FAST LEARNING A STABILIZED GENERATIVE MODEL FROM A SINGLE IMAGE

I-Te Tsai, Kai-Lung Hua
National Taiwan University of Science and Technology, Taipei, Taiwan

ABSTRACT

Generative Adversarial Networks (GANs) achieve good results through large-scale datasets. However, collecting such datasets is challenging. Therefore, enabling GAN models to learn from a few or single images without overfitting is essential. This paper proposes the dilated involution operator, which prevents channel redundancy/ learning redundant features while adapting to local information at each location. We also propose a self-supervised discriminator that doubles as a reconstruction function preventing the generated image from diverging too much. Our method achieves state-of-the-art performance with fewer parameters and half the training time.

***Index Terms*—** Unconditional Image Generation, Image Generation, Dilated Involution

1. INTRODUCTION

Image generation applications, such as unconditional and conditional image synthesis [1, 2], paired and unpaired image-to-image translation [3, 4], and image inpainting [5], benefit significantly from Generative Adversarial Networks (GANs) [6]. Through GANs, generated images have become more realistic and indistinguishable from authentic images. Although effective, most GANs are dependent on diverse large-scale datasets. Training on a large-scale dataset costs a lot of time and resources. Apart from this, obtaining a large-scale dataset is also challenging in real-world scenarios given restrictions such as privacy concerns and copyright issues. Therefore, enabling GANs to learn and perform well using a few images is an essential task.

The recently proposed ConSinGAN [7] introduces a GAN trained on a single image. It adopts a multi-stage and multi-resolution training method. The method's training involves multiple stages that slowly refine and increase the image's resolution. The generated image is upsampled at each scale until it reaches the final scale, where it will have the same aspect ratio as the original input image. Although training on a single image could lead to overfitting, they avoid this by only training the top three stages and freezing the other stages. ConSinGAN successfully performs unconditional image generation and harmonization with the help of its architecture.

Convolutions have always been an integral part of deep learning-based tasks. However, existing works [8, 9, 10, 11, 12] show that convolutions often generate features with channel redundancy. This causes the network to consume more memory and computational resources. Furthermore, convolutions use the same kernel at any spatial location to capture specific graphics. However, doing so prevents the network from adapting to different visual elements. An existing approach overcame this by proposing a new operator, "involution" [13]. The kernel weight of this operator is shared across channels, effectively preventing the calculation of redundant features. Therefore, saving computational resources and adapting to elements at different positions. Inspired by involution [13] and dilated convolution [14], we propose a framework for GANs that leverages these two. This allows the model to gain the advantages of involution [13] while also capturing more global information through dilation.

Under harsh training conditions, our GAN would also need a discriminator capable of providing useful signals to the generator. Therefore, we design our discriminator with two decoders, one for restoring the whole real image and the other for restoring a part of the image. We can guarantee that the discriminator will have more comprehensive information in terms of overall or detailed structure and texture through this design. By combining our proposed components, we successfully trained a single image generation model with significantly less time while also preserving the quality of the generated image.

To summarize, our contributions are:

- We proposed a dilated involution operator, which effectively avoids the calculation of redundant features, saving computing resources while also adapting to elements in different locations and better capturing global information.
- We propose a self-supervised discriminator with an additional decoder. This extra decoder forces the discriminator to extract image features to produce a more comprehensive signal to train the generator.
- We achieved state of the art quantitative results for unconditional generation tasks in the field of single image generation.

