SinGAN: Learning a Generative Model from a Single Natural Image

Charbel-Raphaël Ségerie CHARBEL-RAPHAEL.SEGERIE@HOTMAIL.FR
Hugo Laurençon HUGO.LAURENCON@GMAIL.COM

Abstract

GANs allow to generate artificial images, but suffer from the defect of needing a large database to generate quality images. In this study, we test the performance of the SinGAN algorithm, allowing to generate artificial images by training on a single image. We analyze the results of the different functions of SinGAN on many different tasks, such as harmonization with inpainting for example. We verify that the generated images are not simple copies of the training image by defining adapted metrics based on the result of the PatchMatch algorithm. These metrics prove to be stable in practice and reflect the creativity of an image generation algorithm.

1. Introduction

This study is carried out for the validation of the Imagerie Numérique course of the master MVA at ENS Paris-Saclay during the year 2020/2021.

The purpose of GANs is to "learn" the distribution of the data so that new artificial data, similar to the real ones, can be created. This is achieved through the simultaneous training of two adversarial networks, one seeking to discriminate real data from artificial ones, and the other seeking to generate artificial data that can fool the discriminator.

However, this training requires a lot of data and is very unstable. Different techniques have then emerged to improve the stability of this algorithm. For example, it has been shown that GANs use a cost function that globally invokes the optimization of the Jensen-Shannon divergence. Arjovsky et al. used instead the Wasserstein distance to measure the difference between two probability distributions, which is a weaker distance and therefore converges more easily in practice. This requires that the discriminator belongs to the 1-Lipschitz functions, which is achieved through weight clipping in the original paper. Studies followed and Gulrajani et al. proposed instead a gradient penalty (WGAN-GP).

The SinGAN paper [6] constructs a GAN based on these improvements, with the difference that it is now trained only on a single image, which thus solves the problem of obtaining a huge database. The goal is to learn the structures of the image, and to be able to reconstruct an artificial one that is similiar. In this study, we will first present in more detail the functioning of this algorithm. We will then experiment and compare the output results of SinGAN with the PatchMatch [2] algorithm to check if the generated images are not simple copies of patches of the original image. We will also test SinGAN's harmonization function on several denoising and inpainting tasks, as well as the influence of the scale. Finally, we will quickly mention other uses of SinGAN, such as super resolution or animation.

2. Presentation of the different algorithms

2.1 SinGAN networks

2.1.1 General description of the SinGAN networks

To balance the fact that we have only one image in the database during training, SinGANs use a patch representation of the image, so that they have many small images. The algorithm is based on multi-scale, and the SinGAN architecture is described in Fig. 1.

We have a sequence of pairs of generators and discriminator $(G_i, D_i)_{0 \leq i \leq N}$, whose generators are 5-layer residual convolutional networks, which we train one by one starting with (G_N, D_N) and ending with (G_0, D_0). We start by training the GAN (G_N, D_N) corresponding to the lowest resolution of the image, then we keeps this fixed for the training of the second GAN (G_{N-1}, D_{N-1}) corresponding to a lower resolution of the image, and so on until arriving at the highest resolution of the image with (G_0, D_0). The propagation is thus done by recurrence.

Let us take the first GAN (G_N, D_N). We give a noise z_N to the generator G_N and we train it to produce an image in low resolution $\tilde{x}_N$. Now let's take a downsampled version of the initial image by a factor r^N, from which we will extract patches and make the D_N discriminator learn that these patches are indeed from the real image. At the same time, D_N will be trained to indicate that the patches of the image produced by the generator G_N are artificial, while G_N will be trained to fool D_N. We only show patches to the discriminator and not the whole image because since we only have one image, it would automatically learn it and it would not allow G_N to produce images different from this original image.

To go to the next step, we can forget the D_N discriminator which is no longer useful, and we keep G_N fixed. We sample a noise z_N and give it to the generator G_N to get $\tilde{x}_N$. We then upscale $\tilde{x}_N$ by a factor of r, then give it as input to the generator G_{N-1} with another noise z_{N-1} to get $\tilde{x}_{N-1}$ as output. Again, the generator G_{N-1} will be trained so that the patches of $\tilde{x}_{N-1}$ fool the discriminator D_{N-1}, and D_{N-1} will be trained to distinguish between the patches of the artificial image produced by G_{N-1} and the real patches of the original image downsampled by a factor r^{N-1}. The size of the images we manipulate changes at each layer, but the size of the patches does not really change, so the lower layers will have a coarser function by changing the overall image, while the higher layers will change the details.

At each step, a reconstruction loss is added to the GAN loss during training. For noises all equal to 0 except the first z_N which is fixed, we want to force the generator to return the real downscaled image.

