Inpainting with Single Generative Adversarial Nets

Final Project Report
Deep Learning MVA

Megi Dervishi
École Normale Supérieure - Ulm
megi.dervishi@ens.fr

Abstract

*This project explores SinGAN (**Sin**gle **G**enerative **A**dverserial **N**etwork)* [1] *which was a novel architecture proposed by Ref. [23]. Unlike previous GANs used in vision, SinGANs are trained on a single image. The architecture contains a pyramid of fully convolutional GANs, each responsible for learning the patch distribution at a different scale of the image [23]. Such structure allows its application to different image manipulation tasks. In this project I reproduce the key results (quantitative and qualitative) presented for random sampling,harmonization, editing and paint-to-image tasks on various images. Furthermore, I extend the application of SinGAN to in-painting and quantitatively and qualitatively analyze the performance of the model.* [2]

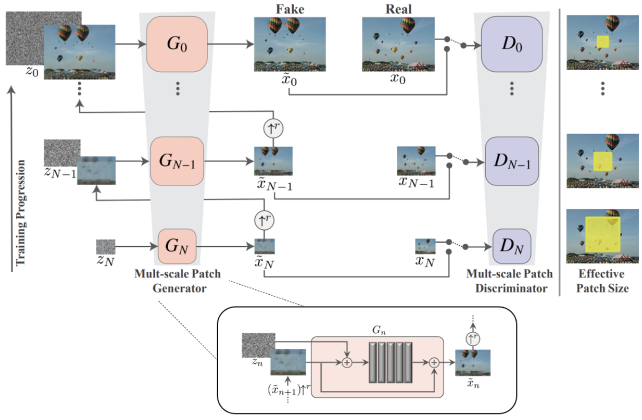

Figure 1: SinGAN architecture. (Up) Multi-scale pipeline of fully convolutional GANs. (Down) Single scale generation.

1. Introduction

Inpainting is a very old problem which used to be solved manually with artists painting by hand in cracks, scratches or spots to restore different works of art as they deteriorated in time. However with the birth of digital images and the wide variety of tools used to manipulate them, inpainting digital images has become a central problem in computer vision [20]. Digital inpainting started at the beginning of the 21st century where Efros [10] and Bertalmio [6] launched the efforts in digital inpainting using Markov modelling and geometric based approaches respectively [20]. These two original papers led to a flurry of improvements [2, 9, 4, 20]. In recent years there has been a shift from 'traditional' algorithms to deep learning methods [20] with the introduction of convolutional neural networks (CNNs) [12] and generative adversarial networks (GANs) [11] which have become the new state of the art [20]. However as noted in Ref. [20] the performance of these new algorithms is strongly dependent on the dataset used to train them, and current state of the art methods usually rely on datasets such as Paris-StreetView and ImageNet [26], Places2 [27], Celeb-A [1] and many others [20]. However, recently a new category of deep learning algorithms, self supervised learning models (SSL), has gained a lot of interest in the research community [19, 25, 21]. As such, a logical next step in inpainting is to find a way of using the state of the art techniques (GANs, CNNs) in a self-supervised fashion, this is what I aimed at answering with this project.

2. Problem Definition

Inpainting can be defined as follows [20]. Denote by $\mathcal{I}$ the clean complete image then denote by $\mathcal{M}$ any mask on that image. The inpainting problem consists in trying to recover the clean image $\mathcal{I}$ for the corrupted image $\mathcal{I} \cdot \mathcal{M}$. Suppose that every pixel of the image is completely independent of each other then it is clear that the inpainting problem is unfeasible since we have no information on the missing pixels. The opposite extreme case where every pixel is per-

[1] Official code implementation at: https://github.com/tamarott/SinGAN
[2] My code is available at: https://github.com/MegiDervishi/singan_res

