EECS 504 Final Project: SinGAN Learning a Generative Model from a Single Natural Image

Ruchir Aggarwal
University of Michigan
Ann Arbor, MI
aggarwr@umich.edu

Xiangyu Peng
University of Michigan
Ann Arbor, MI
xypeng@umich.edu

Alyssa Scheske
University of Michigan
Ann Arbor, MI
scheskea@umich.edu

Abstract

This is the final project of Group 7 for EECS 504: Computer Vision. The goal of our work is to re-implement a paper from 2019 called 'SinGAN', which generates unconditional realistic images from a single natural input image with akin quality to state-of-the art Generative Adversarial Networks (GANs) trained on complete classes of datasets. SinGAN uses a pyramid architecture to allow global structure to be maintained while fine textures are also curated. Additionally, SinGAN injects noise at each level of the multi-level generator pyramid, allowing diverse samples to be created. For this project we implemented the image manipulation tasks of harmonization and editing.

1. Introduction

The goal of this project is to re-implement SinGAN [3] and generate unconditional realistic images, based on a single input image while maintaining a similar quality to state-of-the-art GANs trained on datasets, as seen in Fig. 1.

SinGAN is an unconditional generative model which can be learned from a single natural image. It is able to deal with general images without the need of a database of images from the same class. It is shown that the internal statistics of patches within a single natural image carry enough information to train a powerful generative model. Compared with previous single image GAN schemes, SinGAN is not limited to texture images, and is not conditional (i.e. it generates samples from noise). It can be applied to many image processing tasks, including paint-to-image, editing, harmonization, super-resolution and animation from a single image. Additionally, SinGAN's architecture is resolution agnostic and can thus be used on high resolution images.

GANs demonstrate the ability to generate objects and scenes. GANs are especially useful in tasks like super resolution, photo blending, photo editing, text-to-image translation or generating examples for image dataset types of applications, in addition to many more.

We investigated a few architectural contributions. We implemented simplicity where possible, especially since we never implemented all applications from the original paper. For example, many of the resizing functions used by the original paper manipulated multiple aspects of the tensor, whereas we simply implemented up or down scaling as necessary. We also adjusted the number of convolutional blocks within GAN at a single scale of the network and investigated its impact on the generated random samples. We qualitatively compared our results with the results from the original paper to understand the effectiveness our re-implementation. And we also applied SinGAN on images different than the original paper to evaluate its generality.

2. Related Work

This project is modeled after SinGAN [6], which differentiates from other related works because of it's training being task (ex: super-resolution) agnostic and on a single natural image. The authors of SinGAN were inspired by *Socher et al* [1] to use only a single image to train the GAN; however, different patch sizes at different scales of the network pyramid are used to collect information on both the global and fine-detailed features from the training image. Many other recent works have developed specific image manipulation tasks from training on a class of images. SinGAN combines these aspects while also being an unconditional GAN; unconditional GANs need no class labels for generative modeling, whereas conditional-GANs require class labels for the generator and discriminator modeling. SinGAN expands use of unconditional GANs outside of texture images with non-repetitive global structures, as previous works [8] have accomplished. SinGAN injects noise at each scale to be purely generative and to have capability beyond texture synthesis.

A GAN's generative model quality depends on the ability of its paired discriminative model. If the images are not discerned, then the generator will only train to a certain po-

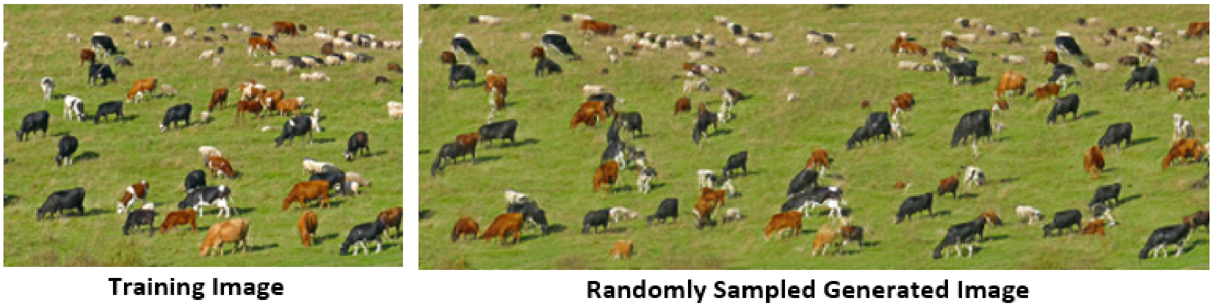


Figure 1. Generated Random Sampling of Arbitrary Size.

tential. The more realistic the generated images become, the more difficult it is for the discriminator to classify the images as fake. GANs calculate loss to comprehend these decisions during training.

