Image Inpainting using Weighted Mask Convolution

Jiwoo Kang, Seongmin Lee, Suwoong Heo, and Sanghoon Lee
Electrical and Electronic Department, Yonsei University, Seoul, Korea
E-mail: {jwkang, lseong721, heartshape, slee}@yonsei.ac.kr

***Abstract*—Despite of many efforts for handling various holes, it has been not sufficiently resolved and the instability and normalization issues exists due to the presence of the invalid pixels. We proposed the weighted convolution that balances the valid and invalid pixels throughout the networks to help the network efficiently cope with various hole shapes. In our convolution layer, the mask is utilized to store the validity of the features by using the real-valued mask. A weighted scheme for the normalization layers is also proposed to adaptively operate along with the weighted convolution. By balancing upon the invalid pixels caused by the holes and zero-paddings, the network can be trained more robust to the hole shapes. The experimental results verified that our method achieved improvements over the state-of-the-art inpainting methods.**

***Index Terms*—Image inpainting, hole completion, hole filling, weighted mask.**

I. Introduction

Image inpainting (or image completion) is a task of filling in missing regions with alternative contents. It enables to generate contents naturally on occluded regions and allows to remove undesired objects. It can be used in many applications [1]–[5] such as image-based reconstruction [6], garment prediction [7], photo editing [8], facial editing [9], and super-resolution [10], [11]. The traditional approaches for the image inpainting are diffusion-based or patch-based ones with low-level features [3], [12]–[14]. Although these approaches work well in capturing high-frequency textures such as background regions, they do not predict high-level texture semantics and not to fill the hole if the similar region surrounded by similar context is unavailable. Recently, deep neural network based inpainting approaches [15]–[18] are introduced where high-level semantics and low-level details are learned into a convolutional encoder-decoder network in an end-to-end manner.

However, as the convolution networks employ filters on images with holes that are commonly filled with the mean values of the images or random values, these approaches inevitably suffer from normalization issues. In other words, the network applies the convolutional filters by treating the valid and invalid (containing the hole) regions equivalently, leading to color inconsistencies, the lack of texture in the hole regions, or distinct edge responses surrounding the hole, especially when the holes have irregular shapes.

Partial Convolution [19] has been proposed to efficiently handle the holes. It filters only for the valid pixels and fills the holes by linearly scaling the value partially measured for the valid one(s). In the partial convolution, however, invalid pixels progressively disappear through deep layers by setting the mask to ones regardless of how many pixels are covered by the filter range in the previous layer. It causes ambiguity during training and leads to visual artifacts such as blurriness and color discrepancy, as reported in [19].

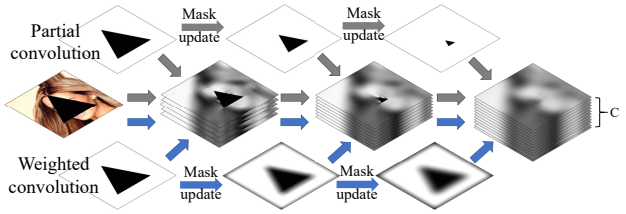


Fig. 1: The Comparison between partial convolution (up) and weighted convolution (down).

The inpainting network that utilizes the dynamic mask [20] is recently proposed where the mask, as well as feature, is learned from the input feature (pixels) of the previous layer, and the output feature pixel-wisely filtered by the mask is given to the next layer. It enables each convolution layer to learn soft masks that can select feature maps according to backgrounds, masks, and sketches. Nevertheless, it still used the input image with the hole regions remained and increased the value discrepancy between the channels by attaching the mask as an additional channel of the input image. Although the soft mask approach can help the network to train with a user-guidance, it does not sufficiently resolve the instability and normalization issues originate from the invalid pixels.

The deep learning-based image inpainting network has been composed of consecutive combinations of convolution and activation layers. Based on the fact that it is a series of processes for generating (filtering) and masking (activating) of feature, we proposed the weighted convolution that balances the valid and invalid pixels throughout the networks to help the network efficiently cope with various hole shapes. During the convolution with the image of the hole regions, the invalid pixels not only come from the hole regions in the sliding window but the zero paddings in the sliding window. The partial convolution efficiently complements the insufficient data from other valid pixels in the sliding window.