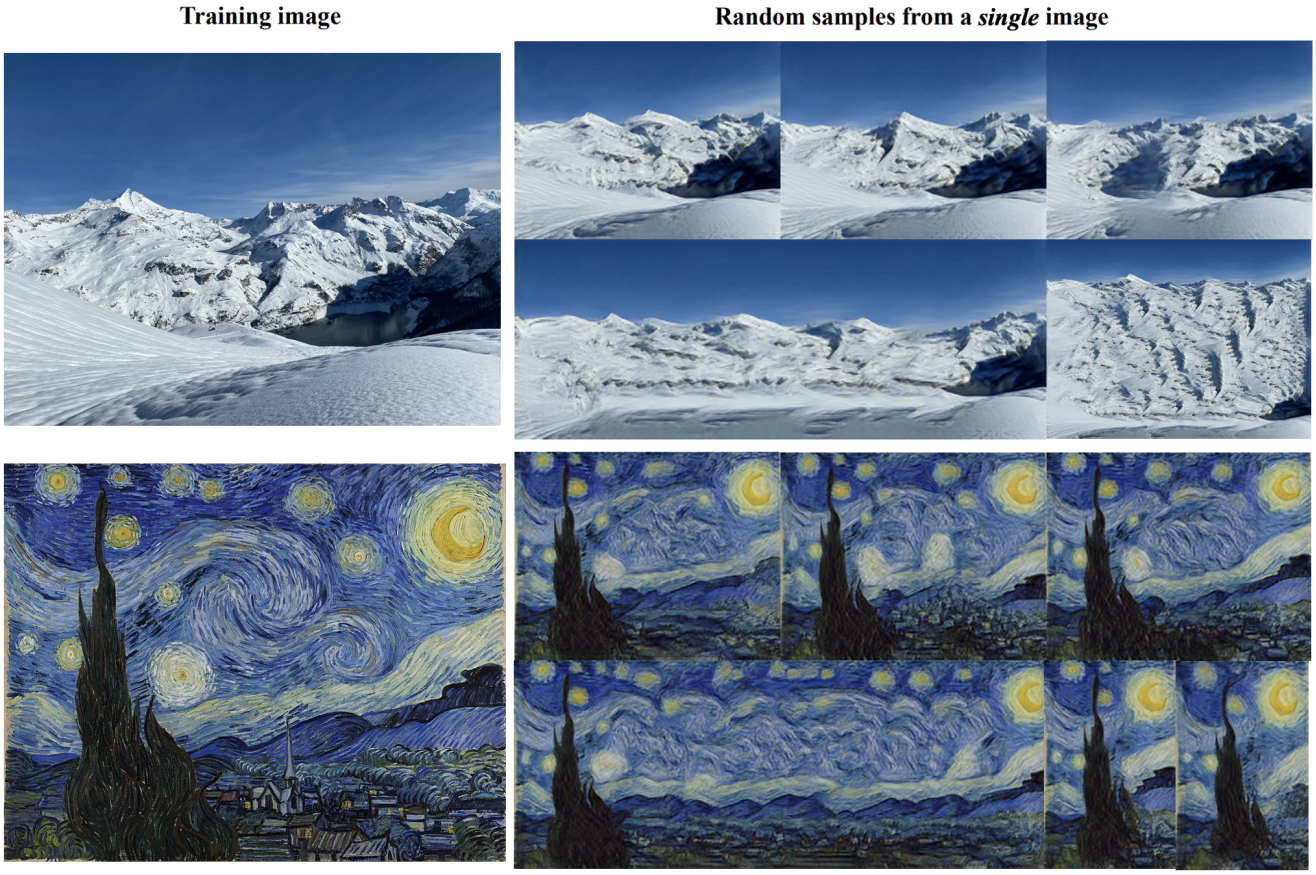

Figure 2: The SinGAN architecture trained on the Training image (left) generates new random samples (top right) by feeding different random latent variables at the coarsest scale. Leveraging the adaptability of the network architecture we can also generate samples of different sizes (right bottom row). Note however that this can sometimes lead to failure cases such as the squeezed sample of the mountain picture.

fectly correlated with each other is a trivial problem since we can simply fill the missing pixels following the general correlation rule. Hence we see that the whole problem consists in finding a way to efficiently learn/leverage the information/correlation present in the clean pixels in order to estimate as accurately as possible the missing ones. CNNs and GANs have proven to be able to very efficiently extract information/correlations from images [20] and it is the reason why they are the current state-of-the-art in computer vision. This is why I decided to use GANs in order to tackle inpainting. The current state-of-the-art in self-supervised GANs for vision tasks is SinGAN [23] which was the starting point for this project. I started by reproducing all the results of Ref. [23] before using it in a novel way in order to perform inpainting in a self-supervised manner.

3. Related Work

The original paper of SinGAN presented many different tasks that this architecture was able to perform, such as Random Sampling, Harmonization, Super-Resolution, Animation or Editing [23] and this goes to show the versatility of this architecture. Since the original paper there have been a number of modifications to the architecture [29, 13, 7, 15] that aim at using it to solve some other specific computer vision problems such as deblurring [18], super-resolution reconstruction [8, 30], explainable models [28] or video modelling from a single GIF [3]. However, using SinGAN for image inpainting has not yet been done to the best of my knowledge [20], although some implementations of closely related problems have been studied [17, 22].

4. Methodology

The following section describes the architecture of SinGAN along with the tasks of Random Sampling, Paint-to-image, Harmonization and Animation introduced in the original paper; the inpainting task and the experimental details used to evaluate the performance of the model.

