MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices

Andranik Sargsyan[1], Shant Navasardyan[1], Xingqian Xu[1,2], Humphrey Shi[1,2]
[1]Picsart AI Research (PAIR), [2]SHI Labs @ Georgia Tech, Oregon & UIUC

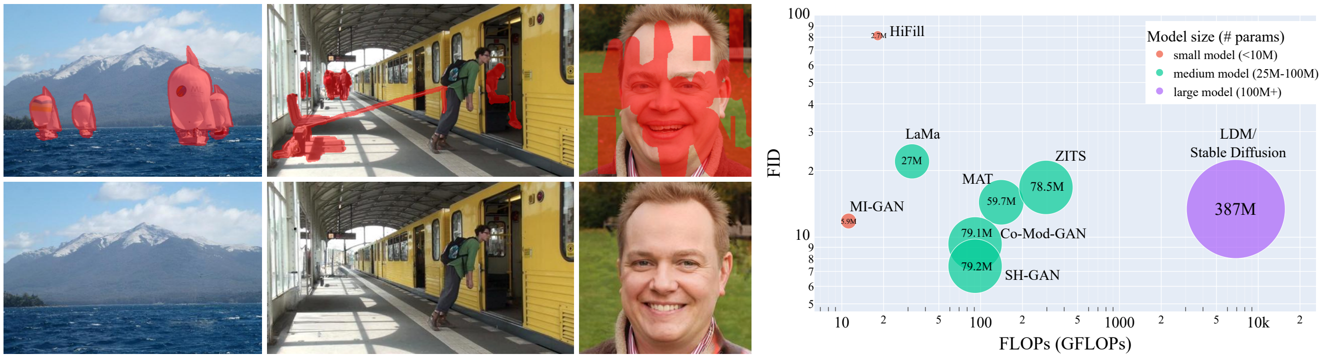

Figure 1. The proposed approach can produce plausible results both on complex scene images as well as on face images. The scene image inputs are taken from the Places 2 [67] dataset and the face image input is from FFHQ [27] dataset. The bubble chart on the right shows the advantage of our network over state-of-the-art approaches. The size of the bubble signifies the relative number of parameters of each approach and the number inside of the bubble shows the absolute number of model parameters. Our approach achieves a low FID, while being one order of magnitude smaller and faster than recent SOTA approaches.

Abstract

In recent years, many deep learning based image inpainting methods have been developed by the research community. Some of those methods have shown impressive image completion abilities. Yet, to the best of our knowledge, there is no image inpainting model designed to run on mobile devices. In this paper we present a simple image inpainting baseline, Mobile Inpainting GAN (MI-GAN), which is approximately one order of magnitude computationally cheaper and smaller than existing state-of-the-art inpainting models, and can be efficiently deployed on mobile devices. Excessive quantitative and qualitative evaluations show that MI-GAN performs comparable or, in some cases, better than recent state-of-the-art approaches. Moreover, we perform a user study comparing MI-GAN results with results from several commercial mobile inpainting applications, which clearly shows the advantage of MI-GAN in comparison to existing apps. With the purpose of high quality and efficient inpainting, we utilize an effective combination of adversarial training, model reparametrization, and knowledge distillation. Our models and code are publicly available at `https://github.com/Picsart-AI-Research/MI-GAN`.

1. Introduction

The task of image inpainting is to complete the missing regions in images in a realistic and visually plausible way. Image inpainting algorithms have found their applications in such problems as image restoration, removal of unwanted objects, etc.

Nowadays, in the era of fast evolving gadgets, particularly smartphones, image/video editing applications are growing furiously. The usage of mobile devices for making creative visual content by professionals as well as by consumers shows a great increasing tendency. A lot of mobile applications such as Photoshop Express [24], Picsart [25], Snapseed [39], TouchRetouch [18] empower the users to create awesome content utilizing AI-powered tools. One of the most popular such tools, *object remover*, allows its users to remove unwanted objects (e.g. watermarks, lines, people or random objects in the background, etc.) from their photos. So millions of users every day benefit from image inpainting technologies integrated in various mobile applications.

Current AI-based state-of-the-art inpainting algorithms such as Co-Mod-GAN [65], SH-GAN [58], LaMa [53], or LDM [46] include an inference of a heavyweight neural network, which is often not feasible to do on low-end mobile devices (see Tables 1, 2). Therefore the processing is being done on server-side resulting in dependency on Internet connection, speed, traffic and server usage costs for image editing companies.

To address this problem we propose a simple baseline for high-quality mobile image inpainting which is an order of magnitude lighter and computationally more efficient than current state-of-the-art neural networks while showing similar or sometimes even better performance (see Fig. 1).

Motivated by the generative nature of the image completion problem we design *MI-GAN*, Mobile Inpainting Generative Adversarial Network, which benefits from a hybrid system consisting of adversarial training, model reparametrization, and knowledge distillation. Thanks to adversarial nature of the MI-GAN training, image completion results appear to be visually plausible without producing blurry regions and pattern-like artifacts. Model reparametrization further increases the output quality at no cost of efficiency. Finally, knowledge distillation by a stronger network also improves the generative ability of MI-GAN.

Due to excessive evaluations MI-GAN shows great performance in terms of both quantitative and qualitative comparison with existing state-of-the-art methods.

To summarize, our contributions are three-fold.

- We propose MI-GAN, to the best of our knowledge the first mobile generative image inpainting network.
- We develop a customized knowledge distillation method that is suitable for our model and problem, as well as we implement a model re-parametrization strategy which enhances the quality of results.
- While being computationally more efficient and lighter than the existing state-of-the-art image completion approaches, MI-GAN produces similar or even better image inpainting results. Furthermore, in comparison to existing commercial mobile inpainting applications, MI-GAN results are generally rated higher by human evaluators.

2. Related Work

2.1. Image Inpainting Methods

The task of image inpainting was and is being actively investigated by researchers during past decades. Classical approaches use diffusion processes [5, 6, 50, 8, 32] or exemplar-based techniques [17, 56, 20, 2, 4, 16, 31, 13] for coherent image restoration and completion. Although the classical methods greatly recover textures and preserve image structures, they lack of semantic understanding of the image context. Hence learning-based approaches were developed and later improved.

