Topic K - Image manipulation with generative adversarial networks (GANs) - sinGAN for image denoising

January 19, 2021

Amrou CHOUCHENE
Telecom-Paris
amrou.chouchene@telecom-paris.fr

Wassim BOUATAY
Telecom-Paris
wassim.bouatay@telecom-paris.fr

Abstract

The goal of this project is to understand how GANs and sinGAN work. Then, we will extend the use of sinGAN to image denoising. Finally, we will make a qualitative and quantitative analysis on the performance of sinGAN on the data set '$set14$' for different types of noise.

1. sinGAN

1.1. Introduction

SinGAN was proposed by Tamar Rott Shaham et al. in 2019 [1]. The major difference between sinGAN and other GAN-based methods is that the first one doesn't need a dataset to learn the distribution within it, but instead, sinGAN learns the internal distribution from a single image using a pyramid of fully convolutional GANs to capture the distribution of all patches in the image at different scales, and then produce realistic image samples that consist of complex textures and non-repetitive global structures.

1.2. GAN

Let's first, introduce the key element of SinGAN which is the GAN [2] (Generative Adversarial Network). Its architecture is composed of two networks jointly trained.

• The Generator G: Its role is to learn the distribution of the data in the domain of interest. Then, the generator takes as input a random noise. The goal of the generator is to fool the discriminator with his fake generated samples.

• The Discriminator D: It is a classifier. Its role is to determine if an input is a real sample from the domain of interest or a fake sample given by the generator.

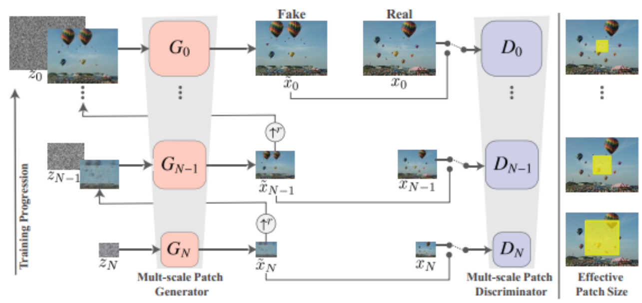

Figure 1. SinGAN's multi-scale Architecture

1.3. Architecture

SinGAN consists of a pyramid of GANs $\{(G_0,\ D_0)$ ... $(G_N,\ D_N)\}$, as we can see in figure 1. At the lowest part, the input of G_N is pure noise. Then, the input at each upper level G_i is the up-sampled output of G_{i+1} and random noise. The input of the discriminator D_i is the image on which we are training SinGAN down-sampled by a factor r^i and the generated sample by G_i.

1.4. Training

The training of SinGAN is done in a sequential approach. We start from the the coarsest scale. Once we finish training a GAN at level i, we move on to train the level $i-1$. At each level i the optimization problem is:

$$min_{G_i} max_{D_i}\ L_{adv}(G_i, D_i) + \alpha L_{rec}(G_i)$$

where L_{adv} is the adversarial loss and L_{rec} is the reconstruction loss. L_{adv} penalizes the distance between the generated patch and the true patch. L_{rec} ensure that there exists input noise maps from which we can recover the input image.

1.5. Results

In [1] the author presented several results of his work. The provided code [3] in GitHub covers different uses of sinGAN. The first thing that we have done was to check these results.

The results of our experiments can be found in the appendix. Figure 3 presents the main problem which is generating random images from a single input image. Although the algorithm generates high-quality results it is worth noting that it fails in some cases. In figure 4, we have the results for the editing problem. Figure 5 shows the results of the paint-to-image application at different scales.

2. GAN methods for image denoising

2.1. Related works - GCBD

The authors in [4] proposed a way to use GAN for image denoising. The proposed algorithm GCBD consists of using GAN to generate noise blocks then construct a paired dataset of noisy blocks and clean images. The dataset is fed to a Convolutional Neural Network for denoising the input. An overview of GCBD's architecture is given in figure 2.

