WHFL: Wavelet-Domain High Frequency Loss for Sketch-to-Image Translation

Min Woo Kim, Nam Ik Cho
Department of ECE, INMC, Seoul National University, Seoul, Korea
{mwk0614, nicho}@snu.ac.kr

Abstract

Even a rough sketch can effectively convey the descriptions of objects, as humans can imagine the original shape from the sketch. The sketch-to-photo translation is a computer vision task that enables a machine to do this imagination, taking a binary sketch image and generating plausible RGB images corresponding to the sketch. Hence, deep neural networks for this task should learn to generate a wide range of frequencies because most parts of the input (binary sketch image) are composed of DC signals. In this paper, we propose a new loss function named Wavelet-domain High-Frequency Loss (WHFL) to overcome the limitations of previous methods that tend to have a bias toward low frequencies. The proposed method emphasizes the loss on the high frequencies by designing a new weight matrix imposing larger weights on the high bands. Unlike existing handcraft methods that control frequency weights using binary masks, we use the matrix with finely controlled elements according to frequency scales. The WHFL is designed in a multi-scale form, which lets the loss function focus more on the high frequency according to decomposition levels. We use the WHFL as a complementary loss in addition to conventional ones defined in the spatial domain. Experiments show we can improve the qualitative and quantitative results in both spatial and frequency domains. Additionally, we attempt to verify the WHFL's high-frequency generation capability by defining a new evaluation metric named Unsigned Euclidean Distance Field Error (UEDFE).

1. Introduction

Sketches are concise and powerful means of intuitive object description. Despite the simplicity, they contain essential information like pose and arrangement of components, and a well-drawn sketch describes an object's shape better than a language. Hence, people often use sketches when they want to explain their ideas or thoughts visually. Also, it became easier to draw and share sketches with the widespread use of smartphones and touchpads. Accordingly, various computer vision tasks related to sketches have been researched, such as sketch recognition [22, 45, 46, 51], sketch-based image retrieval [2–4, 35, 36, 40, 50], and sketch-to-photo translation [7, 9, 23, 25–27, 41, 43, 47].

Sketch-to-photo translation is a computer vision task that takes a binary sketch image as an input and generates an RGB image. Sketch images in computer vision tasks can be roughly divided into edge-map and freehand-sketch, depending on whether edges between sketches and corresponding photos are aligned. Since colorful images need to be generated from a binary input having no color and texture, researchers generally adopt a generative adversarial network (GAN) [11] framework for this task, where convolutional neural networks (CNNs) are usually adopted as generators [18, 24, 31, 32].

However, the CNNs tend to learn low frequency in a biased way, which is called *spectral bias* [17, 33, 34] (see Section D of *supplementary file* for an example of the spectral bias). The bias is caused not only by the network structure but also by loss functions defined in the spatial domain. Since low-frequency parts have order of larger magnitudes than the high in general images, the loss function tends to focus and learn more on the low-frequencies [38]. Since sketches do not have frequency information in most regions other than around the sketch lines, generating both low and high frequency is necessary to make realistic photos. However, the spectral bias makes it difficult for the GAN to generate a wide range of frequencies. As a result, the difference in frequency, called the *frequency gap*, occurs between generated photos and actual ones, which results in an unsatisfactory output or even a degraded image. For example, artifacts in the spatial domain could appear as a repetitive pattern in the frequency domain [1].

To alleviate the frequency gap, various approaches have been proposed regarding network architecture and loss function. In order to transmit high frequency within the network better, Magid *et al*. [28] proposed *Dynamic High-Pass Filtering Layer* (HPF) module and *Matrix Multi Spectral Channel Attention* (MMCA). With the dynamic HPF layer, the network can predict adaptive high-pass kernels for the input feature. Through the MMCA, channels are rescaled using the maximal frequency response after a feature is

transformed to the frequency domain. Xie *et al*. [44] suggested *Frequency-Aware Dynamic Network* (FADN), which separates an input image into low, medium, and high frequency, and then assigns a larger model capacity to high frequency to improve performance and speed. However, since these methods still use loss functions defined in the spatial domain, the spectral bias problem still remains.

Some methods defined the loss in the frequency domain to resolve the frequency gap [5, 17] using the Discrete Fourier Transform (DFT). Specifically, Jiang *et al*. [17] suggested *Focal Frequency Loss* (FFL) used in various tasks including sketch-to-photo translation. This method maps the frequency value at each spectral position to the Euclidean space and calculates the final loss with frequency distance and weight matrix. Cai *et al*. [5] decoupled low and high-frequency bands using a binary mask and measured loss at each band, where the boundary between low and high bands is determined based on spectral energy. However, these methods also have some limitations. If the loss is calculated without refining frequency bands as [17], the low frequency could influence the loss even more than the high frequency. Also, the binary mask adopted in [5] has some room to improve because the mask's boundary is a hand-crafted hyperparameter. Moreover, setting the boundary becomes more challenging work if the size of the training dataset increases because it is necessary to analyze all samples.

In this paper, we design a new loss function to relieve the frequency gap in sketch-to-photo translation. Our method is based on the observation of existing methods that the low-frequency bands are well learned in the spatial domain, but high-frequency bands need explicit frequency-domain manipulations. For the frequency domain processing, we adopt DFT and the frequency distance defined in [17]. This distance is multiplied with a new weight matrix with larger weights on the high frequency, where the weight matrix is designed from the fact that magnitudes of high-frequency components are smaller than those of low bands by an order in most natural images. The loss is designed in a multi-scale form using wavelet transform, which enables more precise control of frequency components. Precisely, the input image is decomposed into multi-scale images by wavelet transform, and the FFL with our new weights is applied for each scale.

The contributions of this paper can be summarized as follows.