Other types of generative models include 'Fully Visible Belief Networks' [9] (apply chain rule of probability to decompose the image, very slow and cannot run in parallel), 'Non-linear ICA' (convert Gaussian distributions to another space, needs invertible transformation function), 'Variational Autoencoder' [4] (marginalize image's density function, low quality images), and 'Boltzmann Machines' (define an energy function, convert to probability distribution to determine state of an image, poor performance on high dimensional images) [7]. The advantage of GANs over these other typologies is that it produces more realistic and higher quality results.

3. Method

3.1. Multi-level architecture

SinGAN is architected using a multi-level pipeline (Fig. 2). Each of the N levels of the pyramid consists of a GAN. The network accommodates N levels, and the number of levels used for each image is calculated based on the image size to keep training resources optimized. Each level of the pyramid evaluates patches from the input image in a coarse-to-fine fashion. In this project, we always resize the training image to maximal dimension 250px while maintaining its aspect ratio. Normalization is also performed on the input image to provide more effective training because the input image will then be on the same scale and intensity value as the injected noise. During training, the SinGAN model consists of a Multi-level Patch Generator and Multi-level Patch Discriminator [3]. The number of levels in the pyramid is determined by the size of the input image. The output from each pyramid level is the input to the next level; except the initial input to the generator pyramid (at the coarsest level) is a random noise image. We use the same randomized seed for generation of noise for all the levels. The scaling factor for the pyramid is re-evaluated after resizing the input image to match the maximal dimension requirements. All the generators and discriminators have similar architecture as shown in Fig. 3. At each layer of the network, the output from the previous level is added to another patch of noise and the result is pushed through a sequence of convolutional blocks of the form Conv(3 × 3)-BatchNorm-LeakyReLU and subsequently up-scaled. At the coarsest level, the number of kernels is 32 and we double the number of kernels every 4 levels as in the original implementation. The receptive field is set at the coarsest layer and remains constant (11 × 11) for every layer. Since the dimension of the image in each level layer is different, a fixed receptive field means a varying effective patch size as shown by the yellow block in Fig. 2. This above procedure is how the network is formed and trained from coarsest layer (bottom) to finest layer (top).

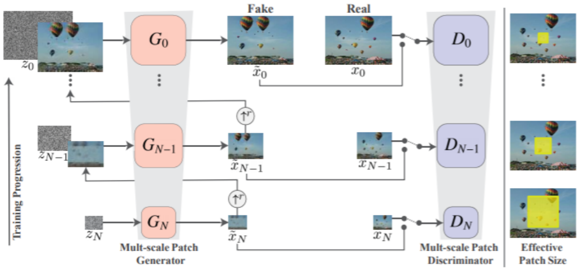


Figure 2. Architecture of the SinGAN network

The Discriminator evaluates adversarial and reconstruction losses. Adversarial training is used to achieve a realistic output at each level; the original authors executed this through WGAN-GP loss [2]. In practice, WGAN with gradient penalty enhances training stability by providing smoother gradients. The adversarial loss is defined over the whole image as opposed to over random patches to allow the network to learn boundary conditions. Reconstruction loss is calculated to ensure the reconstructed image holistically represents the input image by using root mean squared error (RMSE).

SinGAN is trained on all the overlapping patches at mul-

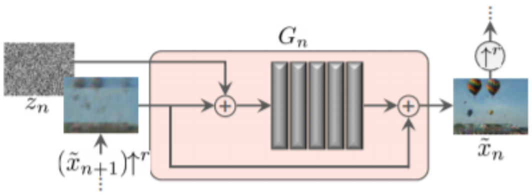


Figure 3. Single scale generation in level of SinGAN network

tiple scales of a single natural image. Because the network has a limited receptive field, it can generate new combinations of patches that do not exist in the training image.

3.2. Loss function

The training loss for GAN in each layer of pyramid contains two parts: adversarial loss and reconstruction loss.

$$\min_{G_n}\max_{D_n} L_{adv}(G_n, D_n) + \alpha L_{rec}(G_n). \tag{1}$$

Adversarial loss In each layer of the pyramid, there is a generator G_n coupled with a Markovian discriminator D_n. They play a similar role as in traditional GAN: G_n is trained to maximize $D_n(G_n(z))$, which is to generate fake image as real as possible to fool the discriminator; D_n is trained to maximize $D_n(x)$ and minimize $D_n(G_n(z))$ so it has the ability to discriminate the real image from fakes. WGAN-GP loss [2] is added to the adversarial loss to help stabilize the training process.