We can then iterate this process until we find a generated image with the same resolution as the initial image.

2.1.2 Other applications of the SinGAN networks

The SinGAN networks allow several applications related to the generation of artificial images.

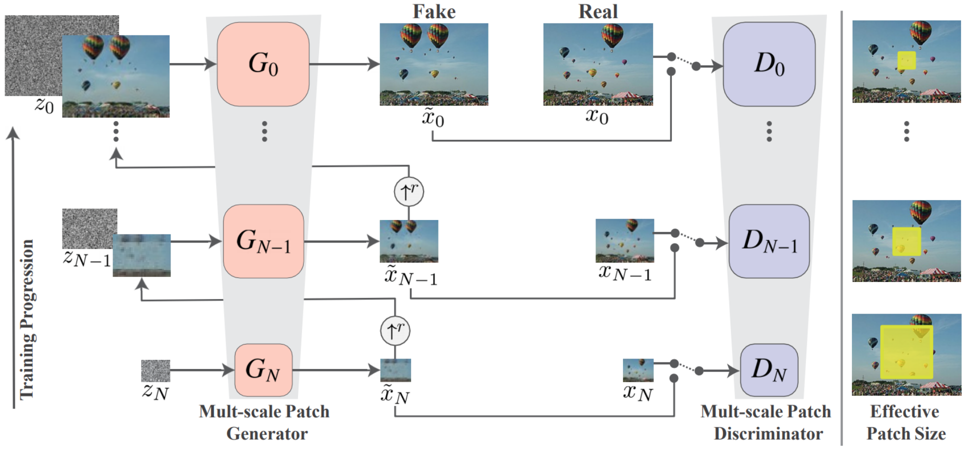


Figure 1: SinGAN's multi-scale pipeline. Figure from [6].

We can also do Paint-to-Image by giving to a pre-trained generator a downscaled image of a drawing. The output will be an image preserving the structure of the drawing, while having the style of the training image.

Harmonization can also be done by including an object in an image in a natural way. This is also done by inputting to the pre-trained generator a downscaled version of the image where an object is roughly pasted onto the background image.

Finally, Single Image Animation can be done by animating the artificial images obtained by continuously varying the noise z.

2.2 PatchMatch algorithm

2.2.1 General description of the PatchMatch algorithm

The patchmatch algorithm allows from two images, usually an "original" image and a "modified" image, to find the arrangement of the patches of the original image in order to reformat the modified image from the patches of the original image. Formally the algorithm returns the offset function for each of the image patch to find the original image patch closest to the modified image patch.

For computational cost reasons, the algorithm does not perform a brute force calculation of distances but uses initialization and propagation heuristics in order to complete the task in real time.

Initialization: The algorithm initializes the function of the closest neighbors in a random way in order not to get stuck in a local minimum.

Propagation: The algorithm traverses the patches of the modified image from left to right and from top to bottom. Then it uses the heuristic that the closest patches should probably have the same offset.

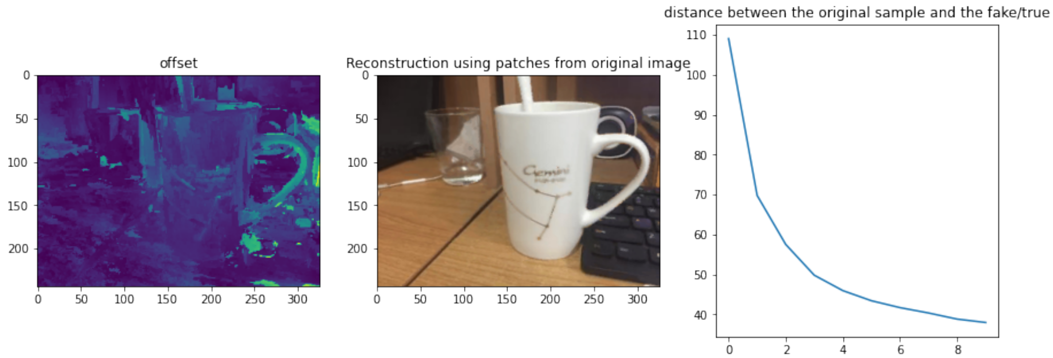


Figure 2: Example of a typical curve showing PatchMatch convergence.

Random Search: The algorithm alternates between neighbor propagation and random search in an exponentially small window. The hope is that the patches having found a good match will be propagated quickly to the whole image, which is empirically always the case in a few iterations.