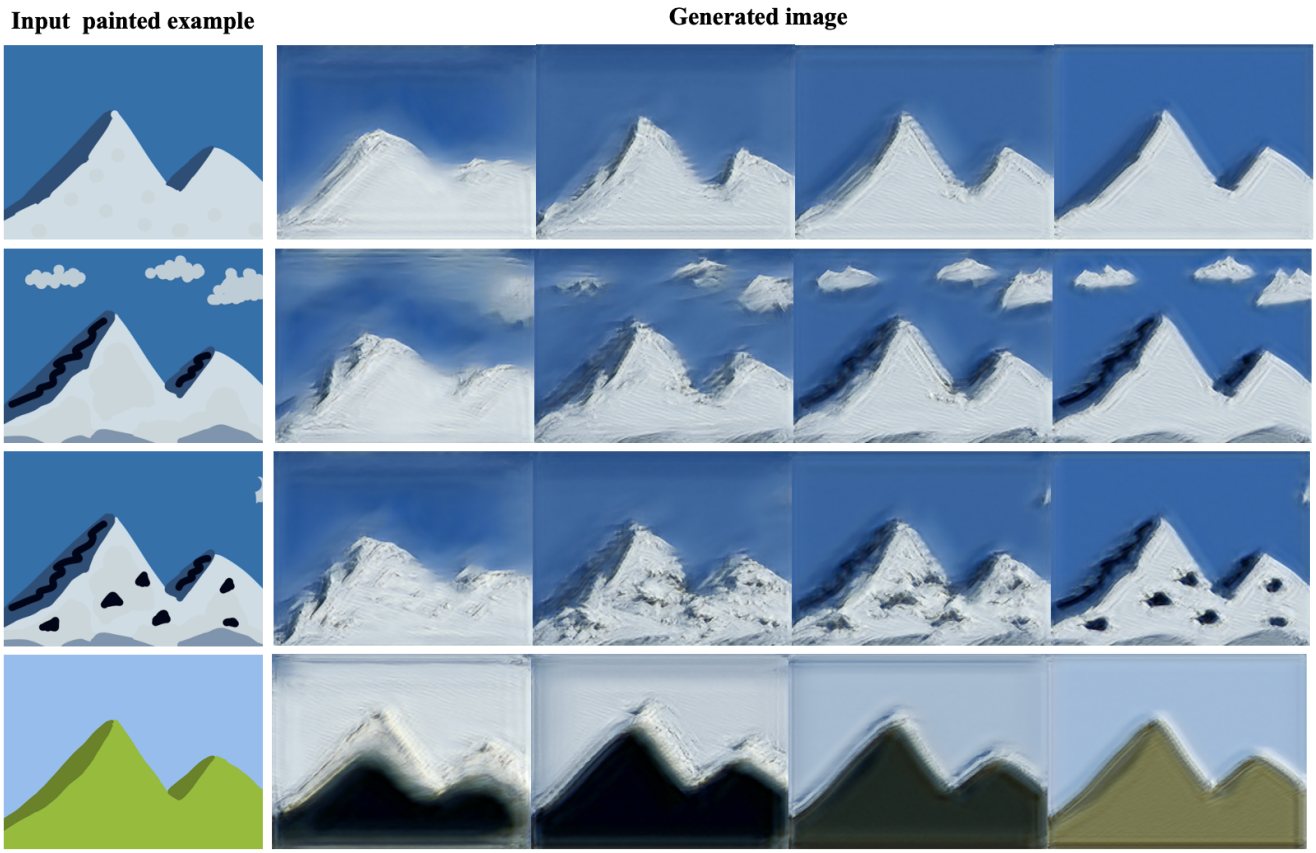

Figure 3: After training the model on the snowy mountain image shown in Figure 2, we insert the input painted image (left) downscale it and feed it to a chosen scale of our architecture in order to transform the painted example to something closer to the actual image. The images on the right were obtained by respectively inserting the downscaled painted image at the first, second, third and fourth (from left to right) scale of the SinGAN architecture.

4.1. SinGAN

The first step of the project was to understand and reproduce the original SinGAN architecture [23]. The architecture, which is presented in Figure 1, is a pyramidal GAN architecture where training and inference are done on downsampled patches of the original image in a coarse-to-fine fashion. At each level of the pyramid the discriminator learns how to distinguish patches from the generated image given by the generator compared to the ground truth and in contrast the generator learns how to fool the discriminator. The pyramidal structure allows the architecture to capture the fact that correlations in the image can be a multi-scale function. In other words, the correlations between pixels will behave differently according to the scale we are looking at. This is what gives SinGAN such a strong expressive power. The loss function used to train the model is given by

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

where G_n (resp. D_n) is the generator (resp. discriminator) at scale n and α is a hyper-parameter called the reconstruction loss weight. The first term of the loss is called the adversarial loss and is the WGAN-GP [14] loss which in essence is a usual patch discrimination loss with a regularization term constraining the gradient of the discriminator. The second term of the loss is called the reconstruction loss which is simply the L_2 square distance to the original image at every scale, this term ensures that the model is able to reconstruct the original image.

Random Sampling.
Once I have trained the model to reconstruct similar images to the clean original image I can generate many different random samples 'close' to the original image by feeding different noise maps at different chosen scales in the pyramidal structure, as can be seen in Figure 2.

Paint-to-Image.
After having trained the model on an image, I feed a

downsampled painting to one of the coarser scales of the model which will then keep the coarse structure of the painting but replace the finer details to make it more similar to the actual image as shown in Figure 3.

Harmonization.
The model is trained on a given background image. After simply naively pasting another image/object on a section of the background I feed this new image to an intermediate scale of the model which will harmonize the heterogeneous image with the background. Figure 6 demonstrates such results.

Animation.
In order to create an animated clip from a single snapshot I do the following: SinGAN is trained on the snapshot, however instead of feeding the model decorelated noise we are going to do something slightly different. We are going to perform a random walk in z-space, thus allowing us to generate similar images using SinGAN which gives rise to a short animated clip. The results can be found in the attached git repo [3]

4.2. Inpainting

The big difference of inpainting with respect to the previous methods is that when we perform inpainting in a self-supervised manner we are assuming that we do not have access to a clean image. Otherwise, we could simply train on the clean image and harmonize the masked sections as is usually done in Ref. [23]. In the inpainting case we have to train on the corrupted image. This means that we are forced to make a first assumption: the majority of a given image is not corrupted. Indeed if the corruption is predominant then the model will be unable to learn the features of the clean image. If we naively feed the corrupted image to the SinGAN model then the model will confuse the masked parts with whichever patches are most similar to it in the rest of the picture. In order to address this we implement two solutions. Firstly during training we will not train the model on the masked regions. Practically this is done by removing the contributions of the masked regions from the loss function. Secondly we help the model by filling the masked sections in such a way that the model will be correctly biased to harmonize them with the neighboring pixels.
The first idea that I implemented to fill the masked sections was to simply replace the masked pixels with the average color of the whole image, however as we have just said, we want to bias the model towards harmonizing with the *neighboring* pixels. Hence a much better prior is to fill the masked section with the average color of the pixels in close proximity to the section. Finally, I test a more advanced prior by using the Telea inpainting algorithm [24]. The Telea inpainting algorithm [24] consists in performing a random exploration of the masked section starting at the borders and filling progressively the missing pixels by using the known or previously estimated ones.

[3] https://github.com/MegiDervishi/singan_res

4.3. Experimental details

For the above tasks, SinGAN is trained for 2000 epochs and for 9 scales on a Nividia GTX-1060 GPU. The training time for an image takes around 5 to 6 hours. A high-resolution image is downsampled to around 250 or 300 pixels as otherwise I could not run the model. The learning rate is $5e-4$ for both the generator and discriminator. The number of layers is 5 and the kernel size is 3.