However, the partial convolution does not take account of the number of valid pixels. It treats the features equivalently in the following layers regardless of the validity of the features. On the baseline of the partial convolution, the real-valued mask is utilized to balance the features by the validity. The degree of the validity is updated into the mask in the weighted convolution, as described in Fig. 1. The real-valued mask enables the features to be filtered relatively in the sliding

windows by complements invalid pixels caused by the holes and paddings. As features measured from valid pixels are more preferred, the convolution operator itself involves the spatially discounted reward, which showed effective for improving the visual quality.

We also introduce a weighted scheme for the normalization layers to adaptively operate along with the weighted convolution. As addressed in [19], it is difficult to apply the previous normalization techniques such as batch normalization generally in the inpainting networks due to the presence of holes. On this account, in [19], the means and variances of the batch normalization layers in the encoder were obtained at the coarse-tuning stage and then frozen at the fine-tuning stage. In [17], [20], the normalizations were not used for both the encoder and the decoder. By balancing the features by regarding the degree of the invalidity due to the holes and the paddings, the weighted convolution layer and the weighted normalization method help to make the network agnostic to the various hole shapes.

The main contributions are summarized as follows:

- We introduce weighted convolution to balance invalidity due to both the hole regions on an image and zero-paddings in the convolution layers, enabling to operate the network robust to various hole shapes;
- We propose a weighted normalization method to operate along with the weighted convolution, efficiently resolving the inconsistency due to the presence of the holes;
- Our weighted convolution method achieves the significant improvements on the base of the partial convolution as well as the state-of-the-art inpainting methods.

II. Approach

A. Partial Convolution

We briefly summarize partial convolution recently proposed in [19] first and explain the ambiguity due to the mask propagation. Let $\mathbf{W}$ and b are the convolution filter and the corresponding bias. For given pixels $\mathbf{X_s}$ and the corresponding binary mask $\mathbf{M_s}$ in a sliding window of the partial convolution operation at every location, the output feature located at (y, x) is computed as:

$$I_{y,x} = \begin{cases} \mathbf{W}^T \left(\mathbf{X} \odot \frac{\mathrm{sum}(\mathbf{1})}{\mathrm{sum}(\mathbf{M_s})} \right), & \text{if } \mathrm{sum}\left(\mathbf{M_s}\right) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\odot$ denotes element-wise multiplication and $\mathbf{1}$ is a tensor of the same shape as $\mathbf{M_s}$ and has all the values of ones. The mask is updated to be valid if at least a valid input value is covered by the convolution as:

$$m_{y,x} = \begin{cases} 1 & \text{if } \mathrm{sum}\left(\mathbf{M_s}\right) > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The rule of the mask update in (2) and the normalization in (1) make only one valid pixel is treated as the same amount as a filter size of the convolution layer. In other words, when W, H, and C denote width, height, and channel of the filter, a pixel value can be used to estimate the feature on behalf of $W \times H \times C$ pixel values maximally. As the partial convolution layers handle infeasible and feasible features caused by the hole regions equivalently throughout the layers of the convolutional network, the results are much affected by the size and shape of the input masks (holes) during training the network, leading to visual artifacts, long convergence times, and high error rates.

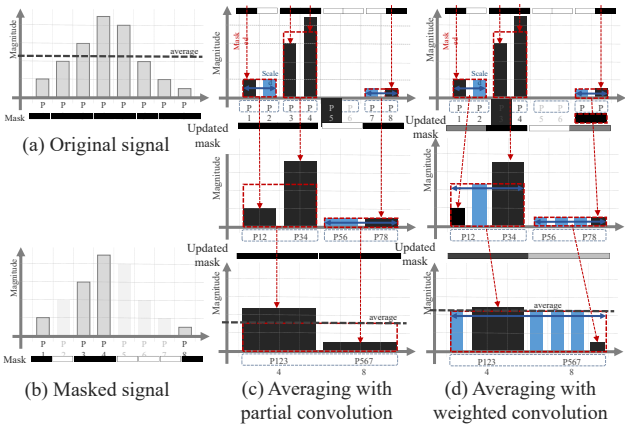


Fig. 2: An example for the comparison between the partial convolution [19] and the weighted convolution. From the signal with the half of pixels missing, the average of the signal is measured by consecutively applying the averaging filter of kernel size 2.