- To generate a realistic photo from a given sketch, we propose a loss function named *WHFL* in the frequency domain. The function focuses on the high-frequency with a weight matrix that imposes large weights on the high-frequency band, where weights are adaptively determined based on the frequency scales.

- The proposed loss is applied to multi-scale images decomposed by the wavelet transform, enabling more fine and scale-dependent weighting.

- We obtain the results with quantitative and qualitative improvements in both spatial and frequency domains. Furthermore, we verify the performance of *WHFL* by defining a new metric, Unsigned Euclidean Distance Field Error (*UEDFE*) suggested in Section 5.1.

2. Related Works

2.1. Sketch-to-Photo Translation

Before utilizing deep learning, the direct conversion of a sketch to realistic photos was almost impossible, but there are some researches to retrieve photos from a large database for the given sketch query. For example, Chen *et al*. [6] proposed a framework in which photos are searched on the Internet given corresponding sketches and text labels. In [8], the framework similar to [6] used bag-of-features (BoF) for the retrieval instead of the text label. Although the generated (actually retrieved) images are realistic photos, they often have different shapes and textures.

Chen *et al*.'s method [7], which is the first to adopt a GAN for this task, suggested an encoder-decoder architecture composed of a Masked Residual Unit (MRU). ContextualGAN [27] regards sketch-to-photo translation as an image completion problem. In [10], users can modify the output image interactively with additional sketch strokes. There have also been attempts that users can control the style of generated images by giving a reference image from a target domain as [21, 53]. However, these approaches manipulate images and losses in the spatial domain and did not explicitly consider high-frequency details of objects.

2.2. Conditional Image Generation

GAN [11] is one of the widely used frameworks for image generation. GAN mainly consists of two components, a generator and a discriminator. They are learned through the adversarial loss to bring the distribution of a generated image closer to that of a real one. Diverse methods [12, 19, 29] are suggested to train the GAN framework better, and many applications have been proposed. In addition to taking a random latent vector as an input, conditional information like texts [14, 42, 52], semantic maps [31], and sketches can be fed to the network to confine the scope of the output image. Also, Isola *et al*. [16] suggested a network trained with a dataset where input and its ground truth are paired for explicitly mapping data from one domain to the other. Furthermore, CycleGAN [54] consists of two GAN architectures learned in pairs. The one translates from a source domain to the target, and the other operates in the opposite direction. Huang *et al*. [15] proposed an image-to-image translation network capable of working in multimodality.

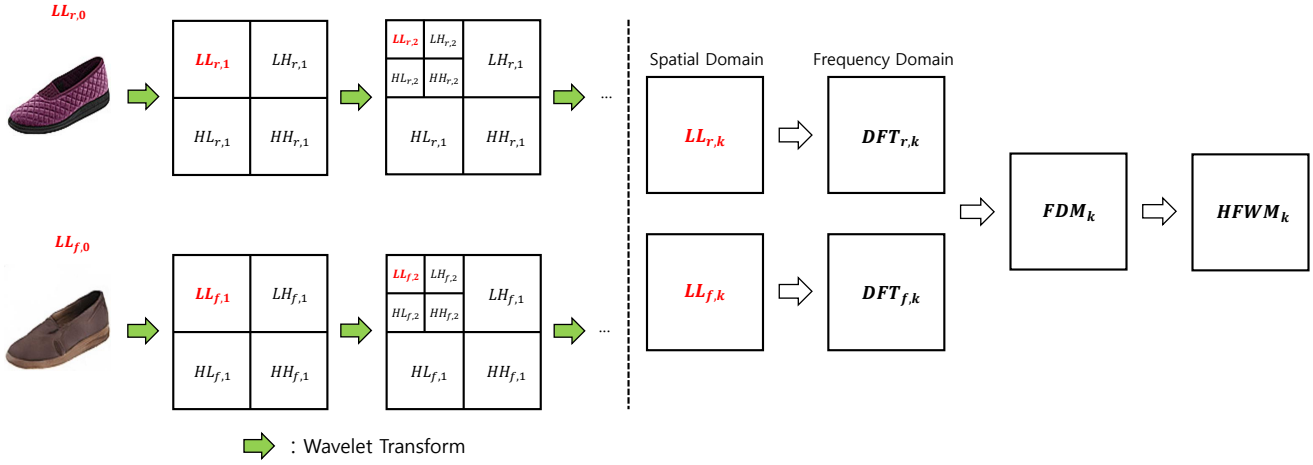

Figure 1. Overview of *WHFL*: (a) denotes multi-scale decomposition by wavelet (Sec. 3.3), where the second subscipts in each box $(1, 2)$ denote decomposition level (image scale). (b) shows a process for generating frequency distance matrix (*FDM*) and high-frequency weight matrix (*HFWM*) (Sec. 3.2) required for the calculation of our loss *WHFL*, for each scale k (Sec. 3.4). Subscript r and f denote real and fake image, respectively. Abbreviation *LL* corresponds to the approximation component obtained from the wavelet transform (Figure 4).

In [20], an attention module is also used to guide the network to focus on the important parts of the source domain. However, these suggestions also did not explicitly consider high-frequency manipulation.

3. Method

Objective functions in the spatial domain (*e.g.*, L1, L2 loss) make the network learn low-frequency components better than the high [16, 33]. Hence, we design a loss function in the frequency domain, which is adaptive to the frequency magnitudes of input images.