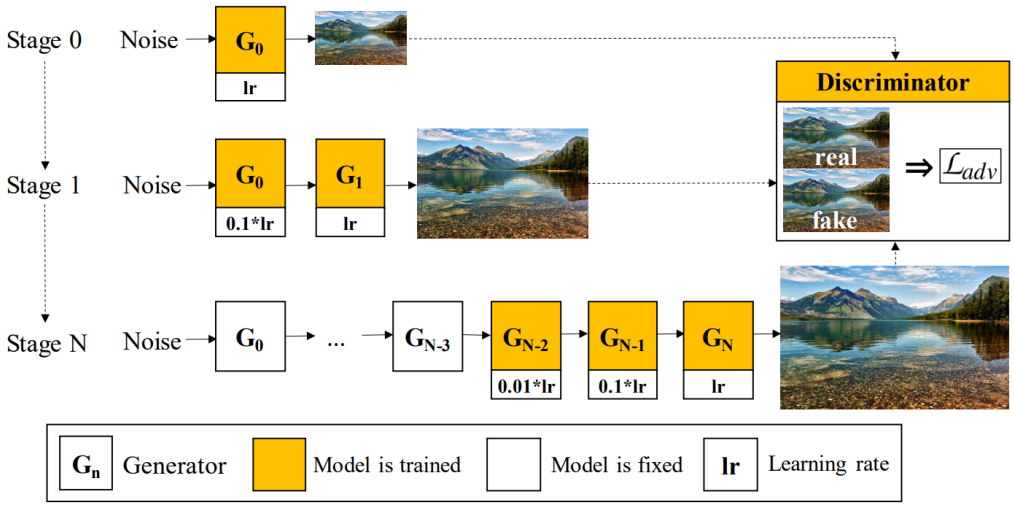

Fig. 1. **Overview of our model.**The size of our model and the resolution of the generated images will become larger and larger as the stages increase. Each G represents the layers added to each stage. In our experimental setup, we trained only one Discriminator. We trained only the highest three stages for the generator and gave the lower stages a lower learning rate (decreasing 0.1 times for each stage).

2. METHOD

This section introduces our proposed architecture for our unconditional generative model. Our architecture utilizes a multi-stage architecture and our proposed dilated involution in the generators. We also discuss our self-supervised discriminator with two decoders.

2.1. Multi-scale training

We use a multi-scale architecture to capture details at different scales. The noise added to the features allows us to generate varying images. As shown in Fig. 1, our model generates a low-resolution image from a random noise vector at stage one. When the training stage converges, we add a new layer to the model and train the next stage. After each stage, the image resolution increases. This process is repeated N times until the resolution is the same as the original image. Apart from the initial noise, we add noise at each stage to increase the diversity of the generated samples.

We utilize adversarial loss and reconstruction loss to train our GAN model

$$\min_{G_n} \max_{D_n} \mathcal{L}_{adv}(G_n, D_n) + \alpha \mathcal{L}_{G_{rec}}(G_n) \tag{1}$$

where n is the current training stage, α is the hyper-parameter, $\mathcal{L}_{adv}(G_n, D_n)$ is the WGAN-GP adversarial loss [15], and reconstruction loss is used to improve training stability. Reconstruction loss ensures that there is a fixed noise vector (z^*) that generates the original image (x_n) at stage n.

$$\mathcal{L}_{G_{rec}}(G_n) = \| G(z^*) - x_n \|_2^2 \tag{2}$$

Reconstruction provides additional information, preventing the generated image from diverging too much and maintaining its realism. This allows both G and D to be more stable.

Simultaneously training all stages can cause over-fitting. Therefore, we only train three stages at a time. We also give the lower stages a smaller learning rate, allowing them to be fine-tuned without overfitting when training the higher stages.

2.2. Dilated involution

The design structure of dilated involution is shown in Fig. 2. Let $\mathcal{H} \in \mathbb{R}^{H \times W \times K \times K \times G}$ denote the dilated involution kernels, where W and H indicate the width and height of the input and output features respectively, and C indicates the number of channels. G denotes the number of groups where each group shares the same kernel. Specifically, for each position (i, j), the corresponding kernel $\mathcal{H}_{i,j,...,g} \in \mathbb{R}^{K \times K}, g = 1, 2, ..., G$. The output feature of dilated involution $Y \in \mathbb{R}^{H \times W \times C}$ is the multiply-add operations after using the kernel on the input feature $X \in \mathbb{R}^{H \times W \times C}$. The calculation of dilated involution is written as follows:

$$Y_{i,j,k} = \sum_{(u,v) \in \delta K} \mathcal{H}_{i,j,u+[K/2],v+[K/2],[kG/C]} X_{i+(u \times d),j+(v \times d),k} \tag{3}$$

where d is the dilation size. Adjusting d allows us to capture a larger range of information with the same number of parameters.

The dilated involution kernel is generated by a single position feature vector $X_{i,j} \in \mathbb{R}^C$ through ϕ function. We define

the kernel as:

$$\mathcal{H}_{i,j} = \phi(X_{i,j}) \tag{4}$$

where $\phi : \mathbb{R}^C \mapsto \mathbb{R}^{K\times K\times G}$ denotes the kernel generation function. Here we use a simple two fully connected(FC) layers to generate the kernel ($FC - InstanceNormalize - relu - FC$).