Reconstruction loss During the training process, we make sure that the generator can generate the original image x from a specific set of input noise map. Just like the original paper, we choose $z_N^{rec}, z_{N-1}^{rec}, ..., z_0^{rec} = z^*, 0, ..., 0$, where z^* is a fixed noise map (drawn once and kept fixed during training).

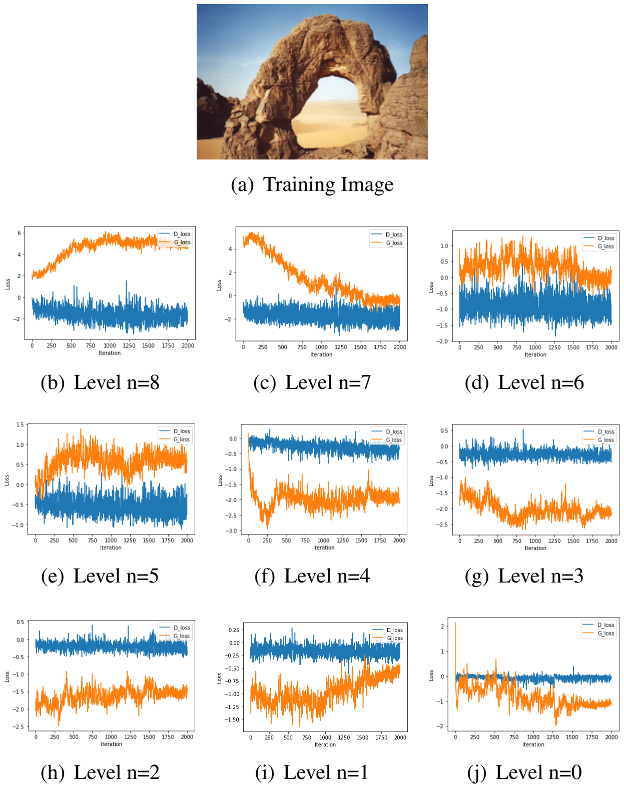
$$L_{rec} = ||G_n(0, (\tilde{x}_{n+1}^{rec})\uparrow^r) - x_n||^2. \tag{2}$$

Here, $\tilde{x}_n^{rec}$ is the generated image at the nth level when using the specific noise map. For $n = N$, $L_{rec} = ||G_n(z^*, 0) - x_N||^2$.

The reconstructed image $\tilde{x}_n^{rec}$ is used to determine the standard deviation σ_n of the input noise z_n for the next level. Specifically, σ_n is the root mean squared error (RMSE) between $\tilde{x}_{n+1}^{rec}$ and x_n, and is a measure of how much details need to be added at that level.

4. Experiments

SinGAN applications that were implemented in this work include random samples, random samples at arbitrary dimensions, harmonization, and editing. The effect of the injection scale for image manipulation refers to the process of injecting a down sampled version of an image into the generation pyramid at some scale n<N. Depending on the pyramid level selected, images will have different lower and higher resolution traits. The coarser the scale that the image is injected into, the more modifications to the larger structures; at finer injection scales, the fine textures are modified while global structures are maintained. This process is used for harmonization and editing applications. Setting the injection scale was a parameter used in experimentation.



(a) Training Image

(b) Level n=8 (c) Level n=7 (d) Level n=6

(e) Level n=5 (f) Level n=4 (g) Level n=3

(h) Level n=2 (i) Level n=1 (j) Level n=0

Figure 4. Training loss for Generator (orange) and Discriminator (blue) for each level. n=0 indicates the top level, which is the finest layer. We trained from n=8 to n=0.

Qualitatively, we compare the results of our SinGAN with the results from the original paper. In addition to the training images used by Shaham et al., we train our network on random images from Berkeley Segmentation Database(BSDS500) dataset [5]. We also analyze how the depth of the pyramid in SinGAN affects the generated image and whether introducing image at certain depth affects the preservation of relevant structures in the generated images or not.

4.1. Random Sampling

Random sampling is a major function of SinGAN, which is to generate fake images that are hard to distinguish from the real one, and it can also be used as a way to evaluate how the network is trained. If the model is trained successfully, it should be able to generate random samples that have the similar texture as the training sample, while many details can be different. We evaluated our model both qualitatively and quantitatively.