3.1. Revisiting Focal Frequency Loss

Jiang *et al.* [17] proposed FFL to reduce the frequency gap between real and generated images through a generative model. For calculating the loss, real images (subscript r) and fake ones (subscript f) are transformed to the frequency domain, which are denoted as

$$F_r(u,v) = a_r(u,v) + jb_r(u,v), \tag{1}$$

$$F_f(u,v) = a_f(u,v) + jb_f(u,v), \tag{2}$$

where (u,v) means a spectral position, $F_r(u,v)$ is the DFT of a real image, and a_r and b_r are real and imaginary parts of $F_r(u,v)$, respectively, where the position (u,v) is omitted when confusion is not likely. Also, $F_f(u,v)$, a_f, and b_f are similarly defined for the fake image.

Each of frequency values from the above equations is mapped to a point on Euclidean-space having real and imaginary value as a coordinate:

$$\overrightarrow{p_i} = (a_i, b_i), \quad i = r, f. \tag{3}$$

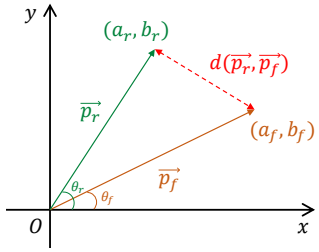

Figure 2. Frequency distance represented on Euclidean-space [17]. If we want to penalize the distance, both magnitudes and phases (θ_r, θ_f) need to be considered in a loss function.

Then, the frequency distance between mapped points on Euclidean-space is defined as:

$$d(\overrightarrow{p_r}, \overrightarrow{p_f}) = ||\overrightarrow{p_r} - \overrightarrow{p_f}||_2^2 = |F_r(u,v) - F_f(u,v)|^2, \tag{4}$$

which is the element of the frequency distance matrix (FDM) at the spectral position of (u,v).

Additionally, a weight matrix was proposed to impose more weight on the spectral positions that is hard to learn for the network. Under the assumption that a hard spectral position has a larger frequency distance, the element of the matrix is defined as:

$$w(u,v) = |F_r(u,v) - F_f(u,v)|^\alpha, \tag{5}$$

where α controls the degree of change in weights. Therefore, the loss can be finally written as:

$$FFL = \frac{1}{HW} \sum_{u=0}^{H-1} \sum_{v=0}^{W-1} w(u,v) |F_r(u,v) - F_f(u,v)|^2, \tag{6}$$

where H, W denote height, width of an image, respectively.

3.2. Weight Matrix Focusing on High-Frequency

To measure the difference between the frequency values of generated and actual images, we adopt *frequency distance* in Equation 4, which considers both magnitude and phase. However, it can be seen that the difference is naively reflected when defining the matrix of Equation 5, while it is well known that the dynamic range of frequency values is very large. Specifically, the high-frequency magnitudes are usually much smaller than DC and low-frequency magnitudes by an order or more. Hence, if the difference in each band is equally treated in a loss function, a difference in higher bands cannot well be reflected in the overall loss. Therefore, the difference should be weighted by orders of magnitudes so that the difference in higher bands can influence the overall loss, which is not considered in the conventional FFL as in Equation 5.

Based on the observations, we suggest a new weight matrix stressing the high-frequency band and name it as *High-Frequency Weight Matrix* (*HFWM*). Unlike previous methods, which design a hand-crafted binary mask based on spectral energy, our proposed weight matrix is adjusted according to the *scale* of frequency. Precisely, we apply the log function to the frequency domain difference. Based on the outputs of the log function, we call the domain having negative values as *low-scale section* (*i.e.*, high-frequency band) and the other section with positive values as *high-scale section* (*i.e.*, low-frequency band). To give larger weights to the distances in the lower scales, the absolute value of the logarithm is used for defining the weight:

$$w_0(u,v) = |log_{10}(|F_r(u,v) - F_f(u,v)|)|^{\alpha}, \qquad (7)$$

where a weight control factor α adjusts the degree of changes in each section, similar to Equation 5. Afterward, the matrix values are divided by the maximum value for normalization as:

$$w_n(u,v) = w_0(u,v)/max(w_0(u,v)). \qquad (8)$$

Subsequently, we can make the matrix have a high value at the high-frequency band as shown in Figure 3(b), while the conventional method gives very small weights to high frequencies as in Figure 3(a). However, the figure also shows that there still remains a problem with the above definition. Although the weight matrix elements for high frequencies are enlarged, the low frequencies are still given large weights due to their inherently large scales, as demonstrated in the third graph of Figure 3(b). To prevent this, we enforce the weights for the *high-scale section* to be zero. As a result, the weights of the part change to zero, but other