5. Evaluation

5.1. Quantitative evaluation.

A common metric for evaluating GANs on computer vision tasks is the Fréchet Inception Distance (FID) [23, 16]. The FID is defined by modelling the features generated by a GAN with a multidimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ and similarly the features of the training images with $\mathcal{N}(\mu_{\text{train}}, \Sigma_{\text{train}})$. Then the FID is defined by [16]

$$\text{FID} = ||\mu - \mu_{\text{train}}||_2^2 + \text{Tr}\left[\Sigma + \Sigma_{\text{train}} - 2\left(\Sigma^{1/2}\Sigma_{\text{train}}\Sigma^{1/2}\right)^{1/2}\right] \quad (2)$$

However this metric could not be used in these cases because when using only one image it makes no sense to model the feature space with a multidimensional Gaussian. Furthermore, for SinGAN we are more interested in the patch statistics rather than the full picture similarities. Hence why Ref. [23] introduces a new metric to compare results for self-supervised GANs performing computer vision tasks, the Single Image Fréchet Inception Distance (SIFID). The idea being that instead of computing the FID between the features of the generated image and the real image (which would yield only one vector for each image) we compute the FID between the features computed after the convolutional layer i.e. before the second pooling layer (which yields one vector per patch for each image). This allows to model the 'distance' in patch distribution between the generated image and the real image which is the metric that we are interested in. Furthermore, in Ref. [23] the authors looked at how the SIFID correlates to confusion for their AMT perceptual study and they saw that they where strongly anti-correlated. Hence a low SIFID indeed corresponds to a low-similarity and hence high-confusion. Indeed when computing the average SIFID across all scales for the mountain image compared to the sea image shown in Figures 9, 10 and 11 the Sea scores around 13 SIFID while the mountain scores around 9 SIFID which corresponds to

my qualitative observations as will be detailed in the next sections. However I found that while the SIFID is able to capture 'general impressions' it can only slightly make the difference between an acceptable and a very good solution as is shown in Table 12, where the SIFID for the average of FMM methods (detailed in the next section) are nearly identical while their qualitative performance is very different as we will see in the following section.

5.2. Qualitative evaluation.

5.2.1 Random Sampling

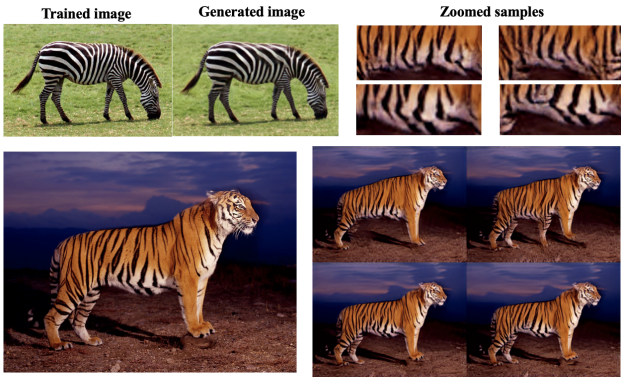

Figure 4: Detailed example of the limiting behavior of random sampling. After training on the image on the left we get the random samples on the right identically to what was done in Figure 2. Notice however that while the network is able to modify the stripe pattern of the zebra, it is not able to significantly modify the stripe pattern of the tiger. This is due to the fact that the training zebra image was rich in stripe orientations, while the training tiger image contained only vertical stripes. Hence the model is not able to 'create' new combinations.

The simplest application of SinGAN is random sampling, however it gives us a lot of insight into the functioning of the architecture and its limitations. We can see in Figure 2 various examples of random sampling. Notice that for the top-row mountain samples the network is able to capture the general structure of the image, snowy foreground and mountainy background, while being able to perform modifications to the actual image. In the first sample we can see that the network added a new mountain peak while in the second it changed completely the silhouette of the mountain and in the last it creates a shadowy valley on the left hand side of the mountain. Similarly with Van Gogh's starry night the model is able to preserve the general components (city/tower/moon/sky) while modifying their composition, by changing the turns in the sky, the shape of the tower or city lights. Now by simply feeding different shapes of latent space noises we can generate arbitrarily sized images. Notice that the model is able to replicate the original image structure over bigger sizes and also understand what is 'important' in the image when cropping an image like in the Van Gogh painting. However this also has some failure cases, as is shown for the cropped version of the snowy mountain, where the network focuses too much on the 'wrong' patches of the image. This can seem almost magical, how can SinGAN generate 'new art'? have we finally created a sentient AI with tact on par with Van Gogh? Of course this is not the case and this is what is shown in Figure 4. The network, in reality, is not able to generate anything that is really 'new'; the network can only reproduce patterns/patches or combinations of such patterns/patches that it has seen before. Indeed notice in Figure 4 that while the network is able to create different stripe patterns for the zebra, it is very vaguely able to do so for the tiger. This is due to the fact that the training image for the zebra contains many different stripe patterns (horizontal/vertical) while the tiger image contains only vertical stripes. Hence for the tiger, the network is only able to perform small stripe-patterns modifications probably by using the stripes that it observed on the legs of the tiger, however it is unable to change the general stripe pattern of the tiger. Here we see the first bottleneck of our model for inpainting. The model has no general 'awareness' hence if I occlude a door in an image of a house and if no other doors are present, then the network will be completely unable to inpaint the hole, while a human would, since a human would know that there must be a door to the house.

