstruction property of wavelet pooling, our model can adopt the multi-level strategy to increase the contrast in the transferred style with minimal noise amplification. As shown in figure 6d, adopting the multi-level approach in addition to WCT$^2$ leads to more vivid results. Note that it maintains photorealism while PhotoWCT produces spotty artifacts due to noise amplification (figure 7c). In addition, performing progressive stylization at the decoder as well, namely conv3.2, conv2.2, and conv1.2, further increases stylization effect. Still, strengthening the style come at the cost of photorealism and multiple SVD computations (please refer to the supplementary material for more results).
Figure 7: Photorealistic stylization results. Given (a) an input pair (top: content, bottom: style), the results of (b) deep photo style transfer (DPST) [22], (c) and (d) PhotoWCT [19], and (e) ours (WCT\(^2\)) are shown. PhotoWCT (full) denotes the results after applying two post-processing steps proposed by the authors [19]. Note that WCT\(^2\) does not need any post-processing.
5. Experimental results

In this section, we show that our simple modification can remarkably enhance the performance of photorealistic style transfer. Here, every result is reported based on the concatenated version of our model. For a fair comparison and time-efficiency, we only perform whitening and coloring on LL components (e.g., convX.1 outputs of the encoder) progressively. Thus, the number of whitening and coloring procedure of our model matches with PhotoWCT.

5.1. Implementation details

We use the encoder-decoder architecture with fixed VGG encoder weights. The decoder is trained on Microsoft COCO dataset [21], minimizing the L2 reconstruction loss and the additional feature Gram matching loss with the encoder. We use ADAM optimizer [15] with learning rate $10^{-3}$. Finally, we utilize the semantic label map as PhotoWCT and DPST did. More specifically, we perform the whitening and coloring transforms between the features from image regions of corresponding labels. Code will be available online.

5.2. Qualitative evaluation

Figure 7 shows the results of DPST, PhotoWCT and ours (WCT$^2$). DPST often generates a “staircasing” or “cartoon” artifacts [4] with unrealistic color transfer, which severely hurts photorealism (figure 7b). PhotoWCT better reconstructs the details of the content image, while it shows spotty artifacts over entire images (figure 7c). Such artifacts can be removed by employing additional post-processing steps (figure 7d). However, it has three disadvantages that 1) optimization is slow, 2) hyper-parameters should be carefully tuned to trade-off between smoothness and fine details, and 3) final image becomes blurry at the expense of removed artifacts. In contrast, our proposed method shows less artifacts while faithfully transferring the reference styles (figure 7e). Note that we do not apply any post-processing after the network output (please see our supplementary materials for more results and comparison to artistic style transfer methods).

Video stylization. To emphasize consistent feature representation of the wavelet pooling and unpooling, we stylize every video frame to target style without any semantic segmentation. Figure 8 shows that WCT$^2$ performs stable video style transfer without any temporal consistency regularization such as optical flow (the full video can be found in our supplementary materials).

5.3. Quantitative evaluation

Statistics. To measure the photorealism, we employ two surrogate metrics for spatial recovery and stylized amounts. We calculate the structural similarity (SSIM) index between edge responses [31] of original contents and stylized images. Following the WCT [18], we report the covariance matrix difference (VGG style loss [8]) between the style image and the outputs of each model. Figure 9 shows SSIM (X-axis) against style loss (Y-axis). Our proposed model (WCT$^2$) remarkably outperforms other methods.

Note that WCT$^2$ and its variants are located at the top-right corner, superior to PhotoWCT (full) and DPST that perform post-processing. Here, DPST has strength on the Gram-based score in nature because it directly optimizes the style loss. Still, it is far from practical usage due to its heavy optimization procedure, as will be shown in the runtime comparison (table 1). We compare between our models by adding WCT to the decoder or performing stylization in the multi-level manner. As expected, the multi-level approach enhances the performance in Gram-based loss perspective at the expense of SSIM index, which is even better than the optimization-based method (DPST).
In addition, by comparing the gap between before and after the post-processing steps (figure 9, dashed lines), we can clearly see that final visual qualities of PhotoWCT majorly come from the powerful post-processing, especially the smoothing step, not the network itself. The original WCT with smoothing already show comparable result to that of PhotoWCT. This demonstrates that the unpooling substitution of PhotoWCT did not fully addressed the information loss but the post-processing did.

**Runtime comparison.** Table 1 shows the runtime comparison of DPST, PhotoWCT and WCT². For PhotoWCT, we separately measured WCT and post-processing steps to better compare with ours. The reported runtime for each model is an average of ten-rounds run on a single NVIDIA P40 GPU. As expected, our model inherits computational time of the original WCT. Note that the runtime of our method does not significantly increase although we use the concatenated version of WCT². Because our model can remove the cumbersome post-processing steps, WCT² can deal with a high resolution image, such as \(1024 \times 1024\), maintaining a high quality of photorealistic style transfer. Compared to DPST, WCT² achieves maximum about 830 times speed-up in runtime.

