Improved Techniques for Training Single-Image GANs

Tobias Hinz[1], Matthew Fisher[2], Oliver Wang[2], and Stefan Wermter[1]
[1]Knowledge Technology, University of Hamburg, Germany
[2]Adobe Research

Abstract

Recently there has been an interest in the potential of learning generative models from a single image, as opposed to from a large dataset. This task is of significance, as it means that generative models can be used in domains where collecting a large dataset is not feasible. However, training a model capable of generating realistic images from only a single sample is a difficult problem. In this work, we conduct a number of experiments to understand the challenges of training these methods and propose some best practices that we found allowed us to generate improved results over previous work. One key piece is that, unlike prior single image generation methods, we concurrently train several stages in a sequential multi-stage manner, allowing us to learn models with fewer stages of increasing image resolution. Compared to a recent state of the art baseline, our model is up to six times faster to train, has fewer parameters, and can better capture the global structure of images.

1. Introduction

Generative Adversarial Networks (GANs) [12] are capable of generating realistic images [6] that are often indistinguishable from real ones [24]. The resulting models can be used for different tasks, such as unconditional and conditional image synthesis [23, 17], image inpainting [9], and image-to-image translation [19, 46]. However, most GANs are trained on large datasets, typically consisting of tens of thousands of images which can be time-consuming and expensive. In some cases, it might be preferable to train a generative model on a small number of images or, in the limit, on a single image. This is useful if we want to obtain variations of a given image, work with a very specific image or style, or only have access to little training data. The recently proposed SinGAN [33] introduces a GAN that is trained on a single image for tasks such as unconditional image generation and harmonization.

SinGAN is trained in a multi-stage and multi-resolution approach, where the training starts at a very low resolution (e.g. 25×25 pixels) at the first stage. The training progresses through several "stages", at each of which more layers are added to the generator and the image resolution is increased. At each stage all previously trained stages (i.e. the generator's lower layers) are frozen and only the newly added layers are trained. We find that exactly how multi-stage and multi-resolution training is handled is critical. In particular, training only one stage at a given time limits interactions between different stages, and propagating images instead of feature maps from one generator stage to the next negatively affects the learning process. Conversely, training all stages end-to-end causes overfitting in the single image scenario, where the network collapses to generating only the input image. We experiment with this balance, and find a promising compromise, training multiple stages in parallel with decreased learning rates, and find that this improves the learning process, leading to more realistic images with less training time. Furthermore, we show how it is possible to directly trade-off image quality for image variance, where training more stages in parallel means a higher global image consistency at the price of less variation.

We also conduct experiments over the choice of rescaling parameters, i.e. how we decide at which image resolution to train at each stage. We observe that the quality of the generated images, especially the overall image layout, quickly degrades when there are not enough training stages with small resolution. Our experiments show that lower stages with smaller resolutions are important for the overall image layout, while higher stages with larger resolution are important for the final image texture and color. We find that we only need relatively few training stages with high-resolution images in order to still generate images with the correct texture. As a consequence, we put a higher weight on smaller resolution images during training while using fewer of the stages to train on high-resolution images.

Finally, since our model trains several stages in parallel, we can introduce a *task-specific fine-tuning stage* which can be performed on any trained model. For several tasks we show how to fine-tune the trained model on a given specific image to further improve results. This shows benefits with as few as 500 additional training iterations and is, therefore, very fast (less than two minutes on our hardware).