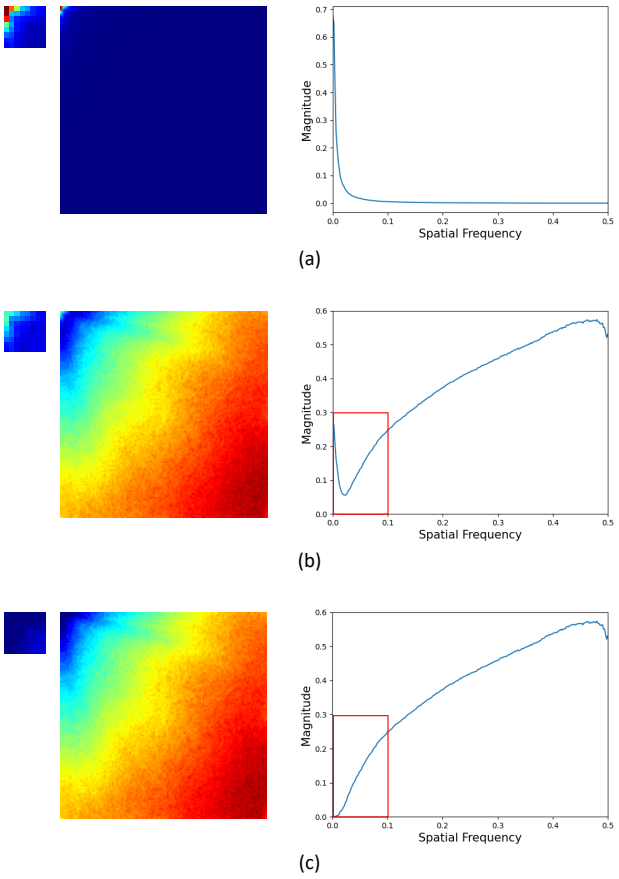

Figure 3. (a) shows a weight matrix calculated by Equation 5 from [17]. (b) displays a matrix derived from our method in Equation 8, and (c) demonstrates our final matrix after applying Equation 9. For each row, the figure in the middle visualizes the second quadrant of the weight matrix (the upper left corner is $(0,0)$ and the lower right corner is (π,π)), and the one on the left shows an enlarged view of the low-frequency part. Moreover, the graph on the right plots the magnitudes of the weights according to the frequency along the diagonal direction (0.5 corresponds to π). For visualization, we use a *jet colormap* where the red color indicates a higher value and the blue one means the opposite. As shown in the red boxes of (b) and (c), weights of (c) become smaller than those of (b) at low frequencies. Note that the weight matrices above are calculated on multiple samples from the training dataset of *ShoeV2* [49]. By averaging the matrices, we can investigate the tendency to where the weight matrix focuses. The weight matrices for individual samples are displayed in Section G of the *supplementary material*.

weights keep almost the previous value (Figure 3(c)). Formally, our proposed weight matrix is finally defined with one more step to the above equation as:

$$w(u,v) = \begin{cases} 0 & \text{at high-scale section} \\ w_n(u,v) & \text{elsewhere.} \end{cases} \qquad (9)$$

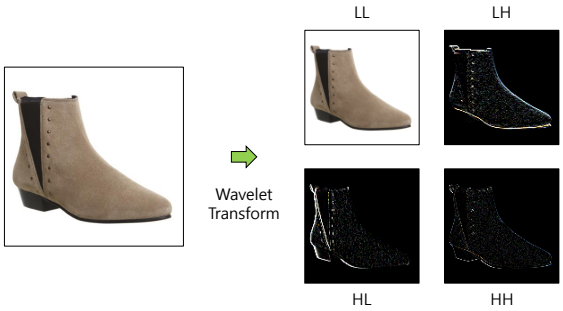

Figure 4. Demonstration of the wavelet transform, which decomposes an image into an approximation (LL) and details (LH, HL, HH).

3.3. Frequency Focal Loss Applied to Multi-scale Images Decomposed by Wavelet

For refining the frequency-domain loss in a multi-scale manner, we adopt the wavelet transform to decompose the input into multi-scale subbands. Specifically, instead of the conventional FFL in Equation 6, we split the frequency regions and use different FDM (Equation 4) and weight matrix (Equation 9) for each of different frequency bands. In order to transform an image to the wavelet domain, 2D Discrete Wavelet Transform (DWT) is performed, which splits an image into four sub-images, as illustrated in Figure 1. One contains low-frequency components in both horizontal and vertical directions, called approximation (*LL*). The others, termed as details, have high frequency of horizontal (*LH*), vertical (*HL*), and diagonal components (*HH*) (Figure 4). When the wavelet decomposition is repeated for k times, then the lowest bands are denoted as LL_k, LH_k, HL_k, and HH_k. If the dimension of an image is $H \times W \times 3$, sub-signals have $H/2^k \times W/2^k \times 3$ dimension at level k. Then, for each of the k-level approximations LL_k, we obtain the FDM_k and $HFWM_k$ as in Figure 1, which will be used for our loss.

3.4. The Final Formula for WHFL

Based on the explanation and notations in the above subsection, the frequency loss for the k-level frequency band is defined as

$$L_k = \frac{1}{H_k W_k} \sum_{u=0}^{H_k-1} \sum_{v=0}^{W_k-1} FDM_k(u,v) \cdot HFWM_k(u,v), \tag{10}$$

where H_k, W_k denote the height and width of approximation at each wavelet decomposition level k. Finally, our loss is the sum of losses from all decomposition levels:

$$L_{WHFL} = \sum_{k=0}^{d} L_k. \tag{11}$$

If the level is zero, the symbols are related to data to which wavelet transform is not applied. Also, d specifies the maximum level.

We can apply *WHFL* as a complementary loss to the losses defined in the spatial domain (*e.g.*, L1 or L2 loss) as:

$$L_{total} = L_{spatial} + \lambda \cdot L_{WHFL}, \tag{12}$$

where λ indicates a hyper-parameter which adjusts the balance between two loss terms.

4. Experiments

4.1. Settings

Experiments are divided into two categories depending on whether a ground truth is given as a photo corresponding to an input sketch or not. If the input and ground truth are paired, we term it as a *paired case* and as an *unpaired case* if not. For the *paired case*, we adopt Pix2Pix [16] as a baseline. We select CycleGAN [54] and MUNIT [15] for the *unpaired case*.

To train the Pix2Pix, we choose *edges2shoes* [48], which provides a set of photos and sketches whose boundaries are aligned to the related photo. For the *unpaired case*, we utilize *ShoeV2* [49] used in fine-grained sketch to image retrieval (FG-SBIR) [2,39,50]. The dataset consists of photos and free-hand sketches which depict a corresponding photo in several ways. We set *Haar wavelet* as a wavelet basis, wavelet decomposition level as 1 or 2, and $\alpha = 1$ in Equation 7 (see Section C of *supplementary file* for ablation studies about α).