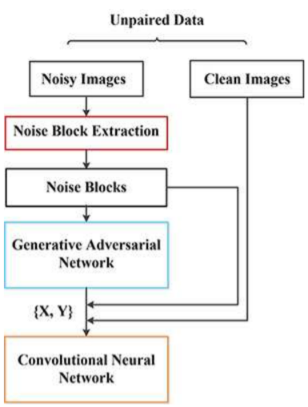

Figure 2. GCBD Architecture

2.2. sinGAN for Image denoising

SinGAN works on single images, unlike GCBD which uses a whole dataset to learn how to extract noise and how to eliminate it.

In our context, we cannot use clean images since we don't have access to it. But of course, simply training sinGAN on noisy images results in very bad distorted images as we can see in figure 6 in the appendix.

Our main idea is to use filtered images as prior and try to improve it using sinGAN. For most approaches, we used a median filter with a window of size 5x5.

We will use two types of noise: Gaussian noise with $\sigma = 30$ or 50 and Salt and Pepper noise with a ratio 10%.

2.3. Super-Resolution on filtered images

It is known that using filters in image denoising results in blurry images and a loss of details. So, our first idea was to filter the image on a small scale of the noisy image because, on a smaller scale, the blur is barely perceptible. Then, we use sinGAN's Super-Resolution (SR) to have a denoised image with better resolution and details.

This method failed to give decent results. Denoised images had a lot of artifacts. And even though, the PSNR improved compared to the noisy image it is still much worse than the PSNR of the denoised image using median filter. The results can be found in figure 7 in the appendix and table 1.

image	Noisy $\sigma = 50$	Median Filter	sinGAN with SR
Baboon	17.09	**18.57**	17.25
Lenna	18.58	**23.36**	19.72

Table 1. PSNR results using sinGAN Super-Resolution on filtered images

2.4. Training on noisy images, and injecting filtered images

Our second idea is similar to the paint-to-image described in [2]. We train the sinGAN on the noisy image and inject the filtered image on a certain scale. We will call this method **SinGAN Noisy-Filtered**.

Our intuition was that we want sinGAN to follow the filtered image which will give the general aspect of the image and avoid the distortions that we got in figure 6.

This method gave better results than the previous one, however, it is still less performing in term of PSNR compared to the median filter which was injected. Nonetheless, qualitatively we have less blurry images. The results can be found in figure 8 and table 2.

image	Noisy $\sigma = 30$	Median Filter	SinGAN N-F
Barbara	19.85	**20.86**	19.95
Lenna	22.27	**24.94**	24.13

Table 2. PSNR results using sinGAN Noisy-Filtered approach

The best results are given by injecting the filtered image to the scale N-1 or N-2

2.5. Training on filtered images, and injecting noisy images

2.5.1 Median Filter

Our last idea was to do the opposite of SinGAN Noisy-Filtered. Here, we train the model on the filtered image and we inject the noisy image at scale N-1 or N-2. We will call this method **SinGAN-Median**.

Since we lose a lot of details by applying the median filter, we want to use sinGAN to improve this result. By injecting the noisy image, we are asking the network to keep the details of the noisy image.

Image	Noisy Image	Median Filter	sinGAN-Median	NLmeans	SinGAN-NL	BM3D	FFDnet
Baboon	19.35	19.03	19.36	19.21	19.54	19.60	**19.64**
Barbara	19.85	20.86	21.06	21.14	21.10	20.42	**22.01**
Bridge	20.94	20.89	21.68	21.67	**22.35**	21.84	21.43
Coastguard	18.86	23.47	25.24	25.23	25.28	**26.32**	24.86
Comic	18.97	20.90	22.66	23.50	23.31	**24.27**	21.46
Face	19.54	27.06	25.31	26.02	27.33	27.84	**29.87**
Flowers	21.58	21.11	21.97	**22.29**	22.21	21.99	**22.29**
Foreman	19.15	26.22	26.52	26.05	26.72	25.90	**27.28**
Lenna	22.27	24.94	25.17	25.85	26.26	26.43	**26.50**
Man	21.46	22.02	22.06	23.28	**23.30**	23.03	22.54
Monarch	20.78	20.34	22.16	22.66	22.78	22.86	**22.98**
Pepper	22.03	24.40	24.63	24.80	**25.58**	23.08	25.25
Ppt3	19.05	16.64	18.85	**19.71**	19.36	18.37	19.00
Zebra	21.24	19.63	21.70	**22.82**	22.30	22.41	21.95
Average	20.36	21.95	22.74	23.15	**23.39**	23.17	23.36

Table 3. PSNR of different methods using gaussian noise $\sigma = 30$.