5.2.2 Editing

As seen in Figure 5 the network is able to seamlessly integrate the shuffled patches with the rest of the images. The problem at hand is very similar to inpainting and hence is a positive foreshadowing that the inpainting will work. Furthermore, I noticed one crucial ingredient which will intervene later in inpainting. The content of the shuffled patch greatly influences what does the network decide to replace it with. Indeed the blue patch is always modified to be part of the sky, the white patch always becomes snow, and so on. Hence, the network is very sensitive to what is put in the missing patches/switched patches and when doing inpainting I will have to be careful with what to initialize the missing patches with.

5.2.3 Paint-to-image

Another impressive application of the SinGAN model is it's ability to transform a painted image into an actual image that looks quite close to the original image it trained on, as can be seen in Figure 3. Notice (especially in the third row) that the network is able to reconstruct a very realistic looking image which preserves the general structure of the

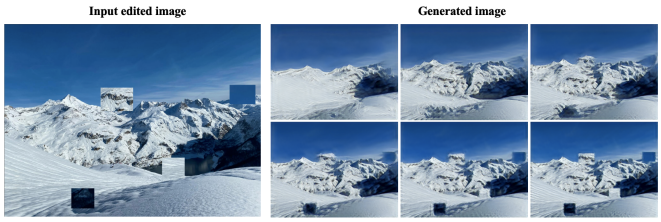

Figure 5: Example of an editing modification which SinGAN then tries to resolve.

input painted image, namely the twin peaks. However as concluded in section 5.2.2, SinGAN is extremely sensitive to the colors of the original painted image. Indeed any dark spot becomes a rock or a shadow, light blue the sky and white the snow, stressing once again the importance of initialization of the masked hole when performing in-painting.

5.2.4 Harmonization

Harmonization shows one of the last key ingredients that will be necessary for inpainting. The idea is to train the model on a clean training image then naively insert an intruding image and downscale the resulting image to insert it at a chosen scale in the SinGAN architecture. What we notice in Figure 6 is that the behavior of the model is completely different according to the scale at which we insert the input image. If we insert the image at the coarsest scale the network will almost completely overwrite the intruder but if we insert it at the finest scale the network will change near to nothing. In a way at coarser scales the network is more prone to completely ignoring what is within the 'disturbing patch' and simply replace it with the general structure around it, while at finer scales the network is more prone to completely ignoring the general structure and simply focus on the small details of the 'disturbing patch'. This will prove very important for inpainting because simply replacing the 'disturbing patch' with what the general structure dictates cannot give very good results. On the other hand, so long as we fill the patch with a good estimate, then small fine scaled modifications performed by the network will nicely fit the patch with it's surrounding.

5.3. Inpainting

As hinted in the previous sections, the performance of in-painting relies on three factors: the complexity of the background, the patches and what is inserted inside them. To test the later, I try naive in-painting whose results are shown in Figure 7.

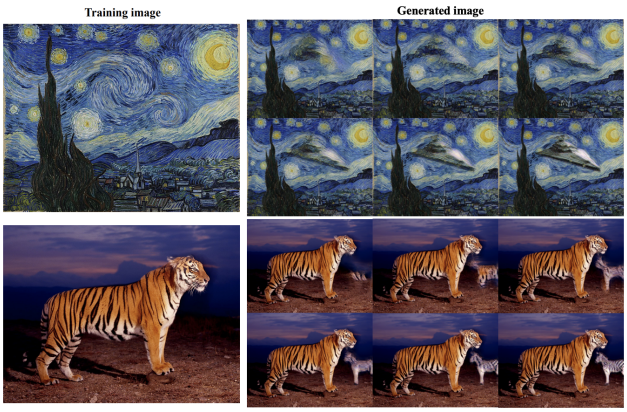

Figure 6: Example of the harmonization problem. After training the model on the training image (left) we naively paste an intruding image on top of the background, then we downscale this image and feed it to a chosen scale of the SinGAN architecture (right, one to six, left to right, top to bottom). Notice that according to the scale at which we insert the image the network will either blend the intruder almost completely with the background or will perform only slight modifications. Notice how in the second scale for the tiger the network is more or less able to transform the zebra into a tiger.

Figure 7: Example of a naive inpainting strategy. After simply masking the desired patches with white we train the model on the rest of the image and then ask it to inpaint these patches. We see that the model does not manage to get rid of the initial 'data' of the patches (i.e. that they were white boxes).

5.3.1 Naive Inpainting

If we simply try to do inpainting by leaving the masked regions completely empty then unless we use a scale which is coarser than the size of the corrupted region (hence effectively ignoring whatever is contained within the corrupted patch) the results are quite bad as seen in Figures 7 and 8. This is because, as we have seen with the previous applications, at finer scales the model cannot ignore what is contained in the corrupted patch and hence will try to uniformize the content of that patch with it's surroundings which can lead to non-sensical result if the corrupted patch