B. Weighted Convolution

We build upon the concept of the partial convolution, where only parts of the pixels (and parts of the convolution filter weights) are used to estimate features due to the holes in the image and the zero-paddings. To enable the network to operate regardless of the shape and size of the holes, we used the real-valued mask to contain the feasibility of the corresponding feature. In the weighted convolution, the update mask value located at (y, x) is defined as the ratio of the mask used in the convolution operation:

$$m_{y,x} = \frac{\mathrm{sum}\left(\mathbf{M_s}\right)}{\mathrm{sum}\left(\mathbf{1}\right)}. \tag{3}$$

The mask update operators help the mask to have the ratio of the valid pixels over the pixels used for the convolution operators passed through.

For the weighted convolution operation, the mathematically same formulation is used in (1), except the real-valued mask is employed. The weighted convolution inherits the major strength of the partial convolution that it substitutes the values of invalid pixels from the hole for the valid pixels. Thus, the features scaled to equivalent-degree of the validity by the convolution operation are used for following normalization and activation layers.

A significant advantage that the real-valued mask updated by the operation in (3) is that it enables the convolution operation to measure 'relatively' to the degree of the validity,

i.e., the magnitudes of the mask values in the sliding window. For example, when the values of the mask are distributed, the feature is calculated by weighted-averaging the pixels by the mask values for each sliding and then scaled (normalized) to the kernel dimension. It gives a higher weight to the feature measured from more valid pixels. By contrast, when all the values of the mask have the same feasibility, i.e., their values are the same, the weighted convolution operation in (1) works the same as a general convolution regardless of their magnitudes. Although the convolution operation is performed in a relatively weighted way for each sliding, the feasibility of the feature is recorded in the mask update step and, therefore, relativity between all the features obtained from the operation is also preserved in the following layers.

The basic assumption of the weighted convolution comes from the distribution of either the valid pixels or the invalid pixels has the similar to the distribution of the whole pixels. Figure 2 shows an example for the comparison between the partial convolution and the weighted convolution operations. For visualization, the binary mask is colored on a white for the hole and on a black for the valid pixel, and the real-valued mask is colored on gray-scales between white and black meaning values between one and zero correspondingly in the figure. Assume that a masked signal in Fig. 2b where 4 of 8 pixels are removed from an original signal in Fig. 2a. Figures 2c and 2d describes the procedures of averaging the pixels by filtering neighboring pixels non-overlappingly using the partial and weighted convolution operations, respectively. The partial convolution completely replaces invalid pixels by the mean of the valid pixels in the sliding window (marked on the blue square in the figure), and the replaced values are used as the values of the invalid pixels in the following layers. In contrast, the weighted convolution substitutes the invalid pixels by the mean of the valid pixels to measure the feature for the current layer and only the used portions of the valid pixels are passed to the next layer through the mask. As a consequence, it can be seen that the weighted convolution produces the same value to the average of the valid pixels on the masked signal.

C. Weighted Normalization Layer

To efficiently normalize the feature values measured from the image containing the holes, we used weighted normalization layers, where the mean and variance of the features are employed weighted by the mask. As the mask values quantify the degree of the validity for the corresponding feature, we made much of the features from the available pixels. The weighted normalization approach can be applied in normalization methods that utilize the mean and variance of the input features such as batch [21], layer [22], instance [23], and group [24] normalization techniques. As the weighted convolution utilizes the mask relatively in the sliding window, the normalization layer complements the weighted convolution layer by balancing the features among the channels. When denoting $\mathbf{X}$ as the feature to be normalized and $\mathbf{M}$ as the corresponding mask, the weighted mean and variance can be defined by following the statistical theory in [25], [26] as:

$$\mathrm{E}\left(\mathbf{X}\right) = \frac{\mathrm{E}\left(\mathbf{X} \odot \mathbf{M}\right)}{\mathrm{E}\left(\mathbf{M}\right)}, \tag{4}$$

$$\mathrm{Var}\left(\mathbf{X}\right) = \frac{\mathrm{E}\left(\mathbf{X}^2\right) - \mathrm{E}\left(\mathbf{M}\right)\mathrm{E}\left(\mathbf{X}\right)}{\mathrm{E}\left(\mathbf{M}\right) - \frac{\mathrm{E}(\mathbf{M}\odot\mathbf{M})}{\mathrm{E}(\mathbf{M})}} \tag{5}$$

where the scope and dimension of the mean $\mathrm{E}\left(\cdot\right)$ and variation $\mathrm{Var}\left(\cdot\right)$ depend on the normalization techniques to be applied. We simplify the equation of the variance in (5) as

$$\mathrm{Var}\left(\mathbf{X}\right) \simeq \frac{\mathrm{E}\left(\left(\mathbf{X}-\mu\right)^2 \odot \mathbf{M}\right)}{\mathrm{E}\left(\mathbf{M}\right)} \tag{6}$$

$$= \mathrm{E}\left(\mathbf{X}^2\right) - \left(\mathrm{E}\left(\mathbf{X}\right)\right)^2 \tag{7}$$

where $\mathrm{E}\left(\mathbf{X}^2\right) = \frac{\mathrm{E}((\mathbf{X}\odot\mathbf{X})\odot\mathbf{M})}{\mathrm{E}(\mathbf{M})}$. The latter term of the denominator in (5) is for deducting the sampled probability and the deduction term of the weighted variation is variously defined according the theories [25], [27]. In our test, the simplified form increases the stability to the dynamic holes as well as decreases the complexity. In our experimental section, the performance verification of the weighted normalization was conducted over the batch normalization layers.

III. Experimental Results

Training Data We employed three different public datasets for training, verifying, and testing: CelabA [28], Places2 [29], and ImageNet classification [30] datasets. For the Place2 and ImageNet datasets, the official division for the train, verification, test sets were used. For the CelebA dataset whose division was not given, we divided partitions by 7:1.5:1.5 for the train, verification, test of the model, respectively.

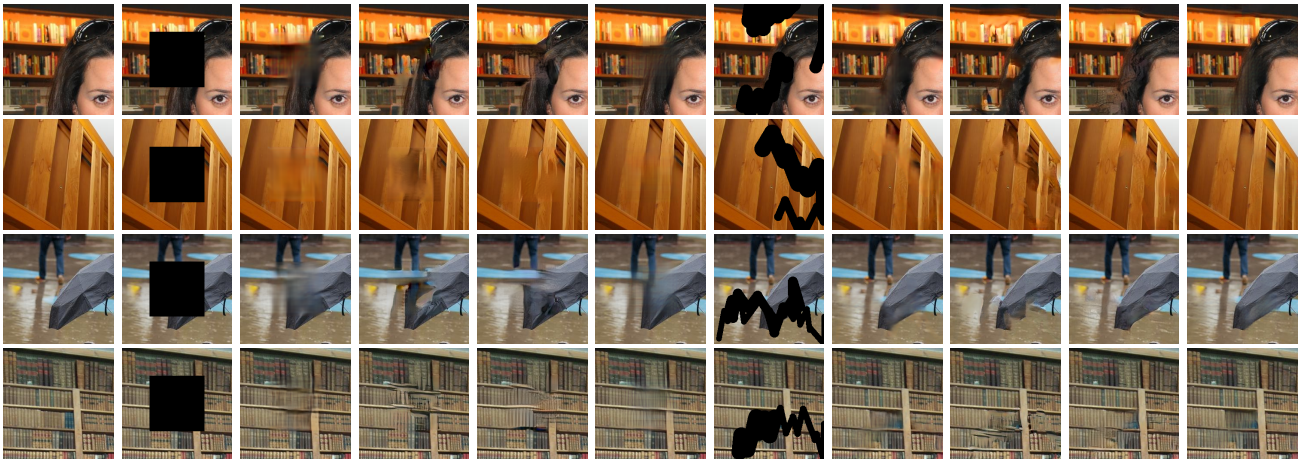
Training and Testing Procedures Ours and all the models for the comparisons are trained with Tensorflow r1.14, CUDNNv7.3, and CUDA10.0. We used Adam [31] for the optimization. We used a learning rate of 0.0005 for the initial training and decreased gradually to 0.00001. We randomly cropped the images to be 256 x 256 without scaling to feed the networks except for the CelebA dataset. For the CelebA dataset, the images were scaled to 256 x 313 at first and randomly cropped as same to the other datasets.