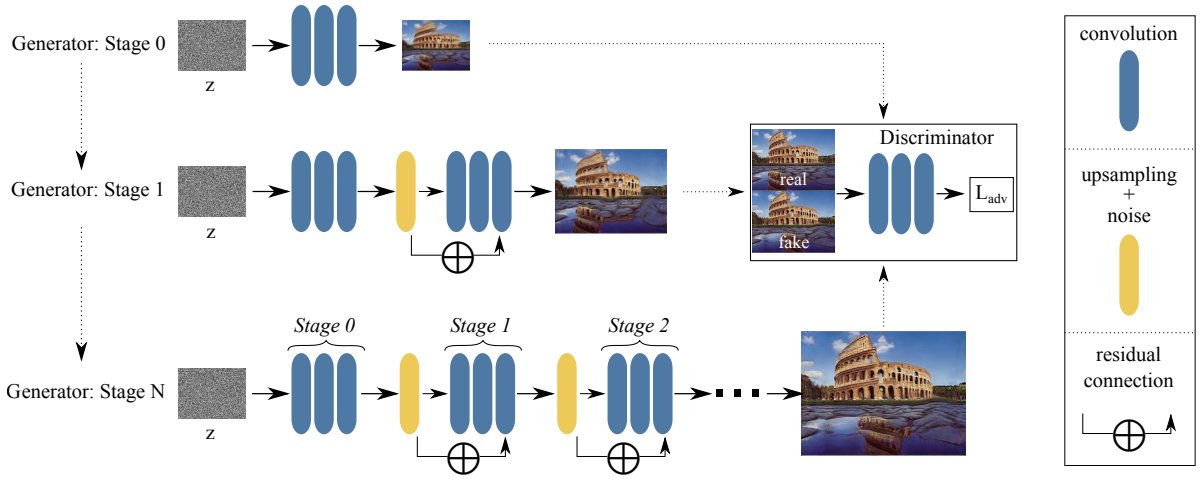


Figure 1. Overview of our model (ConSinGAN). We start training at 'Stage 0' with a small generator and small image resolution. With increasing number of stages both the generator capacity and image resolution increase.

Combining these proposed architecture and training modifications enables us to generate realistic images with fewer stages and significantly reduced overall training time (20-25 minutes versus 120-150 minutes in the original SinGAN work). To summarize, our main contributions are:

1. We train several stages in parallel with different learning rates and can trade-off the variance in generated images vs. their conformity to the original training image.
2. We do not generate images at intermediate stages but propagate features directly from one stage to the next.
3. We improve the rescaling approach for multi-stage training, which enables us to train on fewer stages.
4. We introduce a fine-tuning phase which can be used on pre-trained models to obtain optimal results for specific images and tasks.

2. Related Work

Learning the statistics and distribution of patches of a single image has been known to provide a powerful prior since the empirical entropy of patches inside a single image is smaller than the empirical entropy of patches inside a distribution of images [48]. By using this prior, many tasks such as inpainting [38, 42], denoising [49], deblurring [32], retargeting [29, 30], and segmentation [10] can be solved with only a single image. In particular, image super-resolution [40, 18, 11, 35, 3] and editing [7, 8, 14, 31, 37, 28] from a single image have been shown to be successful and a large body of work focuses specifically on this task. Recent work also shows that training a model on a single image with self-supervision and data augmentation can be enough to learn powerful feature extraction layers [1].

Approaches that train GAN models on single images are still relatively rare and are usually based on a bidirectional similarity measure for image summarization [36]. Some approaches do not use natural images, but instead train only on texture images [20, 45, 5, 25]. At this time, only few models are capable of being trained on a single 'natural' image [33, 34, 39]. Other novel approaches target applications such as image-to-image translation with only two images as training data [26, 4].

The work most relevant to our approach is SinGAN [33] which is the only model that can perform unconditional image generation after being trained on a single natural image. SinGAN trains both the generator and the discriminator over multiple stages of different image resolutions as it is useful to learn statistics of image patches across different image scales [2]. The output at each stage is an image which is used as input to the next stage and each stage is trained individually while the previous stages are kept frozen.

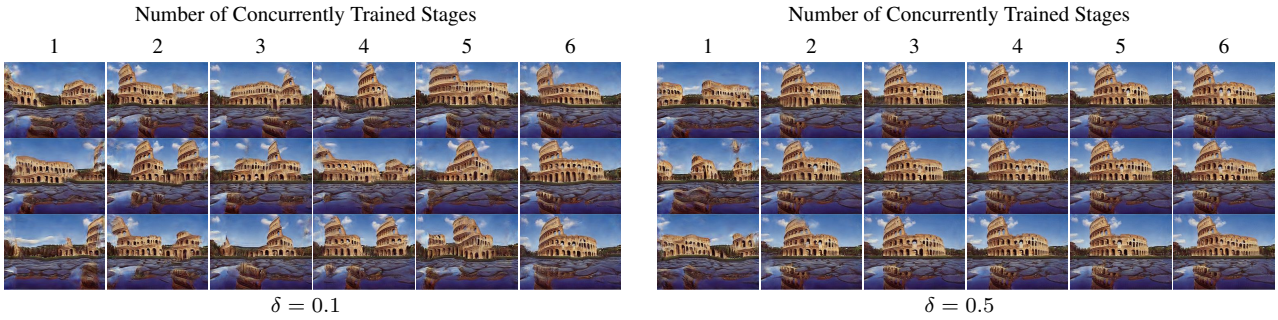
3. Methodology

We now describe our findings in more detail, starting with the training of a multi-stage architecture, followed by best practices we found for scaling learning rate and image resolutions at different stages during training.