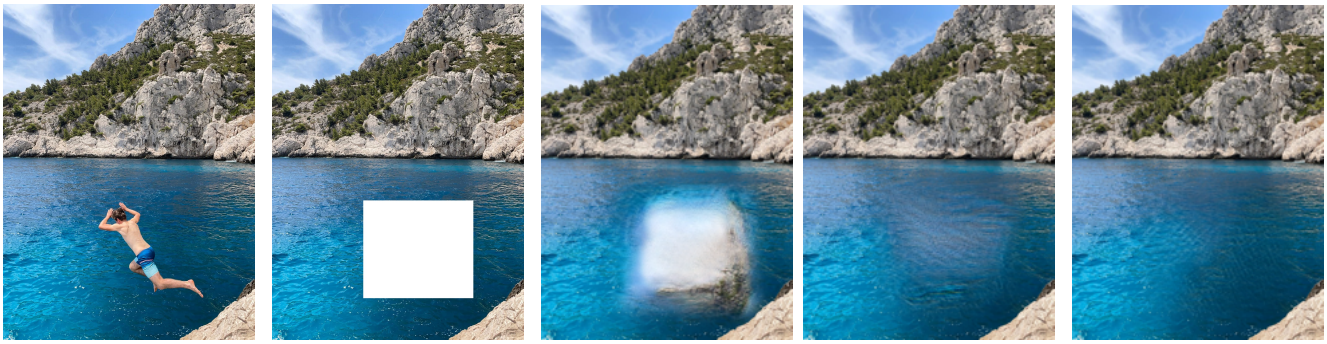

Figure 8: An example of inpainting a large mask. From left to right we have, the original picture, the masked picture, the inpainted image using no initialization, the inpainted image using the average prior, the inpainted image using the FMM prior. Notice how the choice of how to initialize the masked pixels now becomes crucial.

has nothing to do with it's neighbors. Hence the first simplest idea to get decent results was to fill the masked patches with the average color of the image.

5.3.2 Average-color Inpainting

Using the average-color inpainting improvement one studies the effect of the background and patches on the performance of SinGAN on different masked-images. Figure 9, 10 and 11 show the qualitative result of such effects. In particular, in Figure 9 (up.) the patches are concentrated in the sea, which is a less complex area and has (in general) a more uniform color and structure than the mountain in Figure 10. The model hence has an easier time to inpaint over the missing patches of the sea compared to those of the mountain. For example the model has trouble incorporating the triangular geometry of the peak of the mountain when using a box patch. This raises the question of how to improve in-painting when the background complexifies.

As it turns out, I notice that the model is able to handle irregular patches ref. Figure 11 (up). Hence one could perform instance segmentation and only mask the object and not the bounding box. This gives more information to the model on how to fill the patch and hence leads to better performance.

Nonetheless notice how if the masked-patch is along an edge, like the patch on the mountain peak which is half-way on the mountain and half-way on the sky, then filling the patch with a uniform color taken from it's surroundings is not a good estimate. This is what led to the following final implementation.

5.3.3 Improved Inpainting.

Instead of arbitrarily filling the masked patch with the colour of its surrounding I now tried to use existing 'classical' inpainting techniques such as the Fast Marching Method (FMM) [24] or Bertalmio's Navier-Stokes (NVS) [5] in order to produce a very good estimate of the inpainted region before giving it to the SinGAN network. Notice that while I used FMM one could repeat the experiment with any possible inpainting method. The flexibility of the SinGAN architecture basically allows us to improve any previous inpainting done by any other algorithm. As we can see in Figures 9, 10, 11 (down.) and 8 the inpainting done using FMM+SinGAN yields results which are completely indistinguishable from a real image. Furthermore, FMM has another advantage: it can be used to fill in irregular (not a simple geometric shape) holes. This means that for example we can mask more fitted shapes to the object we are trying to remove, hence allowing us to keep more of the original clean image and giving the network more information to work with. In doing so we help the network and it can usually perform better as can be seen in Figure 11.

6. Conclusion

In conclusion, in this project I have studied extensively the SinGAN architecture and expanded its applications to inpainting. I showed that SinGAN can be used in order to fine-tune existing classical inpainting techniques such as the Fast Marching Method (FMM) [24]. More generally it would be interesting to see if the generalization power and versatility of SinGAN allows it to improve any existing inpainting method. Furthermore, following this reasoning one idea would be to apply SinGAN iteratively. This can be particularly useful when the size of the masked patch is relatively big to the normal image, as one inpaints the masked patch progressively. While I wanted to explore this possibility for the project I didn't have the time or resources (training one model takes me 6h) in order to be able to test this idea. Hence with more time and compute it would be interesting to see if doing so allows us to reach even better results. Although I expect that such a structure will be prone to creating artifacts I think it should overall perform better.

References

[1] Semantic image inpainting with deep generative models. 1

[2] Proceedings ninth ieee international conference on computer vision. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages i–, 2003. 1

[3] Rajat Arora and Yong Jae Lee. Singan-gif: Learning a generative video model from a single gif. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1310–1319, January 2021. 2

[4] Nazre Batool and Rama Chellappa. Detection and inpainting of facial wrinkles using texture orientation fields and markov random field modeling. *IEEE Transactions on Image Processing*, 23(9):3773–3788, 2014. 1

[5] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. 7

[6] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 417–424, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 1

[7] Xi Chen, Hongdong Zhao, Dongxu Yang, Yueyuan Li, Qing Kang, and Haiyan Lu. SA-SinGAN: self-attention for single-image generation adversarial networks. *Machine Vision and Applications*, 32(4), July 2021. 2

[8] Guojian Cheng, Fulin Zhang, and Xinjian Qiang. Super-resolution reconstruction of rock thin-section image based on singan. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 9, pages 786–790, 2020. 2

[9] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004. 1

[10] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038 vol.2, 1999. 1

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 1

[12] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Li Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks, 2017. 1

[13] Songwei Gu, Rui Zhang, Hongxia Luo, Mengyao Li, Huamei Feng, and Xuguang Tang. Improved singan integrated with an attentional mechanism for remote sensing image classification. *Remote Sensing*, 13(9), 2021. 2

[14] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. 3

[15] Xiaoyu He and Zhenyong Fu. Recurrent singan: Towards scale-agnostic single image gans. In *Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering*, EITCE 2021, page 361–366, New York, NY, USA, 2021. Association for Computing Machinery. 2

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. 4

[17] Basile Van Hoorick. Image outpainting and harmonization using generative adversarial networks, 2020. 2

[18] Harshil Jain, Rohit Patil, Indra Deep Mastan, and Shanmuganathan Raman. Blind motion deblurring through singan architecture, 2020. 2