Early deep learning papers on image inpainting were trying to fill small missing regions by image-to-image discriminative models [57, 28, 44, 36]. Later, by using generative adversarial networks (GANs) [19], researchers were able to reach extra-high quality in image inpainting task [60, 42, 59, 33, 65, 53, 58, 66]. Some papers [36, 60, 42] additionally design special convolution layers to efficiently use the information in the known region of deep features. RFR [33] suggests a recurrent feature reasoning module which utilizes correlations between neighboring pixels to fill the missing pixels. With their contextual residual aggregation mechanism Yi et al. [59] enables ultra-high resolution inpainting (up to 8K). Several methods [65, 58, 66] benefit from a strong generative model StyleGAN-v2 [54] and make adoptions to image inpainting task. [65] introduces co-modulation blocks conditioning StyleGAN-v2 modulated architecture with the input image. SH-GAN [58] introduces spectral hint units to generate high-frequency details.

By benefiting from the Fourier analysis, LaMa [53] uses Fourier convolutions and greatly improves inpainting of repetitive patterns. ZITS [15] improves over LaMa by introducing a structure restoration mechanism with transformers. MAT [34] combines the strengths of convolutional and transformer-based models for large-hole inpainting.

Recently, diffusion models [22, 52] were introduced and impacted the image inpinting task [46, 40, 48]. Palette [48] provides an image-conditioned diffusion model for image-to-image translation tasks, namely, colorization, inpainting/outpainting, and JPEG restoration. RePaint [40] adopts the unconditional pretrained Denoising Diffusion Probabilistic Model (DDPM) [22] for image inpainting task by guiding its backward process with inpainting masks. LDM [46] applies a diffusion process in its latent space which particularly allows conditioning on textual prompts, by so extending its image inpainting abilities to text guided completion.

2.2. Lightweight Generative Modeling

Although generative adversarial networks are able to create extremely high quality images often indistinguishable from reality with a naked eye, they require a tremendous amount of computations and memory consumption. For this purpose some methods [1, 10, 35, 9, 26, 45, 61, 62, 37] design lightweight generative models. [1] firstly applied the concept of knowledge distillation on GANs and showed incredible compression rates (e.g. 87:1 on CelebA [38]) while preserving the original quality. [10] utilizes the student discriminator in a triplet loss to enforce it benefit from teacher generated images also. In [35] the authors propose semantic preserving distillation loss for image-to-

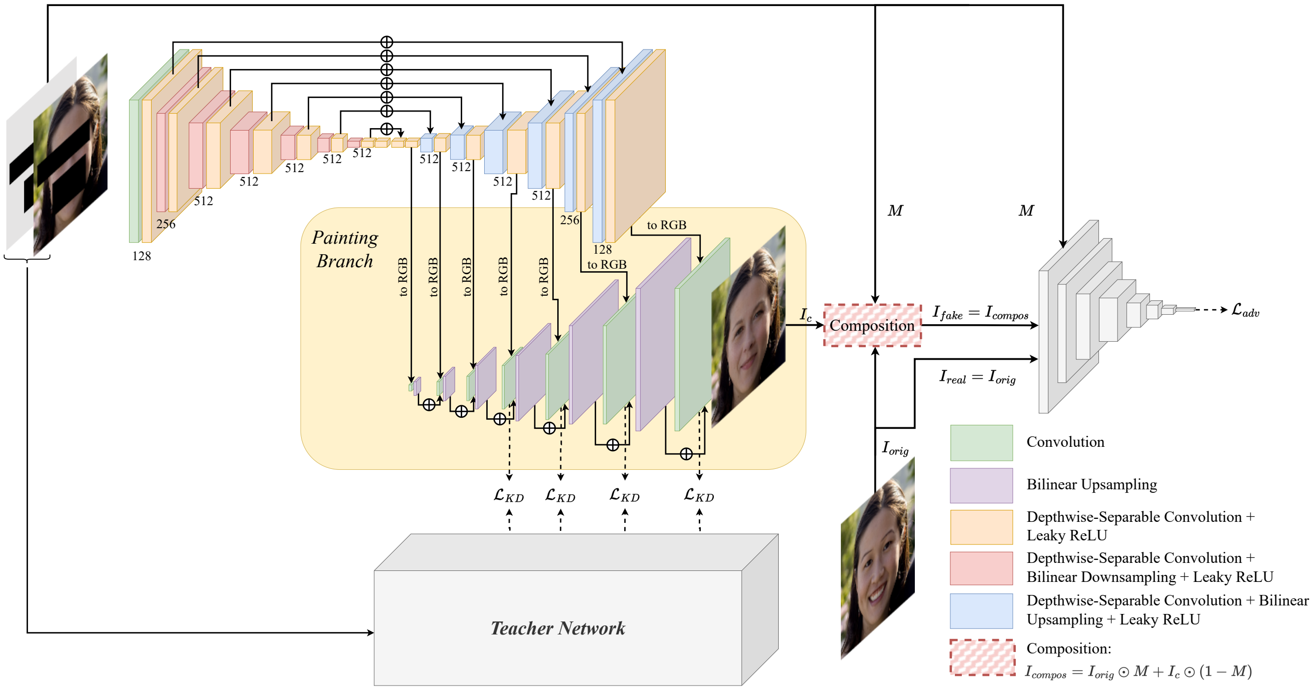

Figure 2. The overview of our method. The generator of MI-GAN consists of two branches: the main branch with a Unet like architecture and the painting branch. The main branch consists of a linear projection layer, depthwise separable convolutions, and bilinear resizing operations. The painting branch completes the missing region with layer-by-layer inpainting similar to the manual work of some experts. The training is conducted by using adversarial loss function and knowledge distillation. The discriminator is similar to the encoder of the main branch, having also residual connections and fully connected layers (more details can be found in supplementary material). As a teacher network in our experiments we take Co-Mod-GAN [65]. Please zoom in for better view.

image translation task to preserve the semantic relations between the teacher and the student features. TinyGAN [9] and TinyStarGAN-v2 [26] are distilling high-fidelity image generation networks BigGAN [7] and Stargan-v2 [12] respectively. [45] proposes a GAN compression method with a student discriminator-free online distillation scheme. Recently in [62] the authors introduce wavelet knowledge distillation to transfer the high-frequency detail generating information from the teacher generative model to the student. [37] proposes a content-aware compression paradigm, which leverages generated contents to guide the process of pruning and distillation.