Compared with convolution, dilated involution shares the weight across channels preventing channel redundancy. Apart from this, since the kernel is not the same at any location in space, this allows the extracted feature to be more adaptive to the content at different locations. Lastly, we can better capture the global information given the larger receptive field. This achieves faster convergence in training compared to smaller receptive fields.

When building the network with dilated involution, we add 1×1 convolution after the dilated involution to better project and fuse the channel.

2.3. Self-supervised discriminator

As the generator generates samples of different sizes according to different stages during training (see Section 2.1), the discriminator also scales the real image to the appropriate size (shown in Fig. 3).

We modify our discriminator similar to patchGAN [16]. We added two decoders at the end of the discriminator. During training, the discriminator extracts image features which the decoder then reconstructs. A simple reconstruction loss optimizes the decoder and the discriminator. The decoder is trained on real images only.

$$\mathcal{L}_{D_{rec}} = \mathbb{E}_{f\sim D(x), x\sim I_{real}}[\| \, dec(f) - \mathcal{T}(x) \, \|] \tag{5}$$

where f is the intermediate feature map from discriminator, function dec contains the processing and decoder of f, and function $\mathcal{T}$ represents the processing of sample x of real image.

Our decoder restores two kinds of images ,namely , the original image during training ($I_{downsample}$) and the second the training image after cropping (I_{crop}). We randomly crop training image with half of its height and width, and then crop the same part of the feature map before the decoder. We call the feature map of the previous layer of the decoder f_1 (for I'_{crop}) and f_2 (for $I'_{downsample}$), respectively. The decoder generates I'_{crop} from the crop f_1 and $I'_{downsample}$ from f_2. I'_{crop} corresponds to I_{crop} and $I'_{downsample}$ corresponds to $I_{downsample}$, thus minimizing the loss of eq. 5. With this design, we can ensure that the discriminator has enough detailed texture features from I'_{crop} and enough overall structure from $I'_{downsample}$.

As mentioned in Section 2.1,we employ the WGAN-GP[15] adversarial loss. Finally, our discriminator total loss is as follows:

$$\mathcal{L}_D = -\mathcal{L}_{adv} + \beta\mathcal{L}_{D_{rec}} \tag{6}$$

Table 1. Comparison with other methods on "Places".

Model	Confusion↑	SIFID ↓	Train Time	Parameters
SinGAN[17]	7.9%	0.085	65.6 min	∼1.34M
ConSinGAN[7]	4.2%	0.061	22.3 min	∼0.67M
Ours	**9.2%**	**0.046**	**9.9 min**	**∼0.21M**

Table 2. Comparison with other methods on "LSUN".

Model	Paired↑	SIFID ↓	Train Time	Parameters
SinGAN[17]	33.7%	0.23	62.1 min	∼1.03M
ConSinGAN[7]	28.7%	0.11	22.2 min	∼0.67M
Ours	**37.5%**	**0.09**	**9.8 min**	**∼0.21M**

where β is the hyper-parameter(β = 5 for all our experiments).

3. RESULTS

Our architecture's design allows it to generate images with different resolutions at test time by changing the size of the input noise vector. In this section, we compare our proposed method with state-of-the-art methods, we also show our ablation study and our model's random generation capabilities (see Fig. 4). For all our experiments, the reconstruction $alpha$ is set to 10, and we used Pytorch on an RTX 2080.

Dataset. Following ConSinGAN[7], we selected 50 images each from Places[18] and LSUN[19] dataset for evaluation. These images are composed of several categories in the Places[18] dataset and 5 from each of the 10 categories in the LSUN[19] dataset. Between the two, LSUN[19] has higher image complexity than Places[18]. We use the 50 images specified in SinGAN and ConSinGAN.

3.1. Quantitative Evaluation

Experiment setup We compared our model with single-image methods: SinGAN[17] and ConSinGAN[7] which are multi-stage methods like ours. We used their default setup for the following experiments. For each image, SinGAN[17] has 8-10 stages, while ConSinGAN[7] has 6 stages, both with 2000 iterations for each stage. For our method, the training consists of 6 stages with 500 iterations each. Fig. 5 shows the qualitative comparison between our method and theirs. It can be observed that our model is more coherent compared to others.