Multi-stage Training Multi-scale image generation is of critical importance [33], however, there are many ways in which this can be realized. SinGAN only trains the current (highest) stage of its generator and freezes the parameters of all previous stages. ProGAN [22] presents a progressive growing scheme that adds levels with all weights unfrozen, and more recently [21, 23] train the entire pyramid jointly.

In this work, we investigate whether the model can be trained end-to-end, rather than with training being fixed at intermediate stages, even in the single image task. However,

Figure 2. Example of unconditionally generated images showing complex global structure generated by ConSinGAN.



$\delta = 0.1$ $\delta = 0.5$

Figure 3. Effect of learning rate scale δ and concurrently trained stages for a model with six stages. Images are randomly selected.

we find that training all stages leads to *overfitting* (see Figure 3), i.e. the generator only generates the original training image without any variation. We develop a novel progressive growing technique that trains multiple, *but not all*, stages concurrently while simultaneously using progressively smaller learning rates at lower stages. Since we train several stages of our model concurrently for a single image we refer to our model as 'Concurrent-Single-Image-GAN' (ConSinGAN).

Training ConSinGAN starts on a coarse resolution for a number of iterations, learning a mapping from a random noise vector z to a low-resolution image (see "Generator: Stage 0" in Figure 1). Once training of stage n has converged, we increase the size of our generator by adding three additional convolutional layers. In contrast to SinGAN, each stage gets the raw features from the previous stage as input, and previous layers are not fixed. We add a residual connection [15] from the original features to the output of the newly added convolutional layers (see "Generator: Stage 1" in Figure 1). We repeat this process N times until we reach our desired output resolution. We add additional noise to the features at each stage [19, 47] to improve diversity. In our default setting, we jointly train the last three stages of a generator (see "Generator: Stage N" in Figure 1). While it is possible to train more than three stages concurrently, we observed that this rapidly leads to severe overfitting (Figure 3).

We use the same patch discriminator [19] architecture and loss function as the original SinGAN. This means that the receptive field in relation to the size of the generated image gets smaller as the number of stages increases, meaning that the discriminator focuses more on global layout at lower resolutions and more on texture at higher resolutions. In contrast to SinGAN we do not increase the capacity of the discriminator at higher stages, but use the same number of parameters at every stage. We initialize the discriminator for a given stage n with the weights of the discriminator of the previous stage $n-1$ at all stages. At a given stage n, we optimize the sum of an adversarial and a reconstruction loss:

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

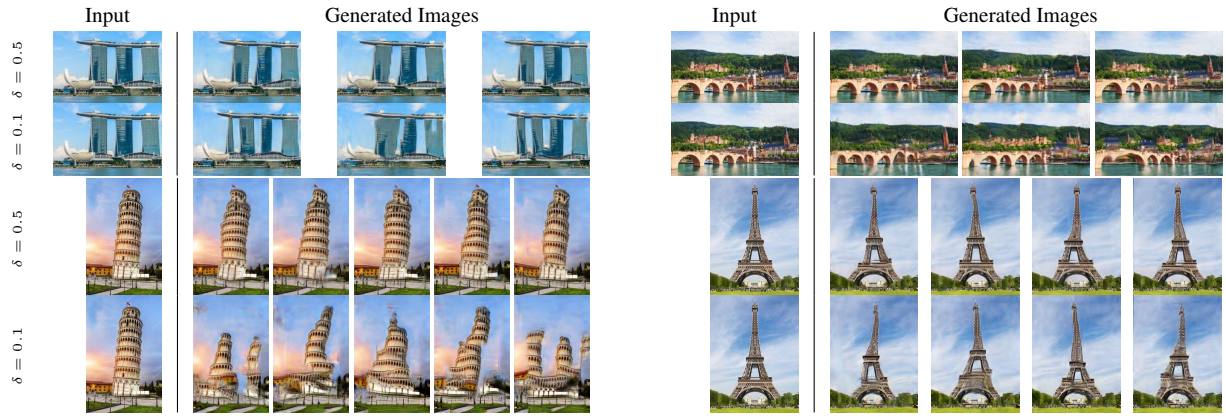


Figure 4. Effect of the learning rate scale δ during training of ConSinGAN.

$\mathcal{L}_{adv}(G_n, D_n)$ is the WGAN-GP adversarial loss [13], while the reconstruction loss is used to improve training stability ($\alpha = 10$ for all our experiments). For the reconstruction loss the generator G_n gets as input a downsampled version (x_0) of the original image (x_N) and is trained to reconstruct the image at the given resolution of stage n:

$$\mathcal{L}_{rec}(G_n) = ||G_n(x_0) - x_n||_2^2. \quad (2)$$

The discriminator is always trained in the same way, i.e. it gets as input either a generated or a real image and is trained to maximise $\mathcal{L}_{adv}$. Our generator, however, is trained slightly differently depending on the final task.