3. Method

3.1. Architecture

We introduce a simple baseline for mobile image inpainting which benefits from a combination of adversarial learning, model re-parametrization, and knowledge distillation. Formally, given an input image $I_{orig} \in \mathbb{R}^{H \times W \times 3}$ and a binary mask $M \in \{0,1\}^{H \times W}$ indicating the known region with 1-values and the missing region with 0-values. MI-GAN takes the concatenated input $In = [I_{orig} \odot M, M] \in \mathbb{R}^{H \times W \times 4}$ and returns a completed image I_c which is finally composited with the original image by using the binary mask M:

$$I_{compos} = I_{orig} \odot M + I_c \odot (1 - M). \tag{1}$$

The overview of our method is shown in Fig. 2.

We design a generative adversarial network, the generator of which consists of two parts: a U-Net [47] like main branch utilizing depthwise separable convolution layers [51], and the *painting branch* imitating the inpainting process of experts. Being inspired by the architecture of StyleGAN [27], the concept of our painting branch is greatly aligned with the inpainting task. Indeed, similar to experts (artists), our painting branch paints the missing region iteratively layer-by-layer in the RGB-space starting from the overall structure and ending with the final result.

The main branch architecture of MI-GAN starts with a convolution layer to project the 4-channel input space to a higher 128 dimension. Then a fully-convolutional encoder+decoder, consisting of depthwise separable convolutions and bilinear resizing operations, is applied resulting in a tensor with dimensionality $H \times W \times 128$. All convolutional blocks are using the $LeakyReLU$ activation. Inspired by StyleGAN, to generate high-frequency details, we also add a random noise component (with a learnable noise

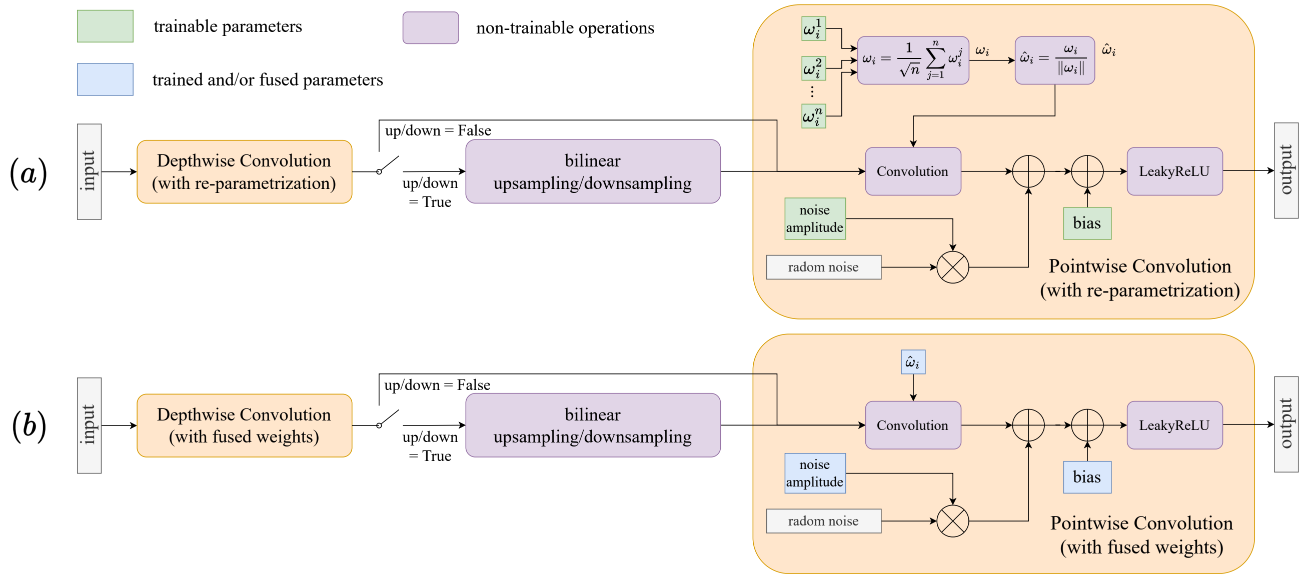
trainable parameters
non-trainable operations
trained and/or fused parameters
(a)
input
Depthwise Convolution (with re-parametrization)
up/down = False
up/down = True
bilinear upsampling/downsampling
ω_i^1
ω_i^2
ω_i^n
$\omega_i = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \omega_i^j$
ω_i
$\hat{\omega}_i = \frac{\omega_i}{\|\omega_i\|}$
$\hat{\omega}_i$
Convolution
LeakyReLU
noise amplitude
bias
radom noise
Pointwise Convolution (with re-parametrization)
output
(b)
input
Depthwise Convolution (with fused weights)
up/down = False
up/down = True
bilinear upsampling/downsampling
$\hat{\omega}_i$
Convolution
LeakyReLU
noise amplitude
bias
radom noise
Pointwise Convolution (with fused weights)
output

Figure 3. The structure of our depthwise separable convolution layers showing (a) the model re-parametrization during training and (b) the utilization of the fused weights during the inference. We apply bilinear resizing in the case when the depthwise separable convolution block changes its input resolution. In the case of both training and inference in each pointwise convolution layer a random noise with a trainable amplitude is applied.

amplitude) to the pointwise convolution layers of the decoder of the main branch (see Fig. 3). The painting branch takes the intermediate tensors from the main branch decoder and "paints" them to the RGB-space with projection convolutional layers, resulting in images $x_i \in \mathbb{R}^{\frac{H}{2^i} \times \frac{W}{2^i} \times 3}$ for $i = 0, 1, \ldots, 6$. Then, the final image $I_c \in \mathbb{R}^{H \times W \times 3}$ is obtained by "painting" (adding) the images x_i as layers on top of each other: $\hat{x}_6 = x_6$,

$$\begin{aligned} \hat{x}_i = BilinearUp(\hat{x}_{i+1}) + x_i,\ i = 5, 4, \ldots, 0, \\ I_c = \hat{x}_0, \end{aligned} \tag{2}$$

where $BilinearUp$ is the bilinear upsampling operation with scaling factor equal to 2.