2.2.2 Other applications of the PatchMatch algorithm

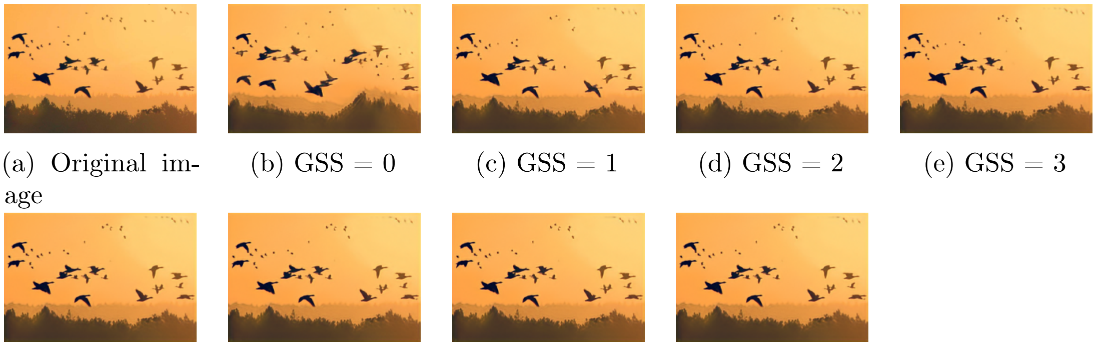
By using the PatchMatch algorithm, the authors of the article managed to manipulate the images in many different ways. We did not focus on the capabilities of PatchMatch for structured image distortion, but it should be noted that this algorithm is very versatile.

2.2.3 Our use of PatchMatch

The use of PatchMatch also makes it possible to inpaint the image distortion without any deep learning technique, and therefore allows you to have a baseline of what can be done using only the information present on the image without any learning. The only prior used is the locality prior via the patches. And we can play on the strength of this regularization prior by playing on the size of the patches. The only way to inpaint for PatchMatch is to copy the patches, so we can use PatchMatch to compare the inpainting resulting from a copy with the inpainting resulting from a deeper/original work than a copy, which we hope to be the case with SinGAN.

2.2.4 Number of iterations

We have set the stopping criterion by a number of iterations fixed beforehand for all our experiments. The article proposes to use a number of iterations of 4 or 5. We agree that the visual result is fixed from the first iterations. However, the convergence of the total distance to the L2 norm continues to decrease even after the 5th iteration. Therefore, we decided to use a number of iterations equal to 10. An example of the curve showing PatchMatch convergence can be found on Fig. 2.



(a) Original image (b) GSS = 0 (c) GSS = 1 (d) GSS = 2 (e) GSS = 3

(f) GSS = 4 (g) GSS = 5 (h) GSS = 6 (i) GSS = 7

Figure 3: Artificial images outputted by the generator starting from different scales. GSS: Generator starting scale.

2.3 Inpainting

Image inpainting is the completion of an image in which there are unknown regions. This is useful in case one wants to remove an object from a photo.

Newson et al. proposed an image inpainting algorithm based on a multiscale minimization of patch-based energy approach. The functional they seek to minimize ensures that a good solution would have, for each patch inside a hole, a small distance between this patch and its closest neighbor in a known region.

The optimization is done in practice using an iterated alternating minimization, first to find the nearest neighbors and then to reconstruct the image. The nearest neighbors search is efficiently implemented using an approximation thanks to the PatchMatch algorithm [2].

3. Experiments and results

3.1 Artificial image generation with SinGAN

3.1.1 Examples of artificial images generated by SinGAN

We trained SinGAN on an image of birds. We generated artificial images starting from different scales for the generator. We notice that the most different image from the original one is the image obtained starting at scale 0. Starting at scale 1 or more makes the modifications become very slight in this case. The results are on Fig. 3.

3.1.2 Theory of originality in image creation

Motivations and Philosophy of Approach We want to measure how original SinGAN is compared to other image creation techniques. We will use in an undifferentiated way the words "originality" or "absence of copies". Indeed if we take the example of the image of birds generated by SinGAN visible in the previous section, we would prefer that the result of SinGAN does not result from a copy of a bird such as a pasted copy. It could be argued that being able to identify the different moveable/permutable objects in an image without degrading the image quality is a skill that is part of originality, but only if the cut pieces are sufficiently diverse.

This will allow to differentiate metrics evaluating originality modulo the permutations from metrics evaluating originality encompassing the permutations.

The metrics that we will propose will not be used in any case to quantify the visual or aesthetic quality of an image. We know that for bird images, SinGAN does a remarkable job and confuses even the user. Thus we will take for granted in this part that the images produced by SinGAN are visually realistic in appearance.

However, for different realistic-looking images (the fake images) derived from the same original image (original), we can ask ourselves which is the most original fake image *from the reference* of the original image. We will therefore use the PatchMatch algorithm to find different metrics of the offset field.

The metrics we will propose should not be used to train a neural network. Indeed, according to Goodhart's law [7], "when a metric becomes an objective, it ceases to be a good metric". Thus, our metrics are only used in a test phase, and not in a train phase without the network being aware of the use of this metric.