Task Specific Generator Training For each task we use the original image x_n for the reconstruction loss $\mathcal{L}_{rec}$. The input for the adversarial loss $\mathcal{L}_{adv}$, however, depends on the task. For unconditional image generation the input to the generator is simply a randomly sampled noise vector for $\mathcal{L}_{adv}$. However, we found that if the desired task is known beforehand, better results can be achieved by training with a different input format. For example, for image harmonization, we can instead train using the original image with augmentation transformations applied as input. The intuition for this is that a model that is used for image harmonization does not need to learn how to generate realistic images from random noise, but rather should learn how to harmonize different objects and color distributions. To simulate this task, we apply random combinations of augmentation techniques such as additive noise and color transforms to the original image x_N at each iteration. The generator gets the augmented image as input and needs to transform it back to an image that should resemble the original distribution.

Learning Rate Scaling The space of all learning rates for each stage is large and has a big impact on the final image quality. At any given stage n, we found that instead of training all stages (n, $n-1$, $n-2$, ...) with the same learning rate, using a lower learning rate on earlier stages ($n-1, n-2, ...$) helps reduce overfitting. If the learning rate at lower stages is too large (or too many stages are trained concurrently), the model generator quickly collapses and only generates the training image (Figure 3). Therefore, we propose to scale the learning rate η with a factor δ. This means that for generator G_n stage n is trained with learning rate $\delta^0\eta$, stage $n-1$ is trained with a learning rate $\delta^1\eta$, stage $n-2$ with $\delta^2\eta$, etc. In our experiments, we found that setting $\delta = 0.1$ gives a good trade-off between image fidelity and diversity (see Figure 3 and Figure 4).

Improved Image Rescaling Another critical design choice is around what kind of multiscale pyramid to use. SinGAN originally proposes to downsample the image x by a factor of r^{N-n} for each stage n where r is a scalar with default value 0.75. As a result, SinGAN is usually trained on eight to ten stages for a resolution of 250 width or height. When the images are downsampled more aggressively (e.g. $r = 0.5$) fewer stages are needed, but the generated images lose much of their global coherence.

We observe that this is the case when there are not enough stages at low resolution (roughly fewer than 60 pixels at the longer side). When training on images with a high resolution, the global layout is already "decided" and only texture information is important since the discriminator's receptive field is always 11×11. To achieve a certain global image layout we need a certain number of stages (usually at least three) at low resolution, but we do not need many stages a high resolution. We adapt the rescaling to not be strictly geometric (i.e. $x_n = x_0 \times r^{N-n}$), but instead to keep the density of low-resolution stages higher than the density of high-resolution stages:

$$x_n = x_N \times r^{((N-1)/log(N))*log(N-n)+1} \text{ for } n = 0, ..., N-1 \quad (3)$$

For example, with a rescaling scalar $r = 0.55$ we get six stages with the following resolutions and we observe that our new rescaling approach (second line) has more stages

Figure 5. Comparison of SinGAN and ConSinGAN.

with smaller resolutions compared to the original rescaling approach (first line):

$$25\times34, 38\times50, 57\times75, 84\times112, 126\times167, 188\times250,$$
$$25\times34, 32\times42, 42\times56, 63\times84, \quad 126\times167, 188\times250.$$

To summarize our main findings, we produce feature maps rather than images at each stage, we train multiple stages concurrently, we propose a modified rescaling pyramid, and we present a task-specific training variation.

4. Results

We evaluate ConSinGAN on unconditional image generation and image harmonization in detail.[1] For space reasons we focus on these two applications but note that other applications are also possible with ConSinGAN. We show examples of other tasks such as image retargeting, editing, and animation in the supplementary material.

4.1. Unconditional Image Generation

Since our architecture is completely convolutional we can change the size of the input noise vector to generate images of various resolutions at test time. Figure 2 shows an overview of results from our method on a set of challenging images that require the generation of *global* structures for the images to seem realistic. We observe that ConSinGAN is successfully able to capture these global structures, even if we modify the image resolution at test time. For example, in the Stonehenge example, we can see how "stones" are added when the image width is increased and "layers" are added to the aqueduct image when the image height is increased.