Methods We compared our method with the partial convolution [19] as the baseline and the method in the most recently proposed work [20]. We used L_1 error, L_2 error, SSIM [32], PSNR, and TV loss [10] to measure numerically the quality the results followed the previous works. We used the center mask and the randomly generated free-form mask used in [20] for the training and the test. For the optimization, we used Adam [31] with $\beta 1 = 0.9$, $\beta 2 = 0.999$, and $\varepsilon = 10^{-8}$. The model was trained using a NVIDIA 2080 Ti (11GB) with a batch size of 25. For batch normalization [21], we used $\varepsilon = 10^{-8}$ with decay$= 0.999$. For the networks of WConv and PConv, we used the loss functions and hyperparameters reported in [19].

For our network, we used the U-Net network architectures and the losses in [19] and we changed only the convolution

TABLE I: Results of Measurements on Test Images of ImageNet and CelebA Datasets.

	ImageNet Dataset								CelebA Dataset							
	Center mask				Free-form mask				Center mask				Free-form mask			
	WConv	*PConv*	*DF1*	*DF2*	*WConv*	*PConv*	*DF1*	*DF2*	*WConv*	*PConv*	*DF1*	*DF2*	*WConv*	*PConv*	*DF1*	*DF2*
L_1	**7.755**	10.464	13.371	10.432	**7.006**	7.304	13.290	7.640	**7.648**	8.706	10.560	9.837	**5.643**	6.272	8.566	7.448
L_2	**2.494**	2.690	4.461	2.557	**1.572**	1.639	4.072	1.721	**1.801**	2.079	3.236	2.866	**1.176**	1.306	2.330	1.872
TV loss	**2.400**	2.534	7.645	2.652	**3.502**	3.910	9.035	3.752	2.330	**2.010**	5.633	5.214	**3.061**	2.782	5.695	5.237
SSIM	**0.854**	0.844	0.823	0.845	**0.909**	0.904	0.864	0.902	**0.885**	0.874	0.857	0.860	**0.929**	0.923	0.906	0.912
PSNR	**20.731**	17.411	15.304	16.119	**20.731**	20.347	15.279	21.119	**20.705**	19.063	17.754	18.463	**23.344**	22.126	19.564	20.872



(a) GT (b) Rect M. (c) *PConv* (d) *DF1* (e) *DF2* (f) Ours (g) Free. M. (h) *PConv* (i) *DF1* (j) *DF2* (k) Ours

Fig. 3: Qualitative comparisons on the Places2 dataset. Best viewed in zoom-in.

layers to the weighted convolution for fair comparisons. Besides, the measurements of the other state-of-the-art methods [17], [20] were also performed for the same datasets. We denote the networks with weighted and partial layers as *WConv* and *PConv*, respectively. Also, we denote the networks in [17] and [20] as *DF1* and *DF2*, respectively. For the *DF1* and *DF2*, we used the official implementations without modification and model hyper-pameters provided by the authors. [1]

Results We measured L_1, L_2, TV loss, and PSNR on the hole regions whereas SSIM was measured using the composited image whose hole pixels set to the output image from the network while the non-hole pixels set to ground-truth.