- Gaussian Noise $\sigma = 30$:

SinGAN-Median succeeded in improving the median filter. Quantitatively (table 3), the PSNR improves on average by 0.8 dB compared to the Median filter and by 2.4 dB compared to the noisy image. Qualitatively (figure 9), we have less blur and we managed to recover a lot of details as we wanted. We achieved close results to BM3D which is one of the best algorithms in image denoising.

- Salt and Pepper ratio=10%:

SinGAN-Median fails to improve the median filter for this type of noise. In fact, we know that the median filter is one of the best denoisers for Salt and Pepper even better than BM3D. The following table 4 and figure 10 shows the results of our experiments. In the example of Barbara, we can notice qualitatively the difference between sinGAN-Median and Median Filter.

image	Noisy	Median	SinGAN-Median	BM3D
Baboon	17.12	**19.31**	18.92	18.11
Barbara	17.13	**21.64**	20.94	18.53

Table 4. PSNR results using sinGAN-Median for Salt and Pepper noise compared to the median filter and BM3D

2.5.2 Non-Local means

In the last method, We managed to improve the results of the median filter using sinGAN. Now, we try the same approach on another denoising algorithm which is Non-Local means (NL-means). It is one of the best denoising algorithms that doesn't rely on deep learning, but, it results in homogeneous surfaces and a loss of details. We will call this method **SinGAN-NL** where we train SinGAN on the output of NL-means instead of the output of the median filter.

SinGAN-NL succeeded in improving the NL-means. Quantitatively (table 3), the PSNR improves on average by 0.25 dB compared to NL-means and by 3 dB compared to the noisy image. We also managed to achieve better results than BM3D. Qualitatively (figure 11), we have less homogeneous regions and more details.

2.5.3 Comparison with state of the art

As we can see in table 3 our SinGAN-NL gives better PSNR on average compared to BM3D and similar results as the state-of-art FFDnet.

We note that due to the long-running time of sinGAN, we resized images (to have a maximum of 250 pixels per dimension). We applied all the denoising algorithms at the smaller scale. Hence, the seemingly low PSNRs. But, since we applied the same resizing for all methods, our comparison remains valid.

3. Conclusion

sinGAN is a powerful tool for learning the distribution of an image without the need of a whole dataset.

We managed to use sinGAN to improve the results of the Median filter and the Non-Local Means but only for the gaussian noise and we believe that it can be useful for most denoising algorithms.

However, we conclude that sinGAN isn't appropriate for image denoising. It needs around 2 hours to train on a single image (on 12GB NVIDIA Tesla K80 GPU) which is too long compared to BM3D which takes few seconds on CPU and gives better or comparable results.

References

[1] Tamar Rott Shaham, Tali Dekel, Tomer Michaeli. *SinGAN: Learning a Generative Model from a Single Natural Image*. URL: https://arxiv.org/pdf/1905.01164.pdf.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. *Generative Adversarial Nets*. URL: https://arxiv.org/pdf/1406.2661.pdf.

[3] Tamar Rott Shaham. *SinGAN*. https://github.com/tamarott/SinGAN.

[4] Jingwen Chen, Jiawei Chen, Hongyang Chao, Ming Yang. *Image Blind Denoising With Generative Adversarial Network Based Noise Modeling*. URL: https://openaccess.thecvf.com/content_cvpr_2018/papers/Chen_Image_Blind_Denoising_CVPR_2018_paper.pdf.