[1]Code: `https://github.com/tohinz/ConSinGAN`

Ablation We further examine the interplay between the learning rate scaling and the number of concurrently trained stages (Figure 3) and evaluate how varying the learning rate scaling δ (section 3) affects training (Figure 4). As we can see in Figure 3, training with a $\delta = 0.1$ leads to diverse images for most settings, with the diversity slightly decreasing with a larger number of concurrently trained stages. When training with $\delta = 0.5$ we observe a large decrease in image diversity even when only training two stages concurrently. As such, the number of concurrently trained stages and the learning rate scaling δ offer a trade-off between diversity and fidelity of the generated images.

Figure 4 visualizes how the variance in the generated images increases with decreasing δ for a model with three concurrently trained stages. For example, when we look at the top left example (Marina Bay Sands), we observe that for a $\delta = 0.5$ the overall layout of the image stays the same, with minor variations in, e.g., the appearance of the towers. However, with a $\delta = 0.1$, the appearance of the towers changes more drastically and sometimes even additional towers are added to the generated image. Unless otherwise mentioned, all illustrated examples and all images used for the user study where generated by models for which we trained three stages concurrently with $\delta = 0.1$.

Baseline comparisons We compare our model to the SinGAN [33] model in Figure 5. For SinGAN, we show the results of both the default rescaling method (8-10 stages) and our rescaling method (5-6 stages). In the first example we observe that SinGAN struggles to model recurring structures (faces) in the generated images. In the second example we observe a loss of global structure independent of the number of stages trained. Our multi-stage training helps ensure a more consistent global structure.

Figure 6 further highlights the advantages of our approach by showing a detailed comparison of the images each model generates after being trained with the new or old rescaling technique. Each column depicts three randomly sampled images from each model. We can see the positive effect of the rescaling technique for both models, regardless of the number of trained stages. Furthermore, we can see that our model retains better global coherence in both cases.

Quantitative evaluation The Fréchet Inception Distance (FID) [16] compares the distribution of a pre-trained network's activations between a sets of generated and real images. The Single Image FID (SIFID) is an adaptation of the FID to the single image domain and compares the statistics of the network's activations between two individual images (generated and real). In our experiments, we found that SIFID exhibits very high variance across different images (scores range from $1e-06$ to $1e01$) without a clear distinction of which was "better" or "worse". In this work, we focus mostly on qualitative analyses and user studies for our

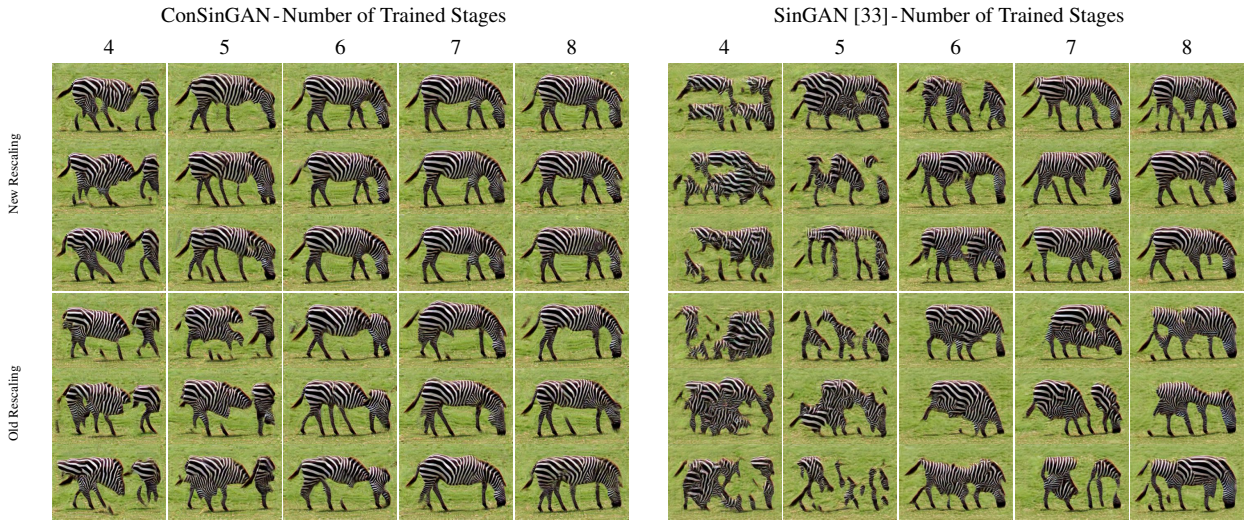


Figure 6. Comparison of the effect of the number of trained stages and rescaling method during training. Images are randomly selected.