Table I summarizes the performance measurements on test images of ImageNet and CelebA datasets and some results on Place2 dataset are visualized in Fig. 3. All the models were trained with randomly generated free-form masks. We used a 128 x 128 rectangle mask for the center masks. For the free-form masks, we randomly generated the masks corresponding for each test image, and then we used the same mask for all the methods. Although the numerical methods are not sufficient to evaluate image inpainting results perceptually, it has been shown that they can be the criterion to measure the performance especially when the measurements show significant differences in many works [15]–[17], [19], [20]. Building upon the framework of the partial convolution, We did not used the discriminator in our work. As seen in the results of *DF1* and *DF2*, the discriminator made artifacts for unseen images in our test. When using perceptual term rather than the discriminator, the network tended to make a blur image rather than the artifacts. Thus, we followed the framework of the partial convolution without GAN. The weighted convolution can be extended to the general networks as it handles the paddings as well as the hole regions similar to the partial convolution.

IV. Conclusion

We proposed a weighted convolution to efficiently handle the invalidity caused by the hole regions. Built on the baseline of the partial convolution, we used the mask to store the validity of the features by using the real-valued mask. By balancing upon the invalid pixels caused by the holes and zero-paddings, the network can be trained more robust to the hole shapes. By introducing a weighted scheme for the normalization layers that operate along with the weighted convolution layer, the efficiently resolving the inconsistency due to the presence of the holes. Quantitative and qualitative comparisons over several datasets demonstrated the robustness

[1]The official implementation of the partial convolution has not been available yet. Thus, for the partial and our networks, we employed our implementation of the same framework.

of our method in image inpainting.

V. Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1A2C3011697).

REFERENCES

[1] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2007.60.

[2] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (ToG)*, vol. 28, no. 3, 24:1–24:11, 2009, ISSN: 0730-0301. DOI: 10.1145/1531326.1531330.

[4] J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, 129:1–129:10, 2014, ISSN: 0730-0301. DOI: 10.1145/2601097.2601205.

[5] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, "Video inpainting of complex scenes," *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, pp. 1993–2019, 2014.

[6] H. Song, J. Park, S. Heo, J. Kang, and S. Lee, "Patchmatch based multiview stereo with local quadric window," Association for Computing Machinery, 2020.

[7] S. Lee, W. Kim, S. Ahn, J. Kim, and S. Lee, "Physical parameter prediction by embedding human perceptual parameter for 3d garment modeling," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2019, pp. 1945–1949.

[8] A. Brock, T. Lim, J. M. Ritchie, and N. J. Weston, "Neural photo editing with introspective adversarial networks," in *5th International Conference on Learning Representations 2017*, 2017.

[9] S. Heo, H. Song, J. Kang, and S. Lee, "Local spherical harmonics for facial shape and albedo estimation," *IEEE Access*, 2020.

[10] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *The European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 694–711.

[11] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472–2481.

[12] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001. DOI: 10.1109/83.935036.

[13] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," in *The IEEE International Conference on Computer Vision (ICCV)*, 2003.

[14] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Transactions on Graphics (ToG)*, vol. 24, no. 3, pp. 795–802, 2005, ISSN: 0730-0301. DOI: 10.1145/1073204.1073263.

[15] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature learning by inpainting," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, 107:1–107:14, 2017, ISSN: 0730-0301. DOI: 10.1145/3072959.3073659.

[17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[18] J. Kang, S. Lee, and S. Lee, "Uv completion with self-referenced discrimination," *Eurographics—Short Papers*, 2020.

[19] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *The European Conference on Computer Vision (ECCV)*, 2018.

[20] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[22] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[23] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[24] Y. Wu and K. He, "Group normalization," in *The European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[25] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, F. Rossi, and R. Ulerich, *GNU scientific library (3rd ed.)* Network Theory Ltd., 2009.

[26] A. Stuart and K. Rod, *Kendalls advanced theory of statistics (6th ed.)* Wiley, 2015.

[27] A. Madansky and H. G. B. Alexander, *Weighted standard error and its impact on significance testing*. The Analytical Group, Inc, 2017.

[28] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[29] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

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

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.