Moreover, it is difficult to transcribe in mathematical language or by a simple scalar a note of originality. We will therefore define several metrics that we will want with very different definitions and motivations from each other in order to make our study more robust.

Finally, it is possible that we may be mistaken in the mathematical formulation of our originality metrics. By observing the results, are we going to measure the adequacy of the relevance of our metrics to the common meaning of the word "copy" or the originality of SinGAN compared to a classical inpainting? Unfortunately, the answer to this question can only be circular, so we will assume that our metrics are relevant if they are consistent with each other and consistent with common sense.

Choice of different metrics Our metrics are functions of the offset space in an output space (often $\mathbb{R}$). At the experimental level, one must be careful to differentiate the position from the offset. The position tends to grow to the right and downwards, while the offset is null in expectation.

Global Metrics :

- **NbSet** : The number of image fragments with a constant offset. Intuition: remember that the offset is calculated using PatchMatch. When there is propagation over two similar areas, the offset will be constant over the areas of similarity. Thus, we can count the similarity zones by looking at the different number of constant zones per piece. In an image made of a single, completely constant piece, one is faced with a pure copy of the original image. In a very fragmented image with many pieces, the original image has been correctly shuffled, and we get many pieces. Thus, the more original the image is, the more the number of pieces increases. This metric is invariant of the permutations between the pieces because we are only counting the pieces, regardless of their position.

- **L2Norm** : The integral of norm 2 of the offset gradient. Intuition: the previous metric measures the number of pieces but we can imagine a dual metric: instead of measuring the number of pieces we can make the integral of the border area. And thanks to the integral of the gradient norm (the gradient allowing to identify discontinuities), we can now weight differently large and small discontinuities. The integral of the gradient norm 2 is also a global metric. This metric increases when there are permutations between pieces, especially since pieces at one end of the image are pasted adjacent to pieces at the other end of the image. The same can be done for the L1 norm. One can compose by the log the norm to compare more easily different orders of magnitude.

- **AngleHistogramm** : Intuition: If we take a bird in the center of the image, we want to say that the more original the bird is, the more the offsets (the vectors that indicate the origin of the different parts of the bird) come from different places in the image, like a patchwork coming from many different places. In this case, we can then look at the histogram of the directions of the offsets. And for a bird from an adequate cultural mix, one expects the angles to be evenly distributed and therefore one expects to get a uniform histogram. The histogram of angles is not really a metric, but a visual information allowing to quickly locate a preferred direction, such as a Dirac that would represent a bird copied and pasted because all the vectors of origin of the bird would be parallel to each other. One of the difficulties of this metric is the fact that the image is not circular but square which induces biases in the distribution of angles. Another bias comes from the distribution of angles for birds taken from the sides: the vectors will then come from the opposite side. But one can hope that by summing over the whole image the biases of the sides cancel each other out. This is why it is not easy to define a histogram of angles per bird. This measure is therefore global and depends on permutations.

- **SumGrad** : Norm of the sum of the gradient offsets: For a bird, if the original patches come from many different directions, the sum of the gradients should approximately cancel each other out. And conversely if all the gradients come from the same direction, the sum should be large because all the vectors add up. But this is a measure subject to a lot of variance, and it is better to average it for a batch of images than to use it and interpret it for a single image. It is a global measure of originality. Empirically the variance is too high, so we didn't really use this metric.

Local Metric :

- **MeanDotProduct** : Average of the scalar product of the normalized offsets for an object in the image. Intuition: if we take an object in an image, for example a bird, we can try to see if the original offsets of the patches composing the bird have the same direction. Thus, one way to compress the information from the histograms of the angles into a single value is to take two by two the scalar product of the normalized offsets.

patch size	iterations	L2Norm	L1Norm	NbSet
3	10	17.4	13.3	11251
5	10	15.8	12.8	9239
7	10	14.9	12.3	8572
9	10	17.1	12.6	7704

Table 1: Originality metrics and patch size.

> The larger the averages of the scalar products, the more the vector comes from the same direction and the less original the bird will be. Of course, as for histograms, this value introduces a bias for birds on the sides. But we can say that for different images with fixed bird positions but with different birds, since they are the same positions, they have the same bias, so we can compare the two images with each other. Mean dot product is a measure that can approximate the entropy of Angle Histogram. But we did not use entropy because entropy is not bounded while mean dot product naturally gives a real between 0 and 1.

It should be noted that the global metrics NbSet and L2Norm are extensive. While the other metrics are intensive and do not explicitly depend on the image size.

Influence of PatchSize The calculation of our metrics depends on the offset which is calculated via PatchMatch which itself depends on the meta parameter of the patch size.