4.2. Results

Figure 5 shows qualitative results in the spatial domain. As shown in the outcomes of Pix2Pix, there are artifacts in the texture of shoes' front without *WHFL*. We expect that *WHFL* reduces the risk of generating large artifacts in boundaries. Furthermore, with *WHFL*, it can be observed that outermost borders like a sole becomes more distinct in the case of using CycleGAN. In a similar context, we can find in MUNIT that the details of sketches, such as shoe laces, appear more clearly by our method.

Moreover, we evaluate quantitative results with Frechet Inception Score (FID) [13] and Inception Score (IS) [37] for each baseline. Lower FID is better because it means that the statistics of generated images are closer to those of real ones. Also, higher IS indicates superior results owing to the quality and diversity of images. We check whether our *WHFL* improves the metrics for each baseline. Table 1 lists the comparisons, which shows that our method provides better results in most cases. More quantitative results, including loss comparisons and ablation studies about which

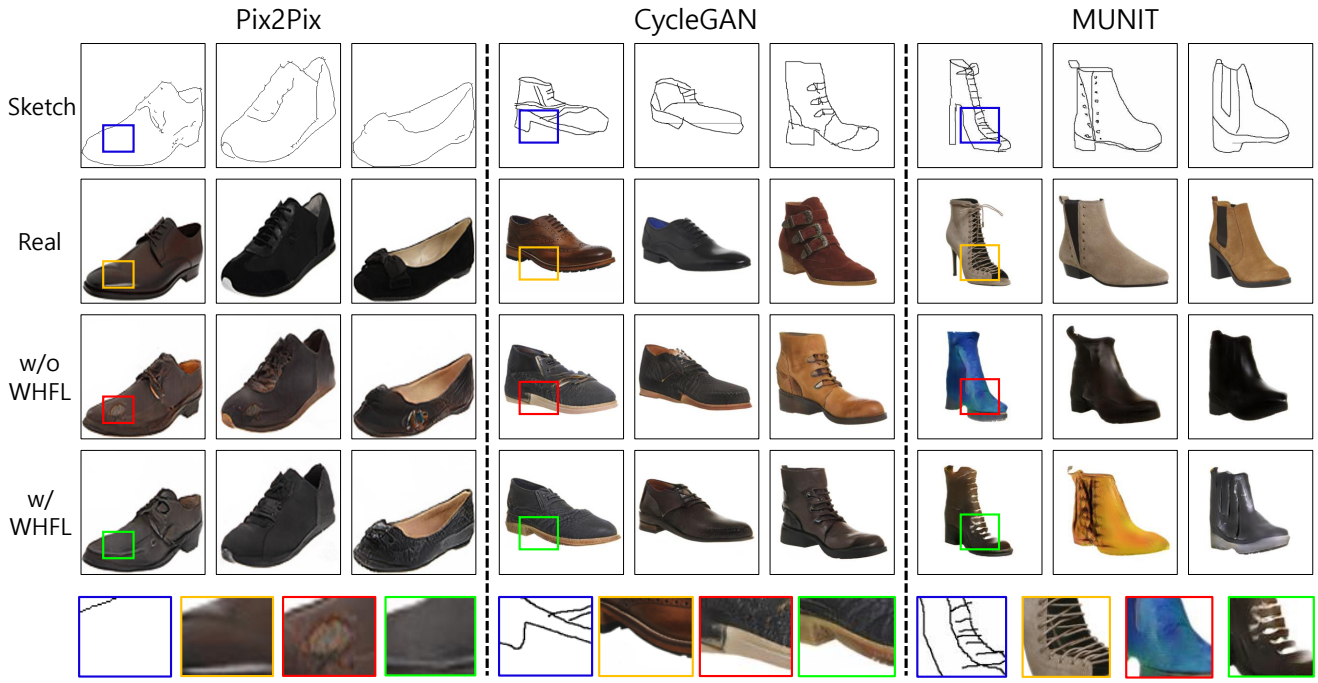

Figure 5. Qualitative results for both *paired* (Pix2Pix) and *unpaired* (CycleGAN, MUNIT) cases. The first row shows input sketches, and the second displays corresponding real photos (ground truths). The third row indicates the generated photos when *WHFL* is not used, and the fourth exhibits the outcomes when the loss is used additionally. The last row shows enlarged images for the areas marked with the corresponding color in the first column. More examples are provided in Section G of the *supplementary material*.

Table 1. Quantitative results calculated by FID and IS.

Network	Loss	FID ↓	IS ↑
Pix2Pix	w/o WHFL	63.580	2.505±0.175
	w/ WHFL	**61.744**	**2.622±0.202**
CycleGAN	w/o WHFL	60.138	**2.787±0.354**
	w/ WHFL	**56.354**	2.756±0.300
MUNIT	w/o WHFL	110.252	2.797±0.278
	w/ WHFL	**102.203**	**2.925±0.344**

component contributes more, are provided in Section B of the *supplementary material*.

5. Analysis

We propose a new evaluation metric in this section, which is suitable for measuring the difference of generated images in high-frequency bands. Also, we investigate the effect of each component suggested in Section 3.2 and Section 3.3 by performing ablation studies using a dataset *ShoeV2* in terms of frequency.