4.1.1 Qualitative Evaluation

Fig. 5 shows some of our generated random samples from our model. Because our network is fully convolutional, we are able to generate outputs of arbitrary size at test time by just changing the dimensions of the noise maps. Therefore, besides generating random samples with the same aspect ratio as the training image, we also generate outputs with scaled dimensions. After testing on lots of different images, we notice that SinGAN performs better on images that are more important as a whole, such as landscapes etc. However, if an image is detail oriented, it is more likely that the output fails to be that realistic, as shown in Fig 6. Our network is able to produce new structures while still maintaining the visual content of the original image. Interestingly, our model was not only able to preserve reflections but also synthesize it as can be seen in the case of mountains and colosseum in Fig. 5.

We also tested to see the effect of the number of convolutional blocks within the GAN at a single level of the network. When we set the number to 8, random generated samples are as shown in Fig. 7. They are almost the same as the original image except some imperceptible details like parts of the mountain or the clouds. We assume that it is probably because the network is over-fitting the training image with 8 convolutional blocks and hence it cannot generate highly varying configurations.

4.1.2 Quantitative Evaluation

In the quantitative evaluation, human ranking was used to understand our results. The process of human ranking is defined as assigning a ’real’ or ’fake’ classification to each image by asking a participant to rank each image. For this project, we averaged the results from each participant to report the human ranking value, shown in Table 1. A total of four participants were garnered. Although qualitative, it allows for comparable metrics across different images. It was noted that the best performing output images were generated through scaled random sampling.

Table 1. Human Ranking Results

Image	**Human Ranking (% assumed Real)**
Balloons	10.4%
Colosseum	25.0%
Cows	31.3%
Flowers	20.8%
Mountains	29.2%
Mountains - Arbitrary Size	49.5%
Trees	27.1%

4.2. Editing

Editing is to copy certain patches of the original image and paste them in other locations. SinGAN can effectively regenerate fine texture and seamlessly stitch those pasted parts and create realistic images.

Take the edited tree for example (Fig. 8), we first trained on the original image so that the network can generate its texture patches. Then we created our mask for edited as well as the edited input. As we can see here, the edited input is not very smooth and we can easily figure out the pasted part. We then injected the input image into a certain scale of the pyramid (not necessarily from the bottom layer) and fed forward along the pipeline. We combined the output with the original image using equation 3

$$output = (1 - mask) \times x_0 + mask \times \tilde{x}_0^{rec}. \quad (3)$$

Here, x_0 is the original image and $\tilde{x}_0^{rec}$ is the final generated image of the network with edited image as the input. The mask is a binary mask, where we set 255 on the pasted part and 0 elsewhere.

The trickier part is that which scale layer to inject. As Fig. 9 shows, injection scale has great impact on the final output. We can see that as the injection scale decreases, the outcome is more like the edited input. That is because the effective patch size of GAN in each scale layer is different. The less scale we injected the input, the smaller the structure gets modified. We found that setting the 2nd, 3rd or 4th coarsest scale layer (count from bottom to top in architecture, which corresponds to n=6, n=5, n=4 in this image) as the injection layer typically generated best result.

4.3. Harmonization

Harmonization is a technique of image manipulation that explicitly matches the style of images before blending them. Using a multi-level technique allows us to transfer the appearance of one image to another. To achieve this technique, we train SinGAN on a background image, then input an image with newly added content to be realistically blended with the style of the background. We also input a binary mask that outlines the newly added content of the input image. Similarly to editing, we investigated the effect of the input image’s injection scale. If the original SinGAN’s results (Fig.11) are compared to our re-implementation of SinGAN (Fig.12), we observe very similar quality at each scale. The best results are achieved at scales 2, 3, or 4 because the newly added content maintains its structure while it is converted to the background’s style.

We evaluated a second image, *’Seaside’*, on our re-implementation of SinGAN (Fig.13). This image was