After the painting branch returns the inpainted image I_c, we get the final result by using Eq. 1.

3.2. Training

To train MI-GAN we use a combination of an adversarial and a knowledge distillation loss functions $\mathcal{L}_{adv}$ and $\mathcal{L}_{KD}$ respectively. For an adversarial training we borrow the discriminator architecture from StyleGAN [27] and replace convolutional layers with depthwise-separable convolutions. Also, similar to StyleGAN [27], we utilize a non-saturating (with softplus non-linearity) loss function with R_1-regularization [41] for discriminator training. For the generator MI-GAN the adversarial component of the loss function will accordingly be as follows:

$$\mathcal{L}_{adv} = \mathbb{E}_{x,m \sim P_{c,m}}[SoftPlus(-D_w(G_\theta(x, m), m))], \tag{3}$$

where$P_{c,m}$ is the joint distribution of corrupted images (which should be inpainted) and the corresponding missing regions, D_w is the discriminator, and G_θ is the generator, i.e. our MI-GAN.

For a knowledge distillation we choose Co-Mod-GAN [65] as a teacher network for MI-GAN due to its strong generative ability inherited from StyleGAN-v2 [54]. Similar to us, inspired by StyleGAN, Co-Mod-GAN also has a painting branch. Hence we put a knowledge-distillation loss between the inpainted regions of intermediate results of the painting branches of Co-Mod-GAN and MI-GAN:

$$\mathcal{L}_{KD} = \sum_{i=0}^{3} \|(x_i - x_i^C) \odot (1 - M_i)\|, \tag{4}$$

where x_i and x_i^C are the intermediate results of painting branches of MI-GAN and Co-Mod-GAN respectively, and $M_i \in \{0, 1\}^{\frac{H}{2^i} \times \frac{W}{2^i}}$ is obtained by resizing the binary mask M to the size of x_i (with the nearest neighbour interpolation). In Sec. 4.6 we discuss the effect of knowledge distillation loss.

So the final loss function for MI-GAN generator will be the weighted combination

$$\mathcal{L} = \lambda_1 \mathcal{L}_{adv} + \lambda_2 \mathcal{L}_{KD}. \tag{5}$$

In our experiments we take $\lambda_1 = 1, \lambda_2 = 2$.

3.3. Model Re-Parametrization

Re-parametrization is widely used for such tasks as compact model design [29], neural architecture search [11, 63]

	FFHQ		**Places2**		FLOPS	Params
Method	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	(GFLOPS)	($\times 10^6$)
LaMa	32.71	0.259	22.00	0.378	32.05	27.05
Co-Mod-GAN	4.70	0.257	9.32	0.397	91.21	79.17
SH-GAN	**4.33**	0.254	**7.40**	0.392	91.27	79.21
ZITS	-	-	16.78	**0.356**	295.72	78.49
MAT	7.00	**0.231**	14.38	0.394	140.74	59.78
LDM	-	-	13.40	0.385	6,896.16	387.25
HiFill	-	-	81.27	0.488	18.14	**2.72**
MI-GAN (ours)	4.99	0.257	11.83	0.394	**11.19**	5.95

Table 1. Quantitative comparison of our method with state-of-the-art approaches on 256 resolution images.

and pruning [14]. Inspired by recent work on online re-parametrization [23], we adopt the technique to our architecture. We skip the linear scaling layer, instead we use weight normalizations described in [49]. We apply the re-parametrization for all convolution layers in MI-GAN architecture.

For a convolutional layer L_i during training we define n kernels $\omega_i^1, \omega_i^2, \ldots, \omega_i^n \in \mathbb{R}^{C_O \times C_I \times k \times k}$, where C_O and C_I are the numbers of output and input channels of L_i respectively (for depthwise convolutions $C_I = 1$ regardless of input channel number of L_i), and k is the kernel size. Then we initialize the kernels randomly from standard normal distribution, combine them via summation, and scale the result to have standard deviation equal to 1:

$$\omega_i = \frac{1}{\sqrt{n}} \sum_{j=1}^{n} \omega_i^j. \tag{6}$$

This follows by a weight normalization operation $\hat{\omega}_i = \frac{\omega_i}{\|\omega_i\|}$. Finally $\hat{\omega}_i$ is being used as a convolutional (or depthwise convolutional) kernel:

$$L_i(x) = Conv(x, \hat{\omega}_i) + b_i, \tag{7}$$

where b_i is the convolutional bias initialized with zeros. During inference all additional parameters are fused and $\hat{\omega}_i$ is directly applied in Eq. 7.

Our re-parametrization for depthwise-separable convolution blocks is shown in Fig. 3. In our experiments we take $n = 9$ to somehow "compensate" and make the network's capacity close to the capacity of its teacher network. The impact of the re-parametrization trick is discussed in Sec. 4.6.

4. Experiments

4.1. Implementation Details

Our experiments are conducted on Places2 [67] and FFHQ [27] datasets. Though our proposed model is fully convolutional, and can be used for inference on varying resolutions and aspect ratios, we create separate architectures for training on 256 and 512 resolution inputs. The purpose of having separate architectures is to minimize the FLOPS and the memory usage for higher resolution inputs. The architectural details of our models can be found in the supplementary material.

To train our models on Places2 dataset we use *Places365-Standard* subset, which contains $\sim$ 1.8 million images. All the metrics are calculated on Places2 validation set with $36,500$ images. For 256 and 512 resolution models we train them for $\sim$ $950,000$ and $\sim$ $1,600,000$ iterations respectively. The training time for 256 and 512 resolution models was 23 days with 2 Nvidia Quadro-RTX 8000 GPUs and 30 days with 8 Nvidia A6000 GPUs respectively.

We train the teacher Co-Mod-GAN model and our 256 resolution model on FFHQ by using randomly chosen $60,000$ images, and evaluate all models on the rest $10,000$ images. We train our model for $\sim$ $800,000$ iterations for 10 days with 8 Nvidia A6000 GPUs.