We can see the influence of the patch size on the following table by varying the size of the patch and fixing the two images. Results are shown in Tab. 1.

The patch size defines the length of one side of the patch square, and patch sizes are always odd so that a center pixel can be easily defined.

- We can see that NbSet decreases regularly with the size of the patches, which is normal because the bigger the patches are, the less there are different patches.
- SumGrad also decreases regularly but this phenomenon is difficult to interpret.
- The L2Norm and L1Norm standards are relatively stable.

We can see that the metrics are not totally invariant to the size of the patches. In order to be rigorous we will take care to take the same patch size for two different comparisons.

Since a patch size of 5 seems to give good results, this will be the size we will work with for the rest of the article unless otherwise stated.

3.1.3 Visual analysis of originality via patchMatch

Baseline: analysis of the result of a copied and pasted image We can see in Fig. 4 that the portion of the image copied and pasted is correctly identified by the PatchMatch algorithm and all the visual originality metrics are in red:

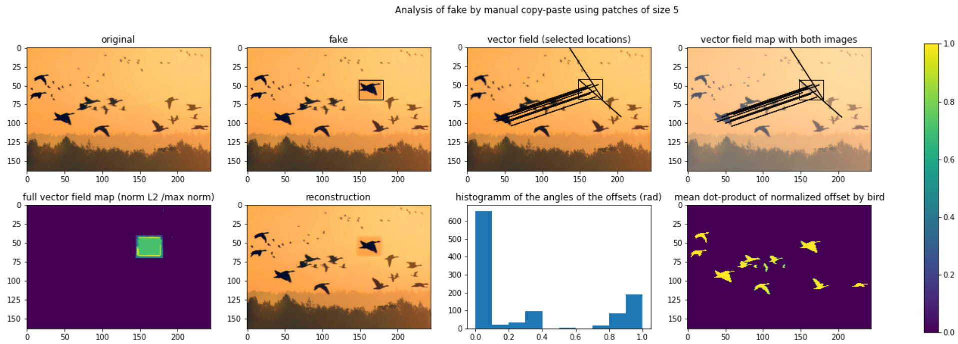


Figure 4: Result of the analysis of the copied and pasted image.

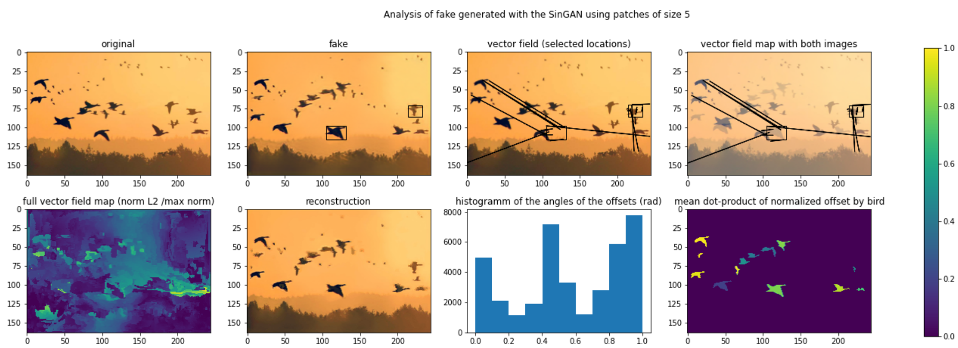


Figure 5: Result of the analysis of the image created by SinGAN.

- **AngleHistogramm** The histogram is dominated by a Dirak: all non-zero vectors come from a single direction.
- **MeanDotProduct** Since the vectors are all parallel, the mean of the scalar product is 1 (yellow) on the copied bird and the other birds are also yellow by default when all offsets are null (there is no angle associated with a null vector).

The copy and paste technique is not a technique that creates originality.

Analysis of SinGAN's creations Results are shown in Fig. 5. From this we can conclude :

- **AngleHistogramm** The histogram is not uniform but contains several dominant directions.
- **MeanDotProduct** We can see on the two selected birds that all the vectors do not go in the same direction even if it is true that some birds are clearly recopied. We see on the last figure that the originality analysis must be done bird by bird and that if

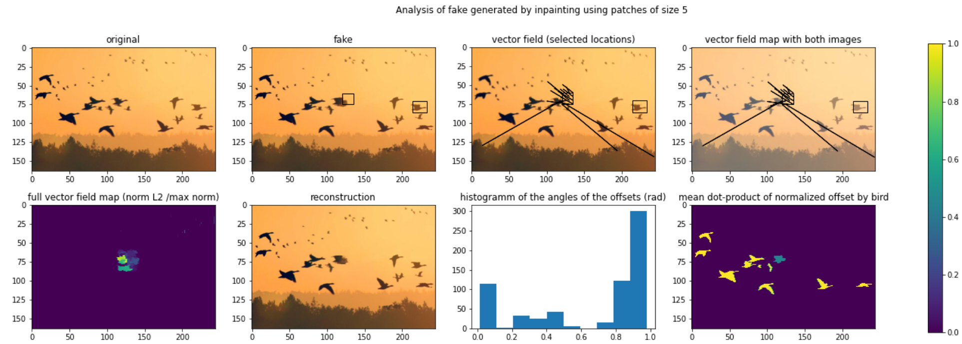


Figure 6: Result of the inpainted image analysis.