**User study.** We conducted a user study to further evaluate the methods in terms of fewer artifacts, faithfulness to the style input, and overall qualities. Our benchmark dataset consists of content and style pairs provided by Luan et al. [22]. Total 40 sets of questions are asked to 41 subjects to choose one among three stylized images from each model. The results are shown in random order with content and style images. Table 2 shows the percentage of the model outputs that are chosen out of 1640 (\(= 40 \times 41\)) responses. Our method is preferred by human subjects against the other state-of-the-art methods by a large margin in all aspects (please see our supplementary materials for more results that are used for user study).

### 6. Conclusion

In this paper, we proposed the first end-to-end photorealistic style transfer method, dubbed WCT². The exact recovery of the wavelet transforms allows our model to preserve structural information while providing stable stylization without any constraints. By employing progressive stylization, we achieved better results with less noise amplification. Compared to the other state-of-the-arts, our analysis and experimental results showed that WCT² is scalable, lighter, faster and achieves better photorealism quantitatively and qualitatively. Our results were preferred by human subjects in every aspect with a significant margin. Future study will include extending this work to other wavelets that have different characteristics, such as rotational invariance. Another interesting direction would be to remove the necessity of semantic segmentation map for stylization.
References

[1] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM Transactions on Graphics (TOG)*, 25(3):637–645, 2006. 3

[2] W. Bae, J. Yoo, and J. C. Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *CVPR Workshops*, pages 1141–1149, 2017. 3

[3] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013. 3

[4] T. F. Chan and J. J. Shen. Image processing and analysis: variational, PDE, wavelet, and stochastic methods, volume 94. Siam, 2005. 7

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009. 3

[6] B. Dong, Q. Jiang, and Z. Shen. Image restoration: wavelet frame shrinkage, nonlinear evolution PDEs, and beyond. *Multiscale Modeling & Simulation*, 15(1):606–660, 2017. 3

[7] S. Fujieda, K. Takayama, and T. Hachisuka. Wavelet convolutional neural networks for texture classification. *arXiv preprint arXiv:1707.07394*, 2017. 3

[8] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423. IEEE, 2016. 1, 3, 4, 5, 7

[9] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. Controlling perceptual factors in neural style transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[10] S. Gu, C. Chen, J. Liao, and L. Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018. 3

[11] L. Guéguen, A. Sergeev, R. Liu, and J. Yosinski. Faster neural networks straight from jpeg. In *ICLR*, 2018. 3

[12] Y. Han and J. C. Ye. Framing u-net via deep convolutional framelets: Application to sparse-view ct. *IEEE transactions on medical imaging*, 37(6):1418–1429, 2018. 5

[13] X. Huang and S. J. Belongie. Arbitrary style transfer in realtime with adaptive instance normalization. In *ICCV*, pages 1510–1519, 2017. 3

[14] E. Kang, W. Chang, J. Yoo, and J. C. Ye. Deep convolutional framelet denosing for low-dose ct via wavelet residual network. *IEEE transactions on medical imaging*, 37(6):1358–1369, 2018. 3

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[16] A. Levinskis. Convolutional neural network feature reduction using wavelet transform. *Elektronika ir Elektrotechnika*, 19(3):61–64, 2013. 3

[17] B. Li and X. Chen. Wavelet-based numerical analysis: a review and classification. *Finite Elements in Analysis and Design*, 81:14–31, 2014. 3

[18] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, pages 386–396, 2017. 1, 2, 3, 4, 7

[19] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A closed-form solution to photorealistic image stylization. *arXiv preprint arXiv:1802.06744*, 2018. 1, 2, 3, 4, 6, 7, 8

[20] J. Liao, Y. Yao, L. Yuan, G. Hua, and S. B. Kang. Visual attribute transfer through deep image analogy. *ACM Transactions on Graphics (TOG)*, 36(4):120, 2017. 3

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[22] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. 1, 2, 3, 6, 8

[23] E. Oyallon, E. Belilovsky, and S. Zagoruyko. Scaling the scattering transform: Deep hybrid networks. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[24] F. Pitie, A. C. Kokaram, and R. Dahyot. N-dimensional probability density function transfer and its application to colour transfer. In *null*, pages 1434–1439. IEEE, 2005. 3

[25] E. Reinhard, M. Adhikmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001. 3

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[27] J. Ryu, M.-H. Yang, and J. Lim. Dft-based transformation invariant pooling layer for visual classification. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[28] L. Sheng, Z. Lin, J. Shao, and X. Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018*. 3

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 5

[30] T. Williams and R. Li. Wavelet pooling for convolutional neural networks. In *ICLR*, 2018. 3

[31] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2017. 7

[32] J. C. Ye, Y. Han, and E. Cha. Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM Journal on Imaging Sciences*, 11(2):991–1048, 2018. 2

[33] R. Yin, T. Gao, Y. M. Lu, and I. Daubechies. A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets. *SIAM Journal on Imaging Sciences*, 10(2):711–750, 2017. 2

[34] J. Yoo, A. Wahab, and J. C. Ye. A mathematical framework for deep learning in elastic source imaging. *SIAM Journal on Applied Mathematics*, 78(5):2791—2818. 5