Adam optimizer is used with learning rate 0.001 and with $\beta = (0, 0.99)$ both for the generator and the discriminator. We fix batch size 32 for all experiments.

4.2. Quantitative Comparison

Tables 1 and 2 show comparison of our method with state-of-the-arts Co-Mod-GAN [65], SH-GAN [58], LaMa [53], LDM [46], ZITS [15], MAT [34] and HiFill [59] on Places2 and FFHQ datasets respectively. All the models besides Co-Mod-GAN are taken from their official repositories or are kindly provided by the corresponding authors. For Co-Mod-GAN we re-implemented it in PyTorch [43], and trained and tested the replicated version. The FID [21] scores of the replicated Co-Mod-GAN match the FID scores in the original paper.

For the purpose of metric calculation we do the inference of all models in comparison using the free-form mask generation strategy as in Co-Mod-GAN [65]. The mask generation process combines multiple brush strokes and rectangles. The number of brush strokes is chosen randomly between 0 and 20. The widths of the brush strokes are chosen uniformly from the range (12, 40). Generated masks contain between 0 to 5 random rectangles up to the image size, and between 0 to 10 rectangles having maximum side fixed to the half of the image size. With this method of mask generation, masked area ratio varies between 0 and 1.

We use *Fréchet Inception Distance* (FID) [21] as our primary metric. Also, we provide a comparison in terms of the perceptual metric LPIPS [64]. To show the computational complexity of the models we provide the number of *floating point operations per second* (FLOPS) and for the size of the models we report the number of model parameters.

As can be noticed from Table 1, on Places2 dataset our model outperforms LaMa, ZITS, MAT and HiFill in terms of FID and is comparable with LDM, while slightly underperforms Co-Mod-GAN and SH-GAN being on par with

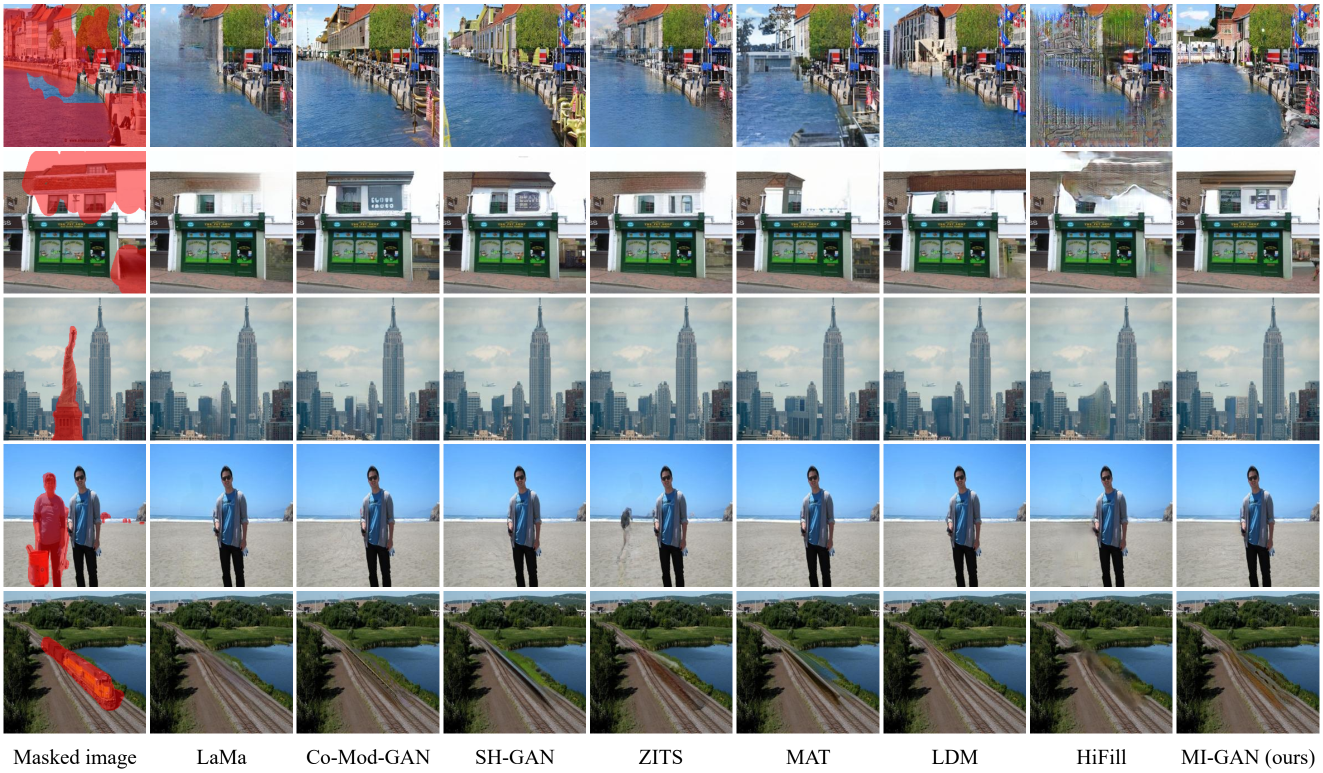

Figure 4. Qualitative results of our model and other state of the art approaches on 256 resolution Places 2 samples with free-form masks.