We trained the model for each training image in the dataset and randomly generated 100 samples. We then averaged the Single Image Frechet Inception Distance (SIFID) of all samples and corresponding real images for our evaluation

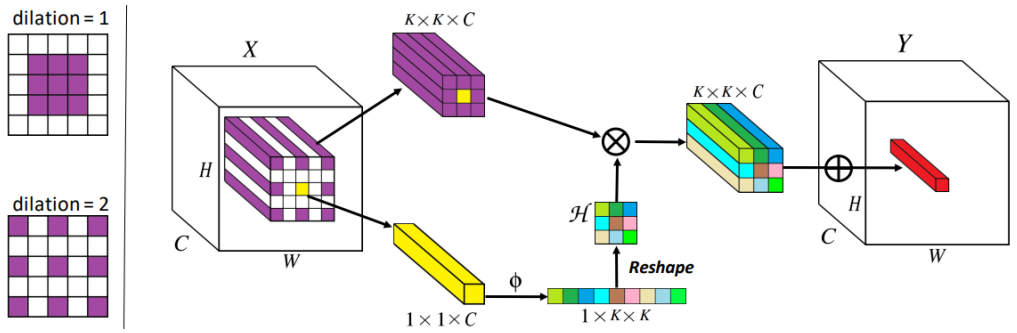

Fig. 2. **Design of dilated involution.** $\otimes$ represents $\mathcal{H}$ multiplying through the channel dimension. $\oplus$ stands for aggregation in K $\times$ K space. In this example, kernel size K = 3, G = 1, dilation size = 2 for demonstration.

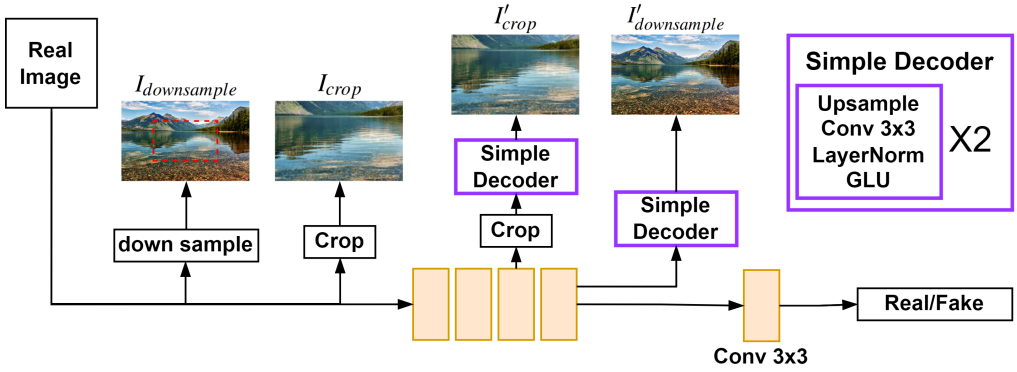

Fig. 3. **The structure of the discriminator.** The basic patchGAN[16] architecture consists of five 3 $\times$ 3 convolution. Purple box for additional decoder.

Table 3. Comparison of SIFID scores under different dilation of dilated involution on "Places".

Dilation	1	2	3	4	5
SIFID ↓	0.119	0.061	0.046	0.036	0.034

results.

Image diversity To evaluate the diversity of the generated samples, we adopt the pixel diversity metric from [17]. For each training example, we calculate the standard deviation of all pixel values along the channel axis of 100 generated images and average them. We then normalize the value using the standard deviation of the pixel value in the training image. For the data from the Places dataset, SinGAN and ConSinGAN have a diversity score of 0.52 and 0.50, respectively, while our model has a diversity score of 0.49. For the data from the LSUN dataset, SinGAN has a higher diversity score of 0.64, but this is because it often fails to generate a good global structure and the generated images are quite different from the training images. ConSinGAN's diversity score is 0.54, while our method is 0.56.

User Study For the Places dataset, we follow the evaluation method of [3, 17, 20] and compare it with SinGAN and ConSinGAN. We showed participants the generated image and the training image, and within a second, they were asked to identify the real image.