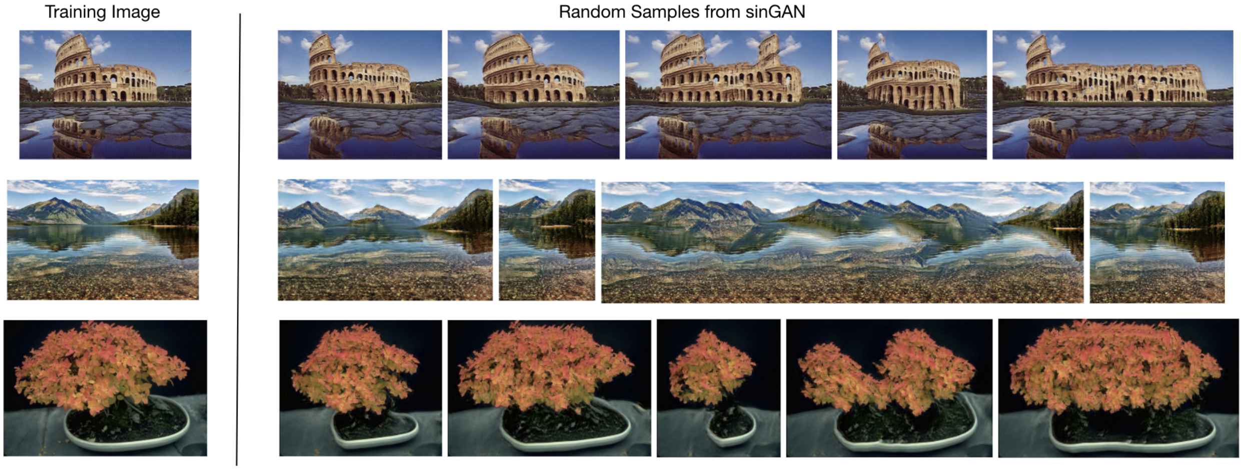
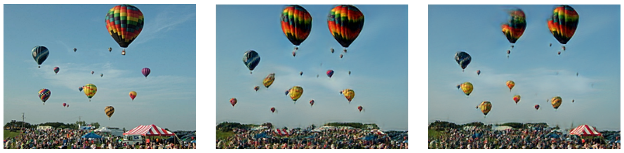


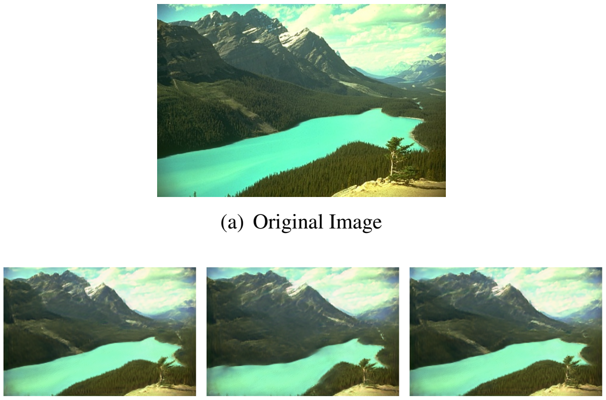
Figure 5. **Random Sampling.** The left column is the original image which is used to train the network. The right column contains random samples the well-trained network can generate. We can even generate images that are in different size as the original one since the model is fully convolutional. As we can see, in the generated samples, same patch distributions are maintained while new structures and object configurations are depicted.

smaller and thus trained on less levels in the pyramid. We qualitatively observed that the less scales trained in an image, the sharper effect of harmonization on the output. At desired scales, SinGAN does not overly-blend the new con-



(a) Training Image (b) Good Result (c) Bad Result

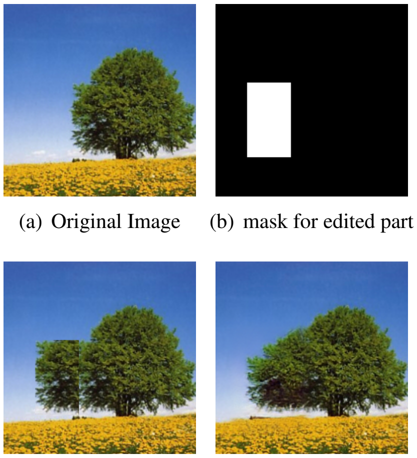
Figure 6. For images that require details, the failure rate of SinGAN output can increase. For example, balloon needs perfect contour so that generated fake (c) is not easily distinguishable from the real image. Still, we can generate large amount of outputs and we can hopefully find some realistic generated samples like (b).



(a) Original Image

(b) Random Samples

Figure 7. Generated random samples by the model containing 8 convolutional blocks each Generator and Discriminator. The random samples are almost same as the training image.



(a) Original Image (b) mask for edited part

(c) Edited Input (d) SinGAN output

Figure 8. **Editing.** From the original image (a), we used a binary mask (b) to copy and paste a patch and create an edited input(c). We injected the downsampled version of this edited input into an intermediate layer of the network (trained on (a)). The final output generated from equation 3 can be like (d), which is very smooth and realistic.