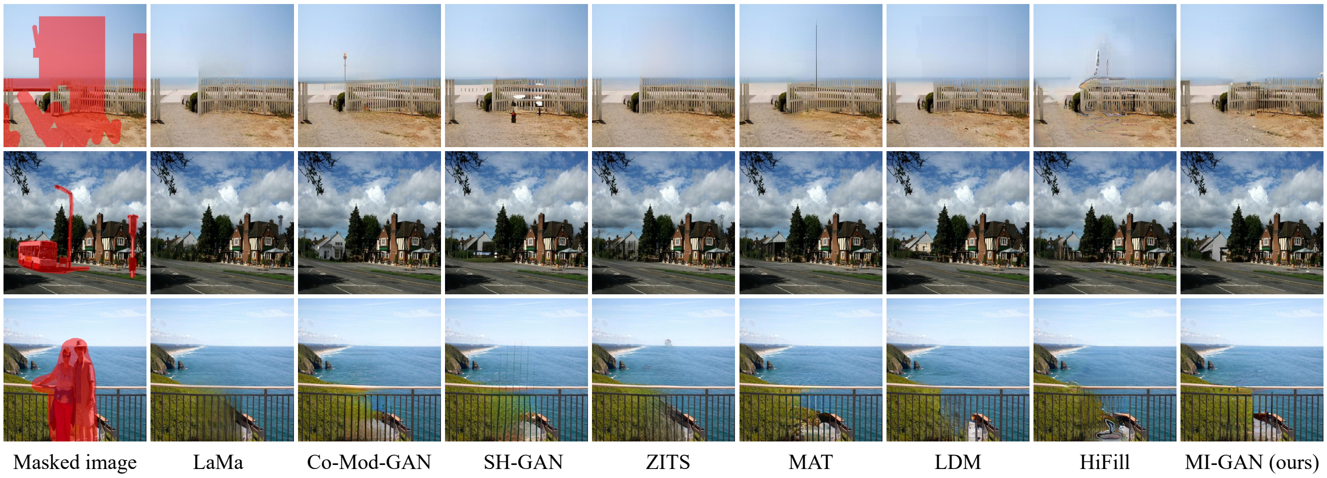

Figure 5. Example results of our 512 resolution model and other state of the art approaches on 512 resolution Places2 test samples using free-form and user masks. Please zoom for a better view.

them in terms of the perceptual metric LPIPS. However MI-GAN is almost 8 times faster and 13 times lighter than Co-Mod-GAN and SH-GAN. This emphasizes a great generative ability of MI-GAN while being a compact model designed for mobile devices. Let us notice that LDM and MAT does not provide a model working on 256 resolution, so to inference on 256 resolution we resize images to 512, process them with 512 models and then resize back to 256. On FFHQ we can observe a similar picture, while our method is even closer to Co-Mod-GAN and SH-GAN (i.e. has similar generative ability). For LaMa to inference on FFHQ dataset we have used the model trained on CelebA [38] dataset. LDM, ZITS and HiFill do not provide an inpainting model trained on faces.

For the image resolution 512 MI-GAN performs similarly to LaMa and ZITS in terms of FID and slightly underperforms Co-Mod-GAN, SH-GAN, LDM and MAT. Qualitative analysis performed in Sec. 4.3 shows that the small

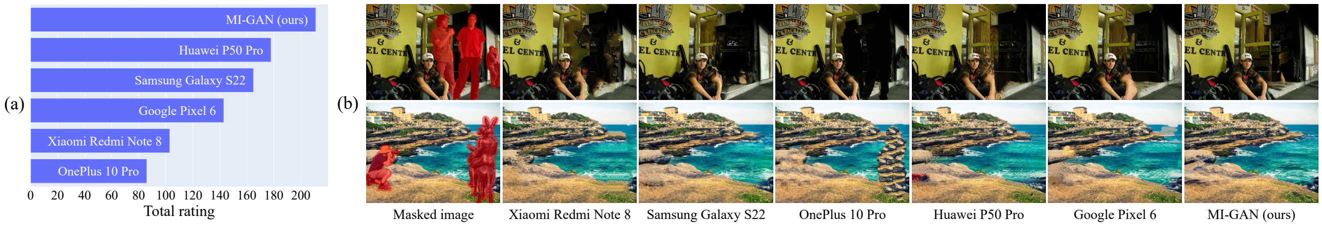

Figure 6. (a) Total rating of each method based on our user study. (b) Visual examples from the user study. Please zoom-in for better view.

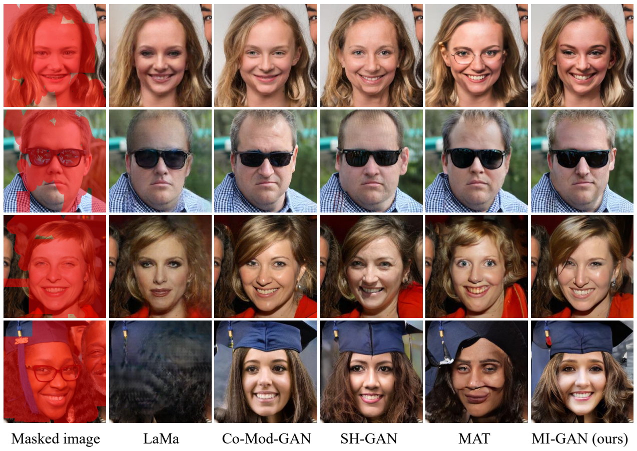

Figure 7. Qualitative results from our model and other approaches on FFHQ dataset samples. Please zoom for a better view.

Method	FID ↓	LPIPS ↓	FLOPS (GFLOPS)	Parameters ($\times 10^6$)
LaMa	12.36	0.314	128.06	27.05
Co-Mod-GAN	8.05	0.343	121.09	79.79
SH-GAN	**7.03**	0.339	121.15	79.82
ZITS	12.94	**0.310**	666.97	78.49
MAT	8.67	0.339	288.78	61.56
LDM	8.46	0.342	27,581.56	387.25
HiFill	64.07	0.438	72.12	**2.72**
MI-GAN (ours)	10.00	0.345	**15.69**	5.98

Table 2. Quantitative comparison of our method with state-of-the-art approaches on 512 resolution images.

difference in terms of FID has a small impact on visual results. Instead, our model is more than 7.5 times faster and 4.5 times lighter than the mentioned state-of-the-art approaches (see Table 2).

Notice that even though HiFill has less number of parameters compared to our approach, MI-GAN is computationally more efficient and significantly outperforms HiFill quantitatively both on 256 and 512 resolutions. The reason that our model is computationally more efficient is that MI-GAN is a single-stage fully convolutional network consisting of depthwise-separable convolutions, whereas HiFill uses regular convolutions on higher resolution features and has a costly attention mechanism in its refinement network.

To show the effectiveness of our approach on different mask sizes, we provide FID comparisons with varying mask ratios in the appendix. Additionally, in order to further prove the robustness of our approach against other metrics, we also include a comparison in terms of PSNR, SSIM [55], P-IDS and U-IDS [65] metrics in the appendix.