Appendix

Figure 3. Generating random images (input in the first column)

Figure 4. Editing: on the left the input, on the right the output

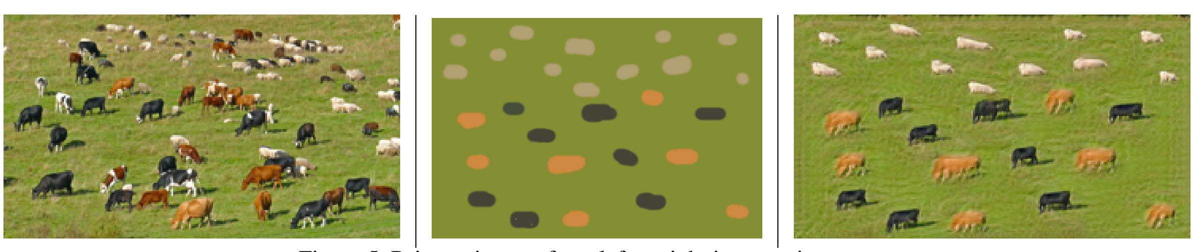

Figure 5. Paint-to-image: from left to right input, paint, output

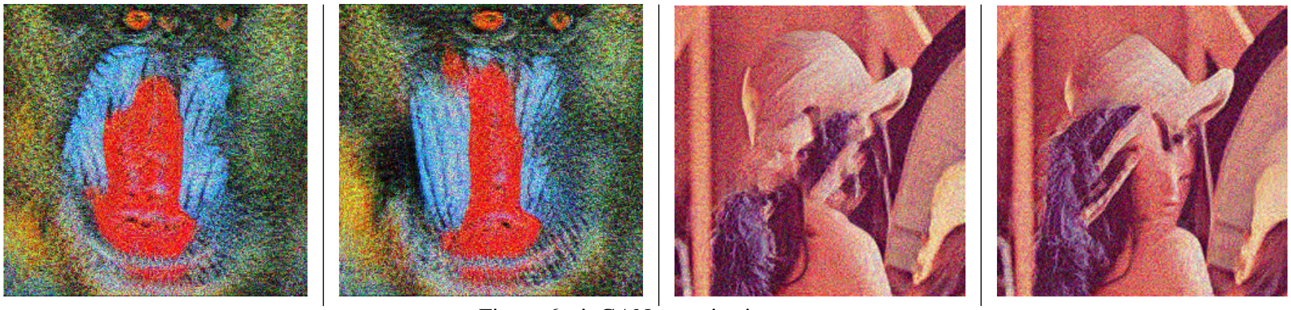

Figure 6. sinGAN on noisy images

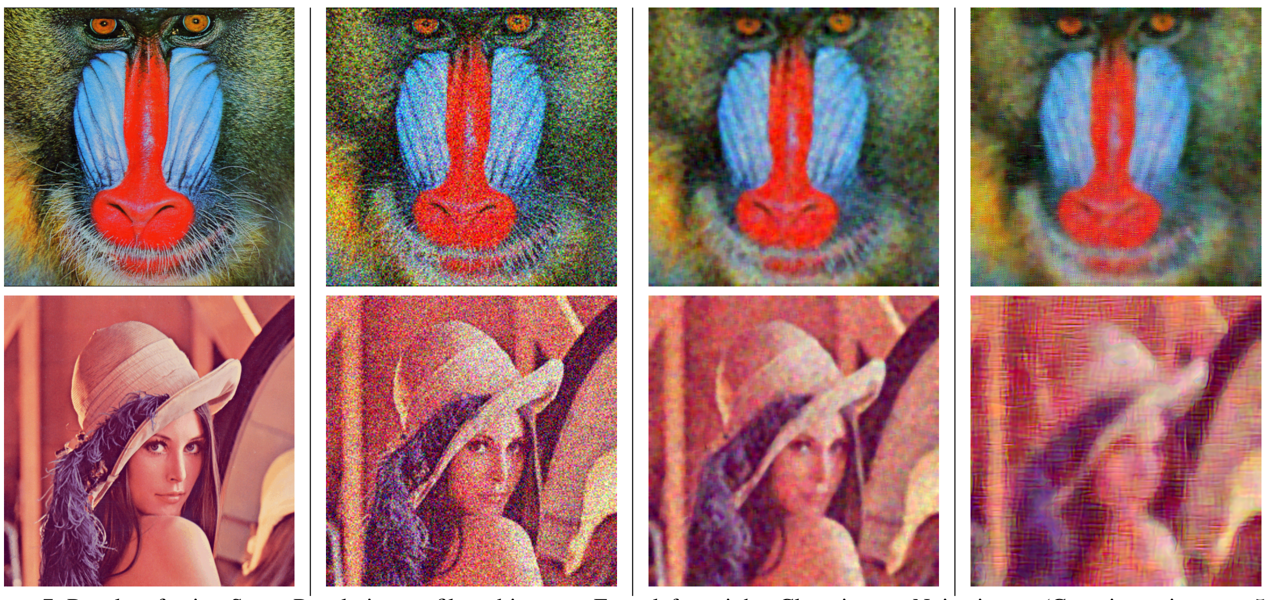

Figure 7. Results of using Super-Resolution on filtered images. From left to right: Clean image, Noisy image (Gaussian noise $\sigma = 50$), Denoised image with median filter and Result image after SR.

Figure 8. sinGAN Noisy-Filtered approach. From left to right: Clean image, Noisy image (Gaussian noise $\sigma = 30$), Denoised image with median filter and Result of sinGAN Noisy-Filtered approach