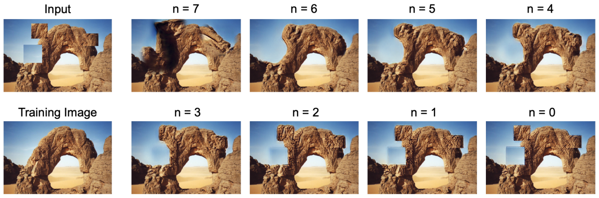


Figure 9. Effect of the injection scale on Editing.

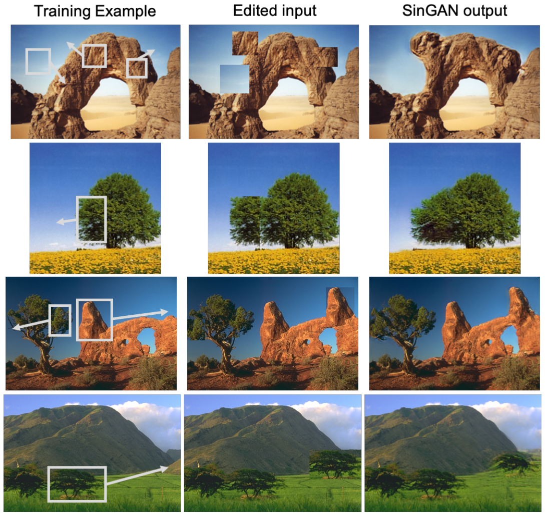


Figure 10. Some sample outputs for **Editing** generated by our model.

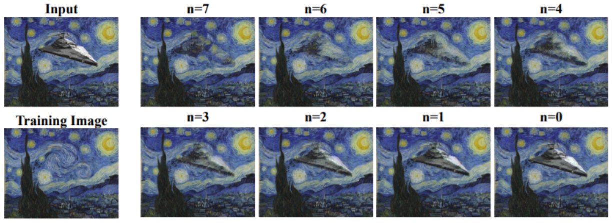


Figure 11. SinGAN's original harmonization of 'Starry Night' with various injection levels

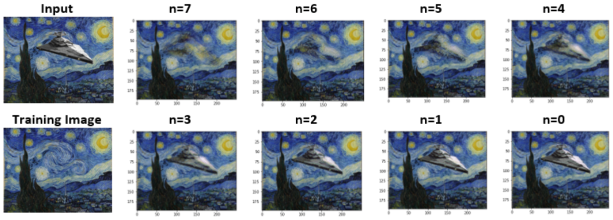


Figure 12. Our harmonization of 'Starry Night' with various injection levels

tent of the input image. We see the output image contains contrast, texture, noise, and blur to reconstruct the output image.

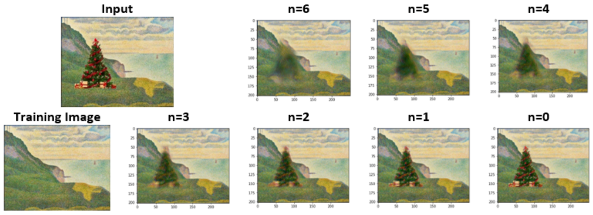


Figure 13. Our harmonization of 'Seaside' with various injection levels

5. Conclusions

SinGAN is a powerful tool. It only trains from patches within a single natural image, meaning that less time is needed for training over a network that uses a class of images. Because noise is injected during generation and the results are purely generative, SinGAN is also able to perform multiple image manipulation tasks. For this project, we successfully re-implemented SinGAN to generate distinct realistic samples. We used SinGAN to perform harmonization and image editing tasks. From investigating the injection scale level on the manipulation, we discovered that a few levels of random noise before injection help the image look more realistic. These factors create GANs that are more robust, realistic and practical.

References

[1] P. I. Assaf Shocher, Shai Bagon and M. Irani. Ingan: Capturing and remapping the "dna" of a natural image. In *arXiv preprint arXiv*, page arXiv:1812.00231, 2018. 1

[2] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 2, 3

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1, 2

[4] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *in Proc. 8th Int'l Conf. Computer Vision*, pages 416–423, 2001. 3

[6] T. R. Shaham, T. Dekel, and T. 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. 1

[7] K. Sudhir. Generative adversarial netowrks- history and overview, June 2017. [Online; posted 21-June-2017]. 2

[8] R. V. Urs Bergman, Nikolay Jetchev. Learning texture manifolds with the periodic spatial gan. In *arXiv*, page 1705.06566, 2017. 1

[9] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016. 2