[19] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021. 1

[20] Jireh Jam, Connah Kendrick, Kevin Walker, Vincent Drouard, Jison Gee-Sern Hsu, and Moi Hoon Yap. A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, 203:103147, 2021. 1, 2

[21] Kriti Ohri and Mukesh Kumar. Review on self-supervised image recognition using deep neural networks. *Knowledge-Based Systems*, 224:107090, 2021. 1

[22] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 2

[23] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image, 2019. 1, 2, 3, 4

[24] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9, 01 2004. 4, 7

[25] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021. 1

[26] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis, 2017. 1

[27] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Free-form image inpainting with gated convolution, 2019. 1

[28] ZiCheng Zhang, CongYing Han, and TianDe Guo. Exsingan: Learning an explainable generative model from a single image, 2022. 2

[29] Ming Zheng, Pengyuan Zhang, Yining Gao, and Hang Zou. Shuffling-SinGAN: Improvement on generative model from a single image. *Journal of Physics: Conference Series*, 2024(1):012011, sep 2021. 2

[30] Guangyuan Zhong and Huiqi Zhao. "zero-shot" super-resolution based on singan. In *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, pages 883–888, 2021. 2

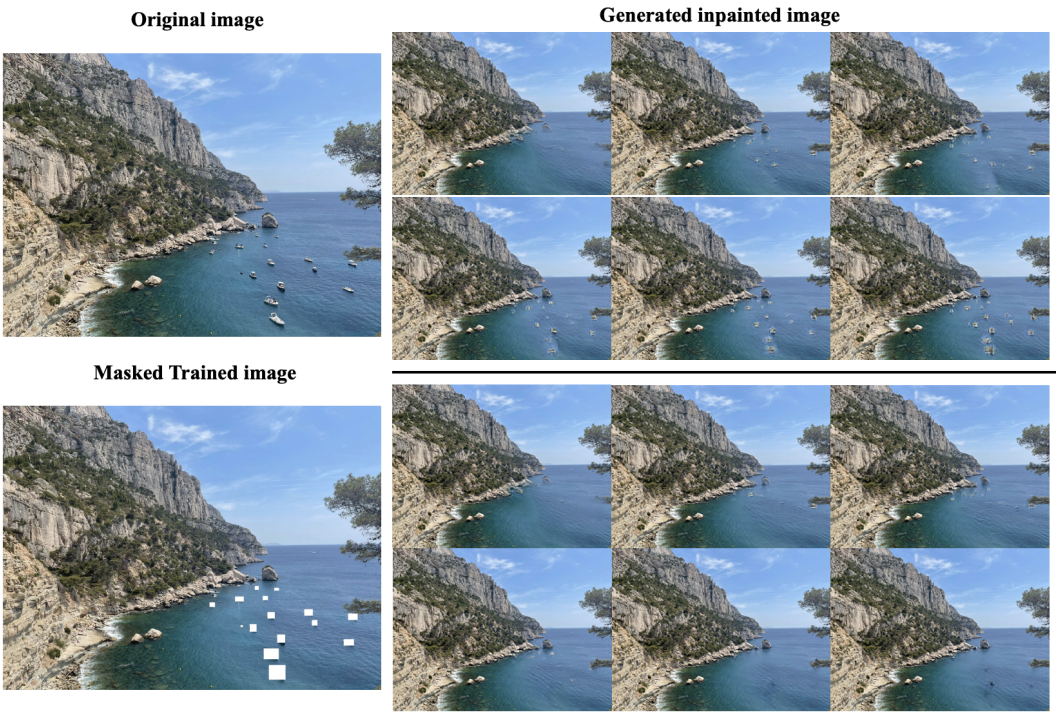

Figure 9: Inpainting example where the masked regions occur on a uniform background. In this example the model trains on the Masked trained image (bottom left) and then inpaints it using an average prior (top right) or a FMM inpainting prior (bottom right). We see that for the average prior the model is able to almost perfectly inpaint the image at low scales but some artefacts appear at medium to high scales. With the FMM prior the model is able to inpaint at all scales except the highest one where some artefacts appear.