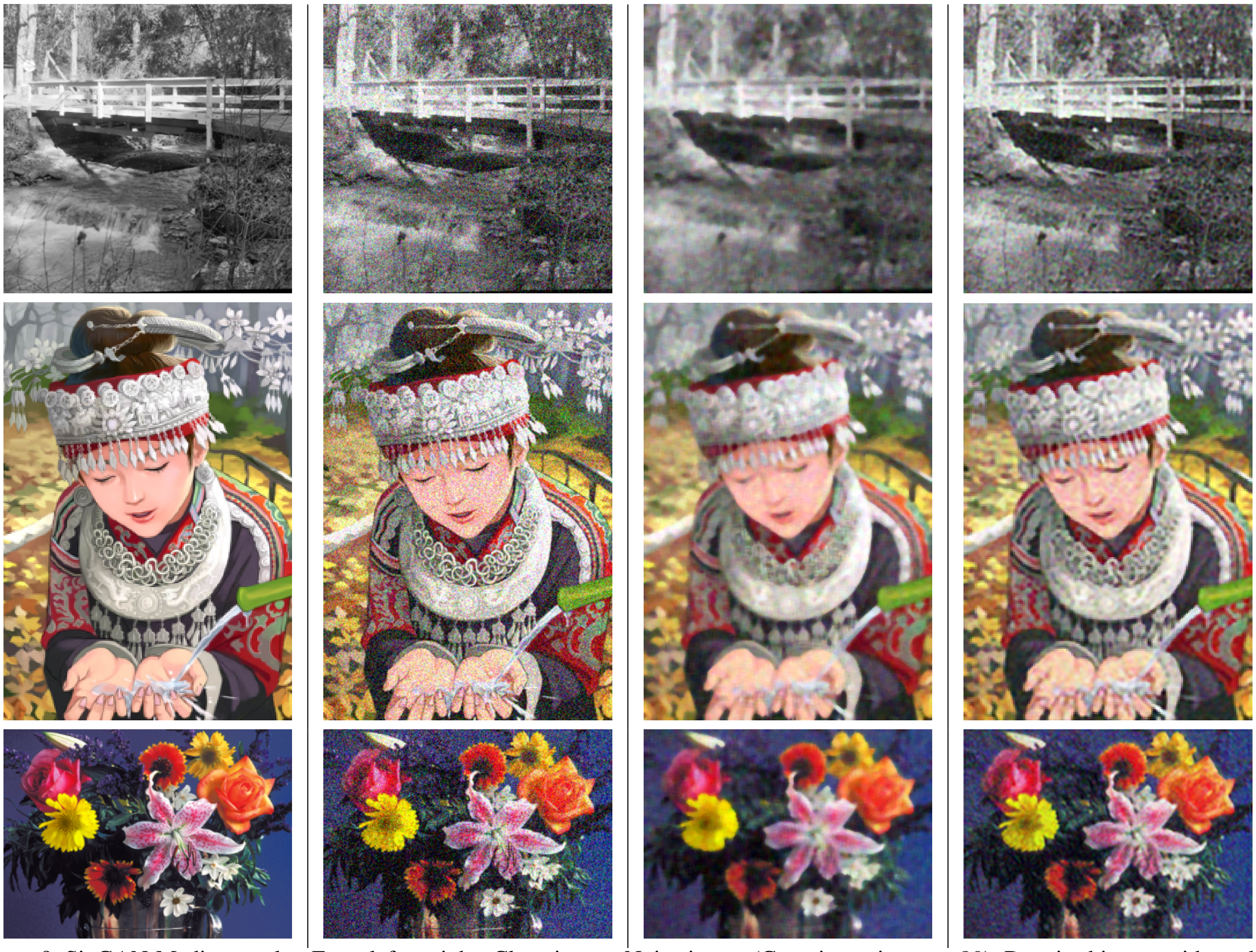

Figure 9. SinGAN-Median results. From left to right: Clean image, Noisy image (Gaussian noise $\sigma = 30$), Denoised image with median filter and Result sinGAN-Median