5.1. Unsigned Euclidean Distance Field Error

As stated previously, Chen *et al.*'s method [7] is the first to adopt a GAN for sketch-to-photo generation. They suggested using the unsigned Euclidean distance field (*UEDF*) to calculate a dense representation for an input. Since similar images are clustered in the field, the difference in this field can be a measure for evaluating the synthesized images, and we define an evaluation metric called Unsigned Euclidean Distance Field Error (*UEDFE*). To calculate the error, we transform an edge-map (Figure 6(a)) to the *UEDF* (Figure 6(b)). In this field, each pixel value indicates the shortest distance to the non-zero pixel in the edge map. Thus, all pixels of *UEDF* contain information about edges. The pixel's intensity of *UEDF* can be written as:

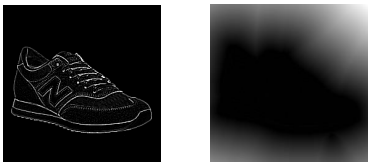

Figure 6. (a) shows an edge-map whose non-zero pixel represents an edge. (b) exhibits an *UEDF* calculated from the edge-map. In *UEDF*, pixels farther from edges have a higher intensity.

$$I_{UEDF}(p_i) = min_{p_j} d_e(p_i, p_j) \; for \; I(p_j) \neq 0, \quad (13)$$

where I means the edge-map, I_{UEDF} denotes *UEDF* of I, p_i indicates a pixel coordinate, and d_e is the Euclidean distance between the pixels. Then, we normalize intensities

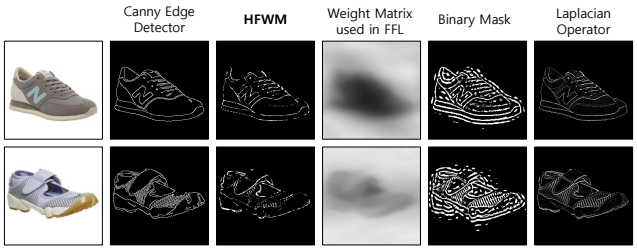

Figure 7. Results by applying Canny edge detector, *HFWM*, weight matrix used in FFL [17], binary mask, and Laplacian operator to the samples in the first column. More visualizations can be found in Section G of the *supplementary file*.

Table 2. *UEDFE* calculated for (a) *HFWM*, (b) binary mask, and (c) Laplacian operator by setting the outcome of the Canny edge detector as the reference. We experiment with four settings for objectivity according to low and high threshold values; (1) 50, 200 (2) 50, 250 (3) 100, 200 (4) 100, 250.

$\times 10^{-4}$	(1)	(2)	(3)	(4)	*Average*
(a)	40	41	39	41	40.25
(b)	101	103	103	108	103.75
(c)	29	31	33	37	**32.50**

to [0,1]. Finally, *UEDFE* can be represented as:

$$UEDFE = \frac{1}{|P|}\sum_{p\in P}(\hat{I}_{UEDF}(p) - \bar{I}_{UEDF}(p))^2, \quad (14)$$

where $\hat{I}_{UEDF}$ and $\bar{I}_{UEDF}$ mean *UEDF*s converted from the edge-map and the reference, respectively. Besides, P specifies a set of pixels making up the *UEDF*s. A lower *UEDFE* is better because it means the two edge maps are similar.

5.2. The Effect of High-Frequency Weight Matrix

In this section, we investigate the property of *HFWM* defined in Section 3.2 using edge maps. As a reference to confirm whether *HFWM* focuses on high frequencies, we select the Canny edge detector that is the most basic method for extracting the frequencies (*i.e.*, edges). Also, we compare our matrix with the weight matrix in FFL [17], binary mask, and Laplacian operator. The binary mask in this experiment is obtained by thresholding, specifically the frequencies are masked when the Euclidean distance between its spectral position and DC signal is less than 20, as denoted by the blue area in the third image of Figure 8(a).

By applying the masks or weights in these methods, we obtain the results shown in Figure 7. In the case of the weights of FFL [17], we can see that the silhouette of an object, which contains low frequencies, remains rather than the edges. On the other hand, edges are left in the results of *HFWM*, similar to the Laplacian operator. The binary mask also extracts edge information, but the precision of extraction is inferior to *HFWM*.

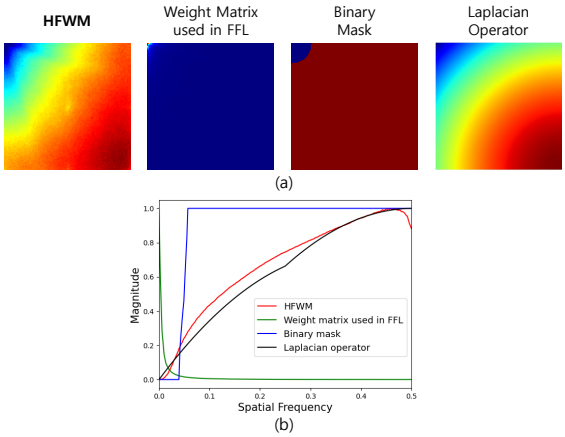

Figure 8. (a) visualizes the second quadrant of *HFWM*, weight matrix used in FFL [17], and binary mask, and Laplacian operator (the upper left corner is (0,0) and the lower right corner is (π, π)). (b) plots the averaged magnitudes of elements along the diagonal direction. Note that all graphs in (b) are normalized to [0,1].

To quantitatively compare the performance of *HFWM*, the binary mask, and Laplacian operator for edge extraction, we calculate *UEDFE* (Equation 14) for the outputs based on the references (*i.e.*, the difference from Canny edge detector in terms of the distance in the field). Table 2 shows that the averaged error between *HFWM* and the references is smaller than that of the binary mask. The error from the Laplacian operator has the minimum value and is closer to that of *HFWM* than the binary mask. Moreover, this is consistent with the observations in Figure 7.

We infer the reason for the above phenomena by visualizing the matrices (*HFWM*, weight matrix in FFL [17], binary mask, and Laplacian operator) (Figure 8(a)), specifically by plotting the averaged magnitudes of elements along the diagonal direction (Figure 8(b)). The matrix in FFL [17] leaves the silhouette from the image as its weights are extremely biased to the low frequency. Since not possible to assign different weights according to each frequency, the binary mask could not extract the edge information as precisely as the other two matrices. Besides, we analyze the factor that makes *UEDFE* of the Laplacian operator superior in terms of design similarity between the Canny edge detector and the Laplacian operator, in that their kernels are based on the derivatives. However, *HFWM* has dynamic and adaptive properties in contrast to the operator and could dynamically impose weights on the frequencies. Hence, *HFWM* has positive effects on the training process and performance. More details about the properties and effects of *HFWM* are presented in Section A of *supplementary file*.