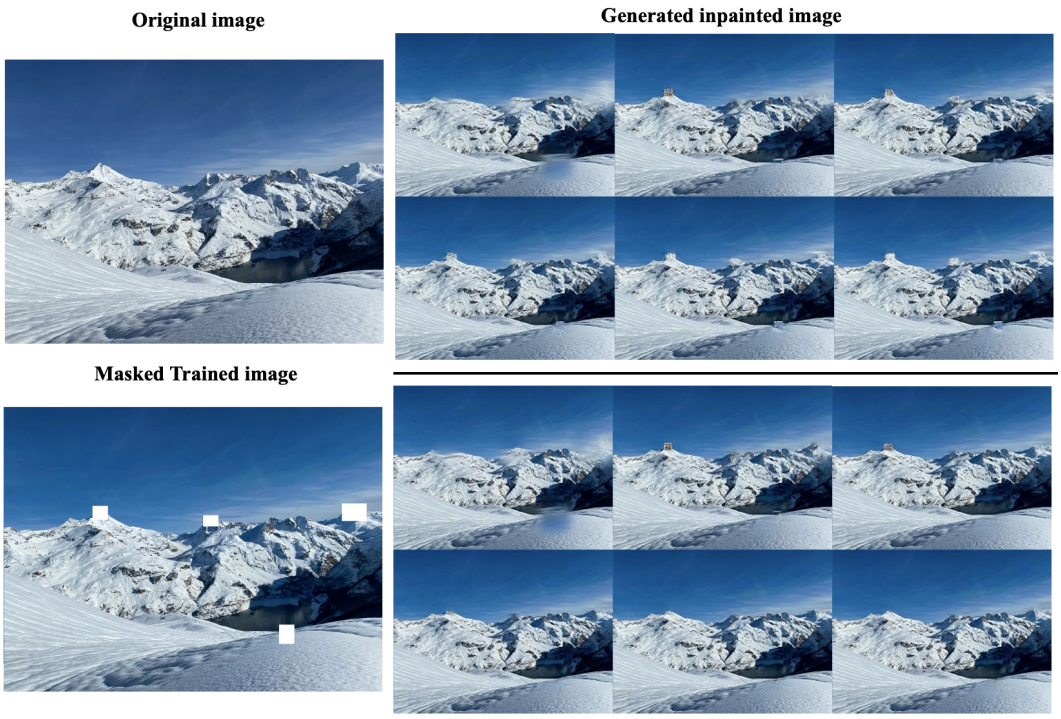

Figure 10: Inpainting example where the masked regions occur on a non-uniform background. We follow the same as what is done in Figure 9. However notice that in this case while the FMM prior is able to give very good inpainting results at medium/high scales, the average prior is only able to give decent results for low/medium scales.

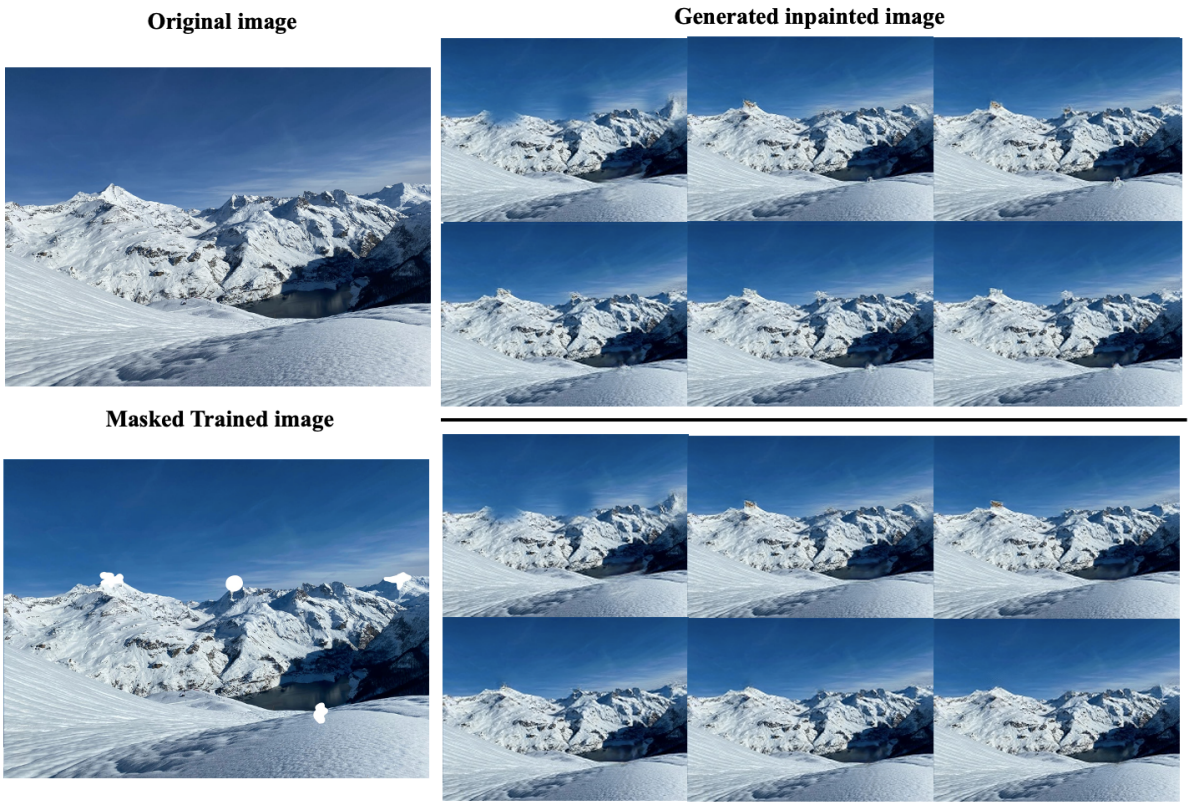

Figure 11: Inpainting example where the masked regions are irregular and occur on a non-uniform background. We follow the same as what is done in Figures 9 and 10. Notice that both the average and FMM priors are able to handle irregular holes. In fact the results of the FMM prior are nearly indistinguishable from the original image.

Image Type vs Injection Scale	1	2	3	4	5	6	7	8
Sea (Average)	13.27	13.27	13.27	13.27	13.27	13.26	13.26	13.27
Sea (FMM)	13.27	13.27	13.27	13.27	13.27	13.27	13.27	13.27
Snow Mountain Regular Holes (Average)	8.95	8.94	8.94	8.94	8.94	8.94	8.94	8.94
Snow Mountain Regular Holes (FMM)	8.95	8.94	8.94	8.94	8.94	8.94	8.94	8.94
Snow Mountain Irregular Holes (Average)	8.95	8.94	8.94	8.94	8.94	8.94	8.94	8.94
Snow Mountain Irregular Holes (FMM)	8.95	8.94	8.94	8.94	8.94	8.94	8.94	8.94

Figure 12: SIFID metric for the different images described in Figures 9, 10 and 11. Notice that in concordance with our qualitative results the sea SIFID is much higher than the snow mountain SIFID. However there is only a very slight difference between the average and the FMM prior. The SIFID is not able to capture the difference between an acceptable solution and a very good solution.