4.3. Qualitative Comparison

In this subsection we discuss the qualitative advantages of our model over some state-of-the-art approaches. As illustrated in Fig. 4 our model produces semantically meaningful content, even if the input mask is large. Our model has a strong generative capability, and the results are sharp, while approaches like LaMa [53] and ZITS [15] produce blurry results on larger masks (see the 1st row of the figure as an example). Besides, as can be noticed from the qualitative comparison, on 256 resolution images our approach can perform better than the bigger and exceedingly computationally expensive model LDM [46].

On higher resolution images (see Fig. 5) our model still produces semantically meaningful results. Besides, our results don't have a color inconsistency issue unlike LDM, LaMa and ZITS results (see the 1st row of the figure as an exmaple).

We also conduct a qualitative comparison on face images (see Fig. 7). On face images our model's results are on par with Co-Mod-GAN and SH-GAN, while being better than LaMa and MAT.

More qualitative results can be found in the supplementary material.

4.4. User Study

In addition to comparing our MI-GAN model with SOTA inpainting methods, we conduct a user study against 5 built-in commercial mobile image inpainting apps. For this, we select a diverse set of 20 real world photos from OpenImages [30] dataset with varying resolutions and, using each method in comparison, carefully mask and remove predetermined objects from each photo. Then, we ask 3 people to anonymously rate each of 5 results from 1 to 5 for each photo and sum the given ratings for each method (see Fig. 6.a). As can be seen from the diagram, our approach has a

Device Name	**256-resolution** MI-GAN speed (ms, mean/std)	 Co-Mod-GAN speed (ms, mean/std)
iPhone7	1030.25 / 13.37	4475.33 / 39.55
iPhoneX	630.80 / 12.21	2746.00 / 28.84
iPad mini (5th gen)	552.40 / 8.10	2686.17 / 41.41
iPhone14-pro-max	296.00 / 1.35	1374.40 / 84.78
Galaxy Tab S7+	686.17 / 12.36	- / -
Samsung Galaxy S8	1476.40 / 5.98	- / -
vivo Y12	2918.08 / 33.47	- / -

Table 3. The actual speed of our 256 model vs Co-Mod-GAN 256 model deployed on various mobile devices. The actual speed is measured in miliseconds, the means and the stds of the speed are presented for each mobile device. Some numbers are missing, since Co-Mod-GAN was providing an out of memory error.

clear advantage over other methods. Fig. 6.b shows qualitative samples from user study, where our result has obvious advantage over commercial mobile apps. We include more results in the supplementary material.

4.5. Actual Speed on Mobile Devices

Besides calculating FLOPS we also measure actual speed of our method on real mobile devices and compare MI-GAN with Co-Mod-GAN. As can be noticed from Table 3 on 256 resolution inputs our method is more than 4 times faster on average than Co-Mod-GAN, which allows MI-GAN to be used in real-world applications on mobile devices.

All models in Table 3 are using CPUs of mobile devices and are deployed by using the neural network framework ONNX [3]. Nothing special was conducted to optimize the speed or memory usage of MI-GAN on ONNX, so we believe MI-GAN can be further optimized in terms of speed or memory consumption (particularly running it on GPUs of mobile devices). The actual running times on 512 resolution images can be found in the supplementary material.

4.6. Ablation Study

In this subsection we show the impact of our reparametrization trick and knowledge distillation loss function for MI-GAN 256 resolution model. Both are designed to close the generative performance gap between MI-GAN and a larger inpainting network. Table 4 shows the quantitative effect of both. Fig. 8 shows that by adding re-parametrization trick we increase the semantic understanding of the model to generate more meaningful results. Further, the knowledge distillation adds more generative ability. Although both modifications has positive impact, one can notice that re-parametrization trick improves the method drastically (see also Fig. 8).

Method	FID ↓	LPIPS ↓
Baseline	14.94	0.407
Baseline + Re-Param	12.22	0.400
MI-GAN: Baseline + Re-Param + KD	**11.83**	**0.39**

Table 4. The effect of model re-parametrization and knowledge distillation on MI-GAN 256 resolution model. As can be noticed both have positive impact in terms of both metrics FID and LPIPS.

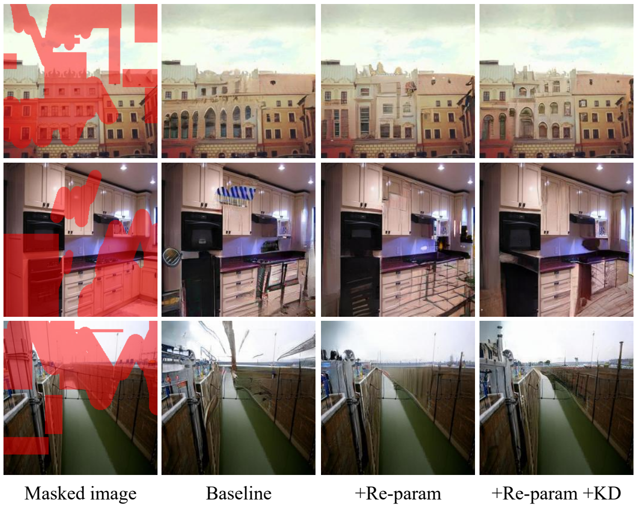

Figure 8. Visual results of ablation study on model reparametrization and knowledge distillation.

5. Limitations and Societal Impact

Although our method has good generation abilities producing visually plausible results, similar to other state-of-the-art approaches it sometimes meets difficulties when reconstructing complex 3D structures. We hypothesize this can be later improved by guiding the network with more information about the 3D structure of the objects in images. For visual examples of MI-GAN's failure cases please see our supplementary material.

Our work aims to make the creative image editing easier and accessible for everyone with their mobile devices. Similar to other deep learning based image editing approaches our method may have both positive and negative societal impact depending on its application. As a positive impact MI-GAN can help the users to restore old photos or remove unpleasant objects from their photos making them more aesthetic. On the negative side it can be used to misrepresent the reality by removing important facts from the images. We strongly believe and hope that our work will give more positive value to creative editing users than its potential negative societal impact.