For the LSUN dataset, we follow the evaluation method of ConSinGAN[7]. Since the LSUN dataset has higher image complexity than "Places" dataset, we did not compare the generated image with the real image. We used images generated by SinGAN, ConSinGAN and our method for comparison. Each of the three methods generates 10 images for each training image, and each of the three models generates 500 images. We perform two user studies with these images. We also showed the participants three images generated by SinGAN, ConSinGAN and our method, and asked them to judge which image is the best. Unlike the Places dataset, there is no time limit, so participants could carefully compare the three images. In the test, paired, we trained the same picture using three methods and compared their generated samples. In each study, participants compared 50 sets of images.

The results are shown in Table 1 and Table 2. In general, our method has a lower SIFID than other methods. In user studies, whether on Places or LSUN dataset, we proved that our approach produced better images. Apart from this, our method has shorter training time and fewer parameters.

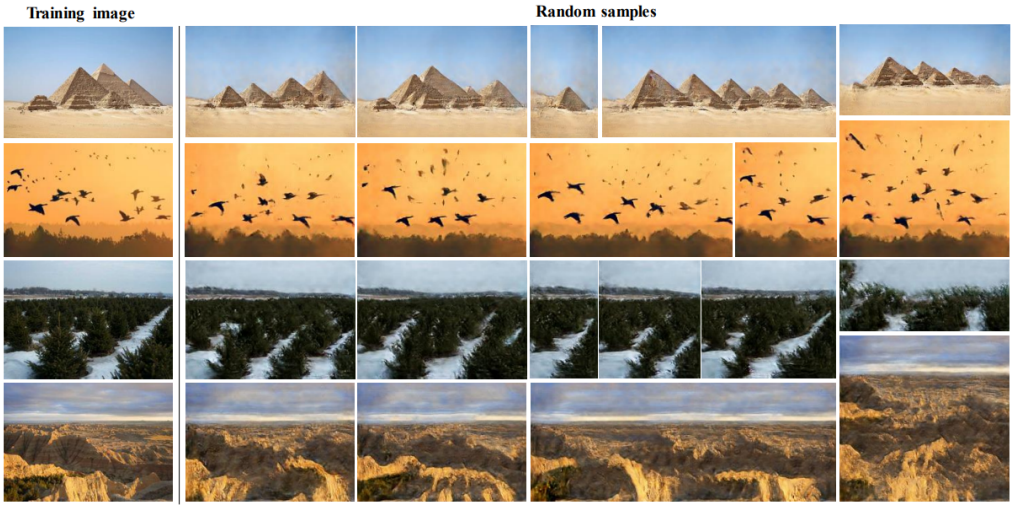

Fig. 4. **Unconditionally generated images by our method.** By inputting different sizes of noise, we can generate samples of different sizes.

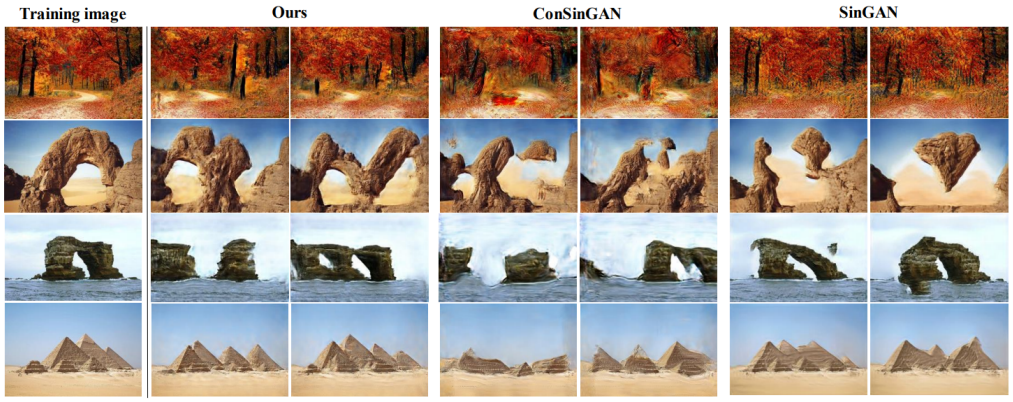

Fig. 5. **Comparison results with ConSinGAN[7] and SinGAN[17].** Our model has better overall structure and less unreasonable distribution.