Technique	PatchSize	L2Norm	L1Norm	NbSet
SinGan	5	15.85	12.80	9239
Inpainting	5	9.23	8.03	209
CopyPaste	5	11.48	9.28	245

Table 2: Global metrics.

some birds are true copies, others are real patchworks coming from different parts of the image.

Analysis of the result of IPOL inpainting In order to produce the fake image analyzed opposite, we simply placed a square mask on a bird, then the IPOL inpaiting algorithm tried to reconstruct what hides the mask. Results are shown in Fig. 6. From this we can conclude :

- **AngleHistogramm** We can see that in the same way as the copied and pasted image, the histogram of the inpainted image seems to present a single dominant direction although this dominant direction seems to cohabit with other minor directions.
- **MeanDotProduct** We can see on the vector field selection that the vectors are not all parallel. According to this observation, the meandotproduct is lower than for the pasted copy but still high.

We can see that on the image of the standard, the reconstruction of the bird is not made of a single piece and the inpainted bird is correctly identified by the mean dot product and it has a rather good originality of 0.3.

3.1.4 Global analysis of originality

One can study the table of global metrics in Tab. 2.

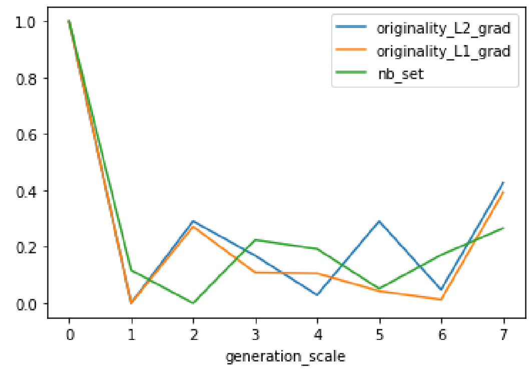


Figure 7: Result of the analysis for different scales (normalized between 0 and 1).

We can see that NbSet is considerably higher for SinGAN than for the other techniques and also the norms are of a different order of magnitude, which is not surprising but still allows us to have confidence in our different metrics.

But we must keep in mind that our global metrics are extensive of the modified area: we must differentiate the originality of the algorithm from the originality of the resulting image.

If we still want to quantify the SInGAN algorithms and networks of the IPOL algorithm we can still keep in mind the different *local* originality for different birds and remember that the meandot produced by the bird of the impainted image which was worth 0.3 was an intermediate score among the different bird scores generated by the sinGAN. This suggests that sinGAN and not local patch inpainting are roughly equivalent at least at the local level, the difference being that sinGAN is able to inpaint an entire image while iPOL inpainting can only inpaint a part of the image.

3.1.5 Analysis of the originality of SinGAN for different Generation Scales

At the end of the SinGAN training, images generated from different heights are available in the network. We can then study the originality of the network for the generations from different heights. We can then look at the metrics for different generation scales.

We can see that, according to intuition, the main difference is between the generation of level 0 and level 1 (see Fig. 7). And the other levels are all equivalent. It would thus seem that the "magic" of the originality of SinGAN lies largely in the very first layer and not in the additional layers which only serve to make the aspect realistic.

We can see that the birds gradually become more and more yellow as the scale increases but with some stagnation beginning at the scale 2 (see Fig. 8) which shows that the vectors are more and more parallel and more and more often null. Thus we can also see here that the "magic" of the originality of SinGAN lies largely in the very first layer.

There is a trade off between the originality and the visual plausibility of the output. We are not sure here but the following track is to be empirically validated in a complementary study: We think that the difference in patch size between the first layer and the next one