Model	Confusion ↑	SIFID ↓	Train Time	# Stages	# Paramers
ConSinGAN	$16.0\% \pm 1.4\%$	0.06 ± 0.03	24 min	5.9	~660,000
SinGAN	$17.0\% \pm 1.5\%$	0.09 ± 0.07	152 min	9.7	~1,340,000

Table 1. Results of our user study and SIFID on images from the **Places** dataset.

evaluation but also report SIFID for comparison.

We performed quantitative evaluations on two datasets. The first dataset is the same as the one used by SinGAN, consisting of 50 images from several categories of the 'Places' dataset [44]. However, many of these images do not exhibit a global layout or structure. Therefore, we also construct a second dataset, where we take five random samples from each of the ten classes of the LSUN dataset [41]. This dataset contains classes such as "church" and "bridge" which exhibit more global structures. We train both the SinGAN model and our model for each of the 50 images in both datasets and use the results for our evaluation.

Image Diversity We evaluate the diversity in our images compared to the original SinGAN model using the same measure as SinGAN: for a given training image we calculate the average of the standard deviation of all pixel values along the channel axis of 100 generated images. Then, we normalize this value by the standard deviation of the pixel values in the training image. On the data from the 'Places' dataset, SinGAN obtains a diversity score of 0.52, while our model's diversity is similar with a score of 0.50. When we increase the learning rate on lower stages by setting $\delta = 0.5$ instead of the default $\delta = 0.1$ we observe a lower diversity score of 0.43 as the model learns a more precise representation of the training image (Figure 4). On the LSUN data, SinGAN obtains a much higher diversity score of 0.64. This is due to the fact that it often fails to model the global structure and the resulting generated images differ greatly from the training image. Our model, on the other hand, obtains a diversity score of 0.54 which is similar to the score on the 'Places' dataset and indicates that our model can indeed learn the global structure of complex images.

User Study: 'Places' We follow the same evaluation procedure as previous work [19, 33, 43] to compare our model with SinGAN on the same training images that were used previously in [33]. Users were shown our generated image and its respective training image for one second each and were asked to identify the real image. We reproduced the user study from the SinGAN paper with our own trained SinGAN and ConSinGAN models. As we can see in Table 1 our model achieves results similar to the SinGAN model. However, our model is trained on fewer stages and with fewer parameters and obtains a better SIFID score of 0.06, compared to SinGAN's 0.09. Furthermore, the images generated by ConSinGAN often still exhibit a better global structure, but one second is not enough time for users to identify this.

User Study: 'LSUN' Since the images from the LSUN dataset are much more challenging than the images from the 'Places' dataset we do not compare the generated images against the real images, but instead compare the images generated by SinGAN to the ones generated by ConSinGAN.

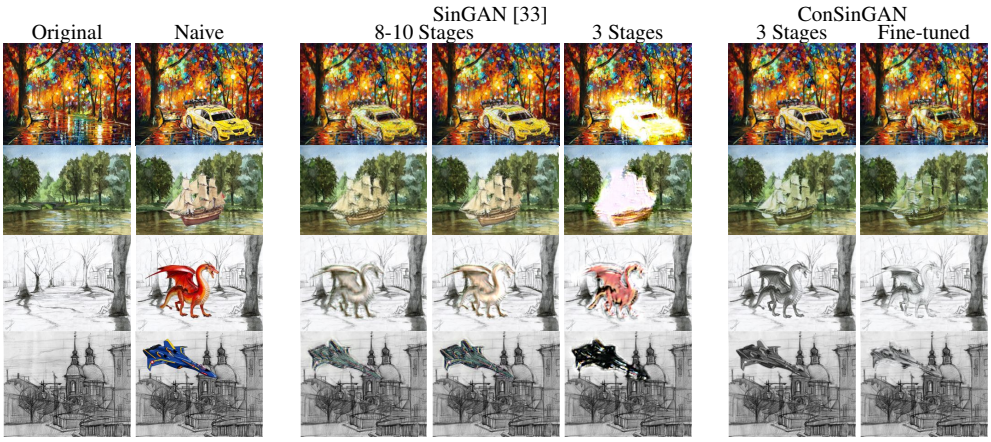


Figure 7. Image harmonization with SinGAN and ConSinGAN

Model	Random ↑	Paired ↑	SIFID ↓	Train Time	# Stages	# Parameters
ConSinGAN	$56.7\% \pm 1.9\%$	$63.1\% \pm 1.8\%$	0.11 ± 0.06	20 min	5.9	~660K
SinGAN	$43.3\% \pm 1.9\%$	$36.9\% \pm 1.8\%$	0.23 ± 0.15	135 min	9.1	~1.0M

Table 2. Results of our user studies and SIFID on images from the **LSUN** dataset.