6. Conclusion

In this paper we introduce MI-GAN, the first generative mobile image inpainting network. By using depthwise separable convolutions, model re-parametrization, and knowledge distillation, MI-GAN is able to produce high-quality results while having a low computational cost and a small number of parameters. Qualitative and quantitative comparisons show that MI-GAN is comparable or sometimes outperforms the current state-of-the-art approaches while being a way faster and smaller.

References

[1] Angeline Aguinaldo, Ping-Yeh Chiang, Alex Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing gans using knowledge distillation. *arXiv preprint arXiv:1902.00159*, 2019. 2

[2] Michael Ashikhmin. Synthesizing natural textures. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 217–226, 2001. 2

[3] Junjie Bai, Fang Lu, Ke Zhang, et al. Onnx: Open neural network exchange. `https://github.com/onnx/onnx`, 2019. 8

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 2

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000. 2

[6] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003. 2

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[8] Tony F Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of visual communication and image representation*, 12(4):436–449, 2001. 2

[9] Ting-Yun Chang and Chi-Jen Lu. Tinygan: Distilling biggan for conditional image generation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 2, 3

[10] Hanting Chen, Yunhe Wang, Han Shu, Changyuan Wen, Chunjing Xu, Boxin Shi, Chao Xu, and Chang Xu. Distilling portable generative adversarial networks for image translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3585–3592, 2020. 2

[11] Shoufa Chen, Yunpeng Chen, Shuicheng Yan, and Jiashi Feng. Efficient differentiable neural architecture search with meta kernels. *ArXiv*, abs/1912.04749, 2019. 4

[12] Yunjey Choi, Youngjung Uh, Jaejoo Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 3

[13] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004. 2

[14] Xiaohan Ding, Tianxiang Hao, Jianchao Tan, Ji Liu, Jungong Han, Yuchen Guo, and Guiguang Ding. Resrep: Lossless cnn pruning via decoupling remembering and forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4510–4520, 2021. 5

[15] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11368, 2022. 2, 5, 7

[16] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001. 2

[17] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999. 2

[18] ADVA Soft GmbH. Touchretouch. 1

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[20] Paul Harrison. A non-hierarchical procedure for re-synthesis of complex textures. 2001. 2

[21] 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. 5

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[23] Mu Hu, Junyi Feng, Jiashen Hua, Baisheng Lai, Jianqiang Huang, Xiaojin Gong, and Xian-Sheng Hua. Online convolutional re-parameterization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 568–577, 2022. 5

[24] Adobe Inc. Photoshop express photo editor. 1

[25] Picsart Inc. Picsart photo editor & video. 1

[26] Paras Kapoor and Tien D. Bui. Tinystargan v2: Distilling stargan v2 for efficient diverse image synthesis for multiple domains. In *BMVC*, 2021. 2, 3

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 1, 3, 4, 5

[28] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *German conference on pattern recognition*, pages 523–534. Springer, 2014. 2

[29] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 4

[30] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 7

[31] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. *Acm transactions on graphics (tog)*, 22(3):277–286, 2003. 2

[32] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV*, volume 1, pages 305–312, 2003. 2

[33] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7760–7768, 2020. 2

[34] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10758–10768, 2022. 2, 5

[35] Zeqi Li, Ruowei Jiang, and Parham Aarabi. Semantic relation preserving knowledge distillation for image-to-image translation. In *European conference on computer vision*, pages 648–663. Springer, 2020. 2

[36] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018. 2

[37] Yuchen Liu, Zhixin Shu, Yijun Li, Zhe Lin, Federico Perazzi, and Sun-Yuan Kung. Content-aware gan compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12156–12166, 2021. 2, 3

[38] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 2, 6

[39] Google LLC. Snapseed. 1

[40] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022. 2

[41] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 4

[42] Shant Navasardyan and Marianna Ohanyan. The family of onion convolutions for image inpainting. *International Journal of Computer Vision*, pages 1–30, 2022. 2

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[44] Jimmy S Ren, Li Xu, Qiong Yan, and Wenxiu Sun. Shepard convolutional neural networks. *Advances in neural information processing systems*, 28, 2015. 2

[45] Yuxi Ren, Jie Wu, Xuefeng Xiao, and Jianchao Yang. Online multi-granularity distillation for gan compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6793–6803, 2021. 2, 3

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 5, 7

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[48] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022. 2

[49] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016. 5

[50] Jianhong Shen and Tony F Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002. 2

[51] L Sifre and S Mallat. Rigid-motion scattering for image classification [phd thesis]. *Ecole Polytechnique*, 2014. 3

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[53] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2149–2159, 2022. 2, 5, 7

[54] Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. Stylegan2 distillation for feed-forward image manipulation. In *European conference on computer vision*, pages 170–186. Springer, 2020. 2, 4

[55] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003. 7

[56] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 479–488, 2000. 2

[57] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, 25, 2012. 2

[58] Xingqian Xu, Shant Navasardyan, Vahram Tadevosyan, Andranik Sargsyan, Yadong Mu, and Humphrey Shi. Image completion with heterogeneously filtered spectral hints. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. 2, 5

[59] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020. 2, 5

[60] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4471–4480, 2019. 2

[61] Linfeng Zhang, Xin Chen, Runpei Dong, and Kaisheng Ma. Region-aware knowledge distillation for efficient image-to-image translation. *arXiv preprint arXiv:2205.12451*, 2022. 2

[62] Linfeng Zhang, Xin Chen, Xiaobing Tu, Pengfei Wan, Ning Xu, and Kaisheng Ma. Wavelet knowledge distillation: Towards efficient image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12464–12474, 2022. 2, 3

[63] Mingyang Zhang, Xinyi Yu, Jingtao Rong, Linlin Ou, and Feng Gao. Repnas: Searching for efficient re-parameterizing blocks. *arXiv preprint arXiv:2109.03508*, 2021. 4

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[65] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021. 2, 3, 4, 5, 7

[66] Haitian Zheng, Zhe Lin, Jingwan Lu, Scott Cohen, Eli Shechtman, Connelly Barnes, Jianming Zhang, Ning Xu, Sohrab Amirghodsi, and Jiebo Luo. Cm-gan: Image inpainting with cascaded modulation gan and object-aware training. *arXiv preprint arXiv:2203.11947*, 2022. 2

[67] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 1, 5