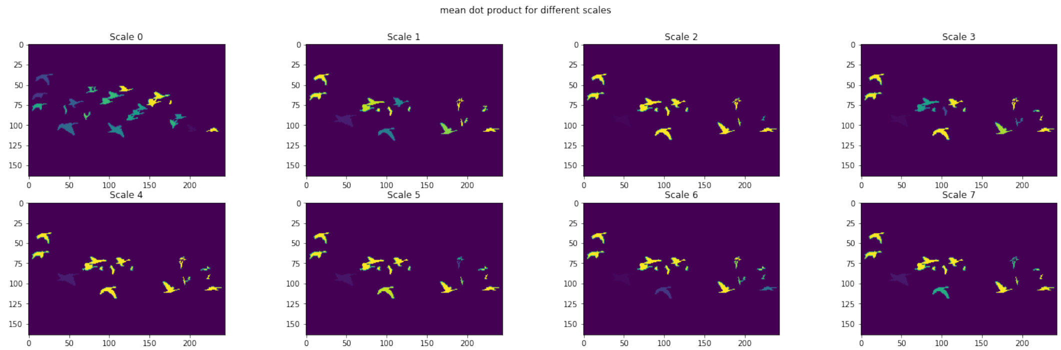
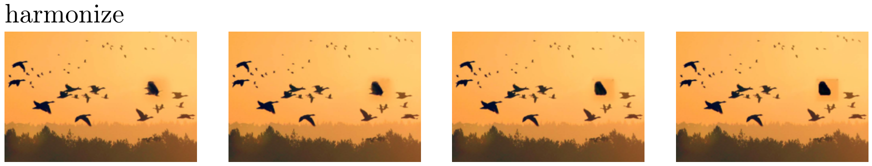


Figure 8: Mean dot product results for different scales.

(a) Drawing to harmonize (b) GSS = 1 (c) GSS = 2 (d) GSS = 3



(e) GSS = 4 (f) GSS = 5 (g) GSS = 6 (h) GSS = 7

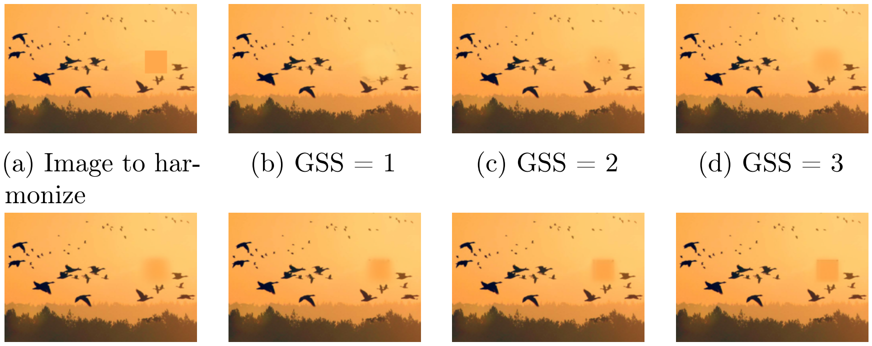
Figure 9: The original drawing and the results after using the Harmonization function of SinGAN for different scales. GSS: Generator starting scale.

is too big. We think that increasing the resolution more gradually between network 0 and network 1 could improve originality by better distributing the originality creation.

3.2 The SinGAN harmonization function

On the original image of the birds, we drew on an orange square in a very rough way the outline of a bird in black. We tested the harmonization of SinGAN on this example to make the drawing more realistic. The results are shown in Fig. 9. Results are best for a starting scale of 1 for the generator (the lowest available), since we need not only to change details here but completely change the whole square.

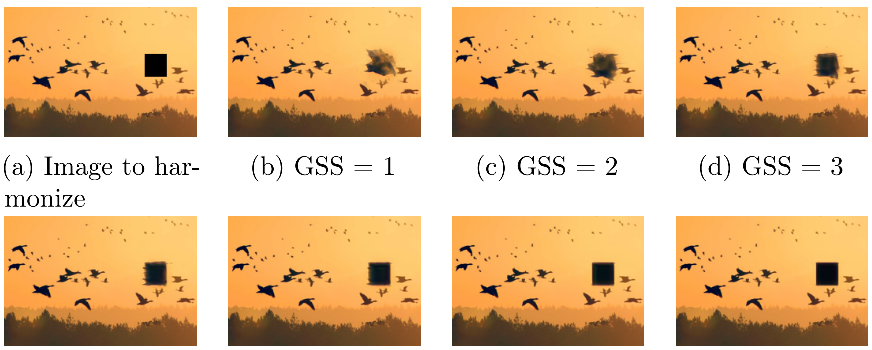
We tried to solve an image inpainting task using the Harmonization function of SinGAN. To do this, we drew a square of different colors in the original image, and the goal for the function would then be to modify this square to obtain a natural image. The results are on Fig. 10 for the orange square, on Fig. 11 for the black square and on Fig. 12 for the white square. Because orange is a color already present in the original image, this is much easier for the function to produce a correct result at a low starting scale. For the black square, the function tries to shape it as a bird, even if the result is not visually satisfying since the



(a) Image to harmonize (b) GSS = 1 (c) GSS = 2 (d) GSS = 3

(e) GSS = 4 (f) GSS = 5 (g) GSS = 6 (h) GSS = 7

Figure 10: Results of image inpainting using the Harmonization function of SinGAN for different scales. GSS: Generator starting scale.