We generate 10 images per training image, resulting in 500 generated images each from SinGAN and ConSinGAN, and use these to compare the models in two different user studies.

In both versions, the participants see the two images generated by the two models next to each other and need to judge which image is better. We do not enforce a time limit, so participants can look at both images for as long as they choose. The difference between the two versions of the user study is how we sample the generated images. In the first version ("random") we randomly sample one image from the set of generated images of SinGAN and ConSinGAN each. This means that the two images likely come from different classes (e.g. 'church' vs. 'conference room'). In the second version ("paired") we sample two images that were generated from the same training image. We perform both user studies using Amazon Mechanical Turk, with 50 participants comparing 60 pairs of images for each study.

Table 2 shows how often users picked images generated by a given model for each of the two settings. We see that users prefer the images generated by ConSinGAN in both settings and that, again, our model achieves a better SIFID. This is the case even though our model only trains on six stages, has fewer parameters than SinGAN, and takes less time to train. The images from LSUN vary in difficulty and global structure. This might explain why our model performs even better in the paired setting since this setting guarantees that we always compare the two models on images of the same difficulty. Overall, our experiments show that ConSinGAN allows for the generation of more believable images, especially when they exhibit some degree of global structure, with less training time and a smaller model than SinGAN.

4.2. Image Harmonization

We now show results on image harmonization examples and compare our model to SinGAN and Deep Painterly Harmonization [27] for high-resolution images.

Training Details We train ConSinGAN with the same hyperparameters for all images without any fine-tuning of hyperparameters for the different images. The general architecture is the same as for unconditional image generation, however, we only train the model for exactly three stages per image. We train for 1,000 iterations per stage and randomly sample from different data augmentation techniques to obtain a "new" training image at each iteration as described in section 3. When we fine-tune a model on a given specific image we use a model trained on the general style image and use the target image directly as input (instead of the style image with random augmentation transformations) to train the model for an additional 500 iterations.

Comparison with SinGAN Figure 7 shows comparisons between SinGAN and ConSinGAN. The first two columns show the original images we trained on and the naive cut-and-paste images that are the input to our trained model at test time. The next three images show the results of a trained SinGAN model, where the first two are the results of a fully trained model. We insert the naive image at all stages of the model and choose the two best results, while the third image is the result when we train SinGAN on only three stages. The final two columns show the results of the ConSinGAN. Training ConSinGAN takes less than 10 minutes for a given image when the coarse side of the image has a resolution of 250 pixels. Fine-tuning a model on a specific image takes

Figure 8. Image harmonization comparison with Deep Painterly Harmonization (DPH) and ConSinGAN on high resolution images

roughly 2-3 minutes. Training SinGAN takes roughly 120 minutes as before, since we need to train the full model, even if only some of the later stages are used at test time.

We see that ConSinGAN performs similar to or better than SinGAN, even though we only train ConSinGAN for 3 stages. ConSinGAN also generally introduces fewer artifacts into the harmonized image, while SinGAN often changes the surface structure of the added objects. See for example the first row in Figure 7, where SinGAN adds artifacts onto the car, while ConSinGAN keeps the original objects consistent. When we fine-tune the ConSinGAN model on specific images we can get even more interesting results, as, e.g., the car gets absorbed much more into the colors of the overall background. The bottom two rows of Figure 7 show results when we add colorful objects to black-and-white paintings. When training SinGAN on only three stages like ConSinGAN it usually fails completely to harmonize the objects at test time. Even the images harmonized after training SinGAN on 8-10 stages often contain some of the original colors, while ConSinGAN manages to completely transfer the objects to black-and-white versions. Again, further fine-tuning ConSinGAN on the specific images leads to an even stronger "absorption" of the objects.

Comparison with DPH Figure 8 shows comparisons between ConSinGAN, adapted to harmonize high-resolution images, and Deep Painterly Harmonization (DPH) [27]. The images have a resolution of roughly 700 pixels on the longer side, as opposed to the 250 pixels used by the SinGAN examples. In order to produce these high-resolution images, we add another stage to our ConSinGAN architecture, i.e. we now train four stages, and training time increases to roughly 30-40 minutes per image. This is in contrast to many style-transfer approaches and also DPH, which have additional hyperparameters such as the style and content weight which need to be fine-tuned for a specific style image.