Table 4. Ablation study on "Places"

Methods	SIFID ↓
Baseline	0.185
Baseline + Decode	0.132
Baseline + Dilated involution	0.054
Baseline + Decode + Dilated involution	0.046

3.2. Ablation Study

We performed an ablation study to evaluate the contribution of our different components. Results can be found in Table 4. We replaced the dilated involution with regular convolution for our baseline model and removed the extra decoder in the discriminator. The results show that both dilated involution and extra decoder improve the model performance, and dilated involution has the most contribution.

We also tested the effect of dilation size on the generated image in dilated involution. The results are shown in Table 3. We observed that larger dilation improves the SIFID score but reduces the diversity of generated images. As shown in Fig. 6, as the dilation size increases, the generated image increasingly becomes similar to the training image. We deduce that large dilation makes the model capture global information when training at various scales, making the overall structure more similar.

4. CONCLUSION

This paper proposes the dilated involution kernel, which adapts to local information, avoids channel redundancy, and better captures the global information. We use this operator to construct our single image generation model and generate

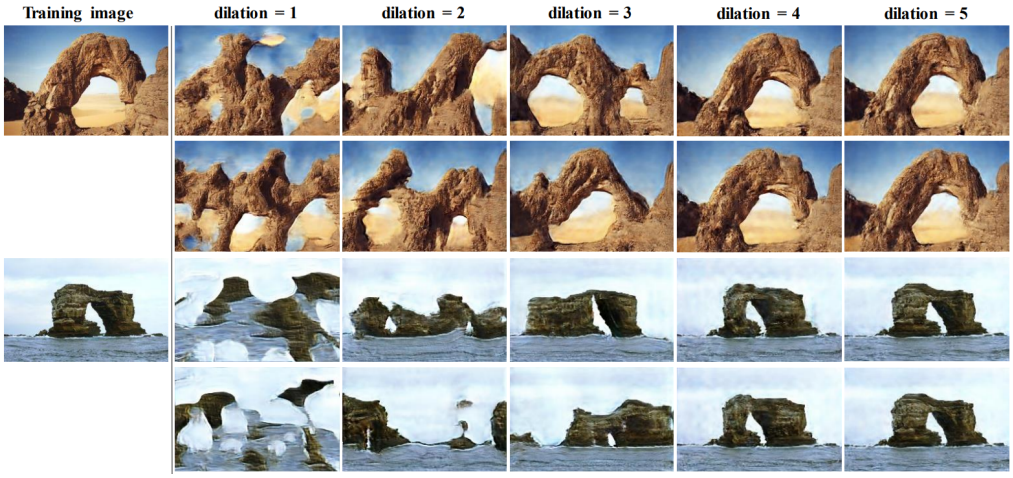

Fig. 6. **Effect of different dilation of dilated involution on "Places".**

realistic images. Our discriminator also uses an additional decoder for stability. Experimental results show that our model generates better images, has fewer parameters, and shorter training time than previous methods.

5. REFERENCES

[1] Karras et al., "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[2] Hinz et al., "Generating multiple objects at spatially distinct locations," *arXiv preprint arXiv:1901.00686*, 2019.

[3] Isola et al., "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[4] Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[5] DemirUnal, "Patch-based image inpainting with generative adversarial networks," 2018.

[6] Goodfellow et al., "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[7] Hinz et al., "Improved techniques for training single-image gans," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1300–1309.

[8] Han et al., "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589.

[9] Han et al., "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[10] Li et al., "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[11] Nie et al., "Ghostsr: Learning ghost features for efficient image super-resolution," *arXiv preprint arXiv:2101.08525*, 2021.

[12] Xiao et al., "Feature redundancy mining: Deep lightweight image super-resolution model," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 1620–1624.

[13] Li et al., "Involution: Inverting the inherence of convolution for visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12321–12330.

[14] YuKoltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[15] Gulrajani et al., "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[16] Isola et al., "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[17] Shaham et al., "Singan: Learning a generative model from a single natural image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4570–4580.

[18] Zhou et al., *Learning deep features for scene recognition using places database*, Neural Information Processing Systems Foundation, 2014.

[19] Yu et al., "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[20] Zhang et al., “Colorful image colorization,” in *European conference on computer vision*. Springer, 2016, pp. 649–666.