In summary, from the above experiments, we can confirm that *HFWM* concentrates on the part of high frequency while having advantages in dynamic and adaptive characteristics compared to the binary mask and Laplacian operator.

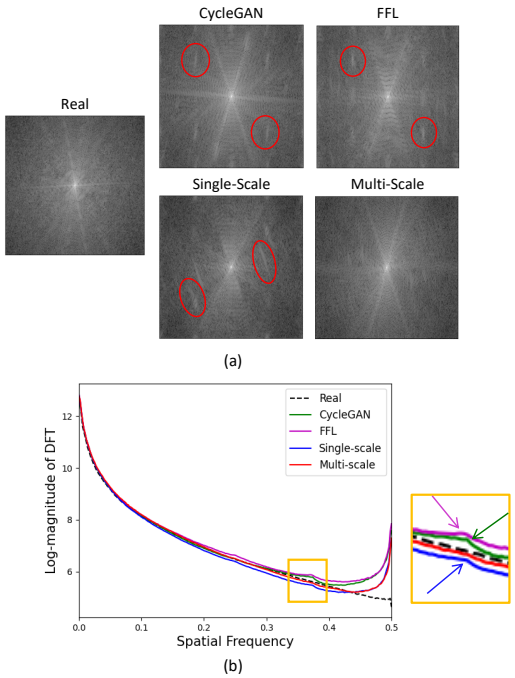

Figure 9. (a) shows the log-magnitudes of DFT using $\log(\cdot)$ in gray scale. Some artifacts are marked by red ellipses. (b) plots the averaged log-magnitudes of DFT along diagonal direction from $(0, 0)$ to (π, π), and the figure on the right is an enlarged view of the yellow box in the left plot. Also, the sharply bent peaks in the graph are indicated by arrows colored corresponding to each case.

5.3. The Effect of Multi-Scale Decomposition Scheme

To analyze the influence of the multi-scale framework in constructing the loss function, we investigate the log-magnitudes of DFT for the generated images. A non-multi-scale method, which does not use the wavelet transform, is denoted as the single-scale method. As illustrated in Figure 9(a), the repetitive artifacts appear in the CycleGAN baseline result, where the repeated blobs are related to the perceptual degradation of the naturalness [1]. Although the blobs remain when FFL [17] or the single-scale method is used, they almost disappear in the multi-scale case. The improvement can be observed in the graph plotted with the averaged log-magnitudes of DFT along the diagonal direction. As indicated in Figure 9(b), the sharply bent peak, which does not emerge in the real case, appears at a high frequency in the cases of CycleGAN baseline, FFL [17], and single-scale method, but is most alleviated with the multi-scale method.

6. Limitation & Future Work

Experiments show that *WHFL* can reduce the occurrence of large artifacts in boundary lines and keep details of sketch inputs in the generated photos, as mentioned in Section 4.2. However, there are also some samples without significant improvement in texture generation. We conjecture that *WHFL* does not significantly generate textures in flat areas because the human visual system is sensitive to the artifacts in low-frequency regions. The limitations are displayed in Section G of the *supplementary file*.

We have explored the possibility that the *WHFL* can also be used for image restoration tasks. We attempt to find out whether *WHFL* focuses on the high frequency in these tasks, where we select a deblurring problem [30] as an example. We estimate *HFWM* using the network output and ground truth. Then we apply the matrix to the ground truth for visualization, as shown in Figure 10. The result masked by *HFWM* is comparable to the outcome from a binary mask, especially for people areas in the scene. Therefore, we can confirm that the high frequency can be sufficiently extracted from natural images through *WHFL*, and we will apply the function to the restoration tasks as future work.

Figure 10. We inspect the applicability of *WHFL* through image deblurring task [30] with GOPRO dataset. (a) Ground truth, (b) Ground truth masked by *HFWM*, and (c) Ground truth masked by a binary mask.

7. Conclusion

We have proposed a new loss function named *WHFL* to improve the quality of results from sketch-to-image translation networks. The function is formulated in a multi-scale manner by using wavelet transform so that it can more finely control the weights for high-frequency bands. By applying the function to GAN-based image generation models, we could overcome the tendency that networks learn a bias towards low frequency. Also, unlike previous methods using hand-crafted binary masks based on spectral energy for weighting the frequency bands differently, *WHFL* uses an adaptive and scale-based weight matrix. Hence, the function can dynamically focus on the high frequency, and we confirmed its performance by several image quality metrics and a new metric *UEDFE*.

Acknowledgements. This work was supported in part by the Technology Innovation Program (ATC+ program, 20014131, 25nm X-ray Inspection System for Semiconductor Backend Process) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea), and in part by the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2022.

References

[1] Yong Bai, Yuanfang Guo, Jinjie Wei, Lin Lu, Rui Wang, and Yunhong Wang. Fake generated painting detection via frequency analysis. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1256–1260. IEEE, 2020.

[2] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. More photos are all you need: Semi-supervised learning for fine-grained sketch based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4247–4256, 2021.

[3] Ayan Kumar Bhunia, Subhadeep Koley, Abdullah Faiz Ur Rahman Khilji, Aneeshan Sain, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketching without worrying: Noise-tolerant sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 999–1008, 2022.

[4] Ayan Kumar Bhunia, Aneeshan Sain, Parth Shah, Animesh Gupta, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Adaptive fine-grained sketch-based image retrieval. *arXiv e-prints*, pages arXiv–2207, 2022.