(a) Image to harmonize (b) GSS = 1 (c) GSS = 2 (d) GSS = 3

(e) GSS = 4 (f) GSS = 5 (g) GSS = 6 (h) GSS = 7

Figure 11: Results of image inpainting using the Harmonization function of SinGAN for different scales. GSS: Generator starting scale.

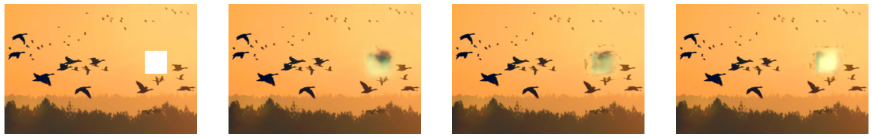
square is too big. For the white square, since white is absent from the original image, the function faces difficulties to produce a good image.

To finally test the Harmonisation function, we tested it on a denoising task, with gaussian and salt and pepper noises. Results are presented on Fig. 13 for the gaussian noise and on Fig. 14 for the salt and pepper noise. The behaviors are globally the same for both types of noise. Light pixels are smoothed to orange, while black pixels from noise tend to turn into small birds at high-scales.

3.3 Other functionalities of SinGAN

We tested the Super-Resolution method of SinGAN to increase the size of an image by a factor 4. Results are presented on Fig. 15. We see that the output is much clearer, while staying realistic, than the low resolution image that we naively upsampled.

We also made some animations that can be seen here.



(a) Image to harmonize (b) GSS = 1 (c) GSS = 2 (d) GSS = 3

(e) GSS = 4 (f) GSS = 5 (g) GSS = 6 (h) GSS = 7

Figure 12: Results of image inpainting using the Harmonization function of SinGAN for different scales. GSS: Generator starting scale.

(a) Image to harmonize (b) GSS = 1 (c) GSS = 2 (d) GSS = 3

(e) GSS = 4 (f) GSS = 5 (g) GSS = 6 (h) GSS = 7

Figure 13: Results of image denoising for a gaussian noise with $\sigma = 30$ using the Harmonization function of SinGAN for different scales. GSS: Generator starting scale.

(a) Image to harmonize (b) GSS = 1 (c) GSS = 2 (d) GSS = 3

(e) GSS = 4 (f) GSS = 5 (g) GSS = 6 (h) GSS = 7

Figure 14: Results of image denoising for a salt and pepper noise using the Harmonization function of SinGAN for different scales. GSS: Generator starting scale.

(a) Low resolution image

(b) Low resolution image upsampled naively

(c) High resolution image outputted by SinGAN

Figure 15: Results of the Super-Resolution method of SinGAN.

And all our results of manipulation of the SinGAN can be seen here.

4. Conclusion

In this study, we tested the SinGAN algorithm on image generation, harmonization, inpainting, denoising, super-resolution and animation tasks.

To analyze more quantitatively the creativity of SinGAN, i.e. its ability to produce images whose patches are not simple copies of the training image, we defined metrics based on the result of the PatchMatch algorithm, returning the vectors linking the similar patches between the original image and the generated image.

These metrics proved to be stable in practice, especially NbSet which was monotonous all the time, and they reflected what we found in practice, i.e. that creativity is much higher when generating at a low scale than at a high scale. Therefore, we have an experimental protocol using the PatchMatch algorithm to measure whether SinGAN makes a lot of copies or not.

The originality of SinGAN lies in the first layer, and the following layers only change details. In opening, we think that the change in resolution may be a bit too strong between the first and second layer, and that we would gain by being more progressive at the beginning.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan, 2017.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3), July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531330. URL https://doi.org/10.1145/1531326.1531330.

[3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[4] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. URL http://arxiv.org/abs/1704.00028.

[5] A. Newson, A. Almansa, Y. Gousseau, and P. Pérez. Non-Local Patch-Based Image Inpainting. *Image Processing On Line*, 7:373–385, 2017. https://doi.org/10.5201/ipol.2017.189.

[6] T. R. Shaham, T. Dekel, and T. Michaeli. Singan: Learning a generative model from a single natural image. *CoRR*, abs/1905.01164, 2019. URL http://arxiv.org/abs/1905.01164.

[7] M. Strathern. 'improving ratings': audit in the british university system. *European Review*, 5(3):305–321, July 1997. doi: 10.1002/(sici)1234-981x(199707)5:3<305::aid-euro184>3.0.co;2-4. URL https://doi.org/10.1002/(sici)1234-981x(199707)5:3<305::aid-euro184>3.0.co;2-4.