We can see that the outputs of ConSinGAN usually differ from the outputs of DPH, but are still realistic and visually pleasing. This is even the case when our model has never seen the naive copy-and-paste image at train time, but only uses it at test time. In contrast to this, DPH requires as input the style input, the naive copy-and-past input, and the mask which specifies the location of the copied object in the image. Again, fine-tuning our model sometimes leads to even better results, but even the model trained only with random image augmentations performs well. While our training time is quite long, we only need to train our model once for a given image and can then add different objects at different locations at test time. This is not possible with DPH, which needs to be retrained whenever the copied object changes.

5. Conclusion

We introduced ConSinGAN, a GAN inspired by a number of best practices discovered for training single-image GANs. Our model is trained on sequentially increasing image resolutions, to first learn the global structure of the image, before learning texture and stylistic details later. Compared to other models, our approach allows for control over how closely the internal patch distribution of the training image is learned by adjusting the number of concurrently trained stages and the learning rate scaling at lower stages. Through this, we can decide how much diversity we want in the generated images. We also introduce a new image rescaling approach that allows training on fewer image scales than before. We show that our approach can be trained on a single image and can be used for tasks such as unconditional image generation, harmonization, editing, and animation while being smaller and more efficient to train than previous models.

Acknowledgements The authors gratefully acknowledge partial support from the German Research Foundation DFG under project CML (TRR 169).

References

[1] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2020.

[2] Shai Bagon, Oren Boiman, and Michal Irani. What is a good image segment? a unified approach to segment extraction. In *European Conference on Computer Vision*, pages 30–44. Springer, 2008.

[3] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[4] Sagie Benaim, Ron Mokady, Amit Bermano, Daniel Cohen-Or, and Lior Wolf. Structural-analogy from a single image pair. *arXiv preprint arXiv:2004.02222*, 2020.

[5] Urs Bergmann, Nikolay Jetchev, and Roland Vollgraf. Learning texture manifolds with the periodic spatial gan. In *International Conference on Machine Learning*, pages 469–477, 2017.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[7] Taeg Sang Cho, Moshe Butman, Shai Avidan, and William T Freeman. The patch transform and its applications to image editing. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[8] Tali Dekel, Tomer Michaeli, Michal Irani, and William T Freeman. Revealing and modifying non-local variations in a single image. *ACM Transactions on Graphics (TOG)*, 34(6):1–11, 2015.

[9] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018.

[10] Yossi Gandelsman, Assaf Shocher, and Michal Irani. Double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, volume 6, page 2, 2019.

[11] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 349–356. IEEE, 2009.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[14] Kaiming He and Jian Sun. Statistics of patch offsets for image completion. In *European Conference on Computer Vision*, pages 16–29. Springer, 2012.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] 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. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.

[17] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In *International Conference on Learning Representations*, 2019.

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 5197–5206, 2015.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

[20] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*, 2016.

[21] Animesh Karnewar and Oliver Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2020.

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, 2020.

[25] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016.

[26] Jianxin Lin, Yingxue Pang, Yingce Xia, Zhibo Chen, and Jiebo Luo. Tuigan: Learning versatile image-to-image translation with two unpaired images. *European Conference on Computer Vision*, 2020.

[27] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep painterly harmonization. *Computer Graphics Forum*, 37(4):95–106, 2018.

[28] Jiayuan Mao, Xiuming Zhang, Yikai Li, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Program-guided image manipulators. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4030–4039, 2019.

[29] Indra Deep Mastan and Shanmuganathan Raman. Multi-level encoder-decoder architectures for image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[30] Indra Deep Mastan and Shanmuganathan Raman. Dcil: Deep contextual internal learning for image restoration and image

retargeting. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2366–2375, 2020.

[31] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. Saliency driven image manipulation. *Machine Vision and Applications*, 30(2):189–202, 2019.

[32] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, pages 783–798. Springer, 2014.

[33] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019.

[34] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the "dna" of a natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[35] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[36] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[37] Tal Tlusty, Tomer Michaeli, Tali Dekel, and Lihi Zelnik-Manor. Modifying non-local variations across multiple views. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 6276–6285, 2018.

[38] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[39] Yael Vinker, Eliahu Horwitz, Nir Zabari, and Yedid Hoshen. Deep single image manipulation. *arXiv preprint arXiv:2007.01289*, 2020.

[40] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

[41] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[42] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An internal learning approach to video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2720–2729, 2019.

[43] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pages 649–666. Springer, 2016.

[44] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014.

[45] Yang Zhou, Zhen Zhu, Xiang Bai, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics (TOG)*, 37(4):1–13, 2018.

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

[47] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017.

[48] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 977–984. IEEE, 2011.

[49] Maria Zontak, Inbar Mosseri, and Michal Irani. Separating signal from noise using patch recurrence across scales. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pages 1195–1202, 2013.