[5] Mu Cai, Hong Zhang, Huijuan Huang, Qichuan Geng, Yixuan Li, and Gao Huang. Frequency domain image translation: More photo-realistic, better identity-preserving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13930–13940, 2021.

[6] Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. Sketch2photo: Internet image montage. *ACM transactions on graphics (TOG)*, 28(5):1–10, 2009.

[7] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018.

[8] Mathias Eitz, Ronald Richter, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Photosketcher: interactive sketch-based image synthesis. *IEEE Computer Graphics and Applications*, 31(6):56–66, 2011.

[9] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5174–5183, 2020.

[10] Arnab Ghosh, Richard Zhang, Puneet K Dokania, Oliver Wang, Alexei A Efros, Philip HS Torr, and Eli Shechtman. Interactive sketch & fill: Multiclass sketch-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1171–1180, 2019.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[14] Seunghoon Hong, Dingdong Yang, Jongwook Choi, and Honglak Lee. Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7986–7994, 2018.

[15] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[17] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13919–13929, 2021.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[20] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwanghee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. *arXiv preprint arXiv:1907.10830*, 2019.

[21] Junsoo Lee, Eungyeup Kim, Yunsung Lee, Dongjun Kim, Jaehyuk Chang, and Jaegul Choo. Reference-based sketch image colorization using augmented-self reference and dense semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5801–5810, 2020.

[22] Hanhui Li, Xudong Jiang, Boliang Guan, Ruomei Wang, and Nadia Magnenat Thalmann. Multistage spatio-temporal networks for robust sketch recognition. *IEEE Transactions on Image Processing*, 31:2683–2694, 2022.

[23] Luying Li, Junshu Tang, Zhiwen Shao, Xin Tan, and Lizhuang Ma. Sketch-to-photo face generation based on semantic consistency preserving and similar connected component refinement. *The Visual Computer*, pages 1–18, 2021.

[24] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations*, 2020.

[25] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Self-supervised sketch-to-image synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2073–2081, 2021.

[26] Runtao Liu, Qian Yu, and Stella X Yu. Unsupervised sketch to photo synthesis. In *European Conference on Computer Vision*, pages 36–52. Springer, 2020.

[27] Yongyi Lu, Shangzhe Wu, Yu-Wing Tai, and Chi-Keung Tang. Image generation from sketch constraint using contextual gan. In *Proceedings of the European conference on computer vision (ECCV)*, pages 205–220, 2018.

[28] Salma Abdel Magid, Yulun Zhang, Donglai Wei, Won-Dong Jang, Zudi Lin, Yun Fu, and Hanspeter Pfister. Dynamic high-pass filtering and multi-spectral attention for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4288–4297, 2021.

[29] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[30] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[32] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[33] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[34] Antônio H Ribeiro and Thomas B Schön. How convolutional neural networks deal with aliasing. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2755–2759. IEEE, 2021.

[35] Aneeshan Sain, Ayan Kumar Bhunia, Vaishnav Potlapalli, Pinaki Nath Chowdhury, Tao Xiang, and Yi-Zhe Song. Sketch3t: Test-time training for zero-shot sbir. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7462–7471, 2022.

[36] Aneeshan Sain, Ayan Kumar Bhunia, Yongxin Yang, Tao Xiang, and Yi-Zhe Song. Stylemeup: Towards style-agnostic sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8504–8513, 2021.

[37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[38] Katja Schwarz, Yiyi Liao, and Andreas Geiger. On the frequency bias of generative models. *Advances in Neural Information Processing Systems*, 34, 2021.

[39] Jifei Song, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Xiang Ruan. Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In *BMVC*, volume 1, page 3, 2016.

[40] Jialin Tian, Xing Xu, Fumin Shen, Yang Yang, and Heng Tao Shen. Tvt: Three-way vision transformer through multi-modal hypersphere learning for zero-shot sketch-based image retrieval. 2022.

[41] Shukai Wu, Weiming Liu, Qingqin Wang, Sanyuan Zhang, Zhenjie Hong, and Shuchang Xu. Reffacenet: Reference-based face image generation from line art drawings. *Neurocomputing*, 488:154–167, 2022.

[42] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2256–2265, 2021.

[43] Xiaoyu Xiang, Ding Liu, Xiao Yang, Yiheng Zhu, Xiaohui Shen, and Jan P Allebach. Adversarial open domain adaptation for sketch-to-photo synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1434–1444, 2022.

[44] Wenbin Xie, Dehua Song, Chang Xu, Chunjing Xu, Hui Zhang, and Yunhe Wang. Learning frequency-aware dynamic network for efficient super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4308–4317, 2021.

[45] Peng Xu, Chaitanya K Joshi, and Xavier Bresson. Multigraph transformer for free-hand sketch recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[46] Lan Yang, Aneeshan Sain, Linpeng Li, Yonggang Qi, Honggang Zhang, and Yi-Zhe Song. S 3 net: Graph representational network for sketch recognition. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[47] Yan Yang, Md Zakir Hossain, Tom Gedeon, and Shafin Rahman. S2fgan: Semantically aware interactive sketch-to-face translation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1269–1278, 2022.

[48] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014.

[49] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016.

[50] Qian Yu, Jifei Song, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Fine-grained instance-level sketch-based image retrieval. *International Journal of Computer Vision*, 129(2):484–500, 2021.

[51] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural network that beats humans. *International journal of computer vision*, 122(3):411–425, 2017.

[52] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked

generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

[53] Ke Zhang, Wen-Li Huang, Peng-Cheng Wang, and Si-Bao Chen. Lsragan: Generating multifarious color photographes from sketch. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1636–1640. IEEE, 2020.

[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.