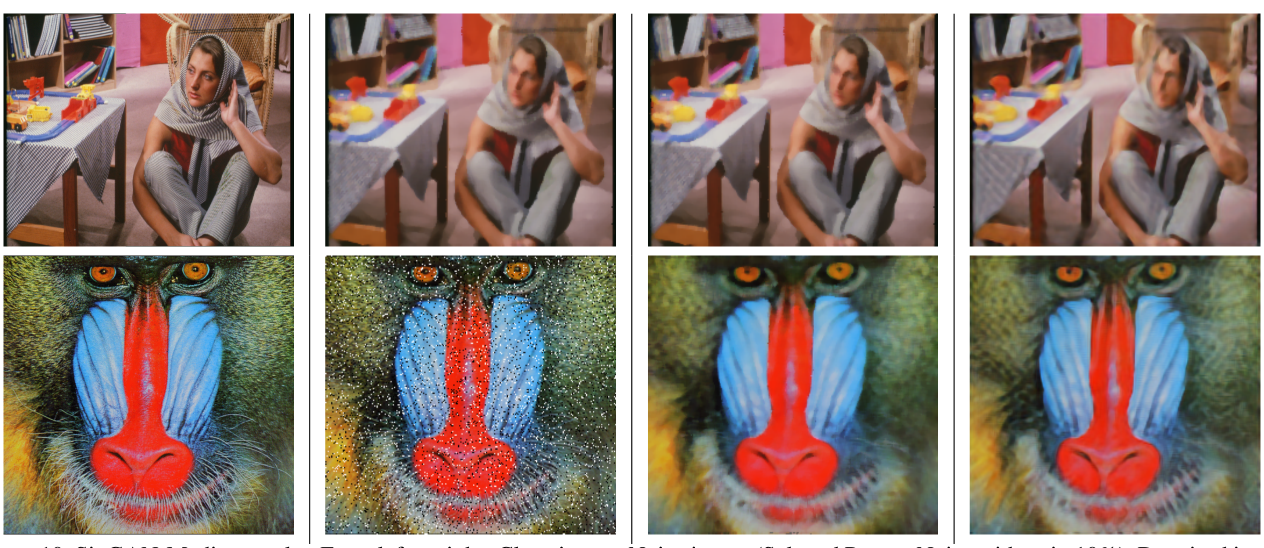

Figure 10. SinGAN-Median results. From left to right: Clean image, Noisy image (Salt and Pepper Noise with ratio 10%), Denoised image with median filter and Result sinGAN-Median

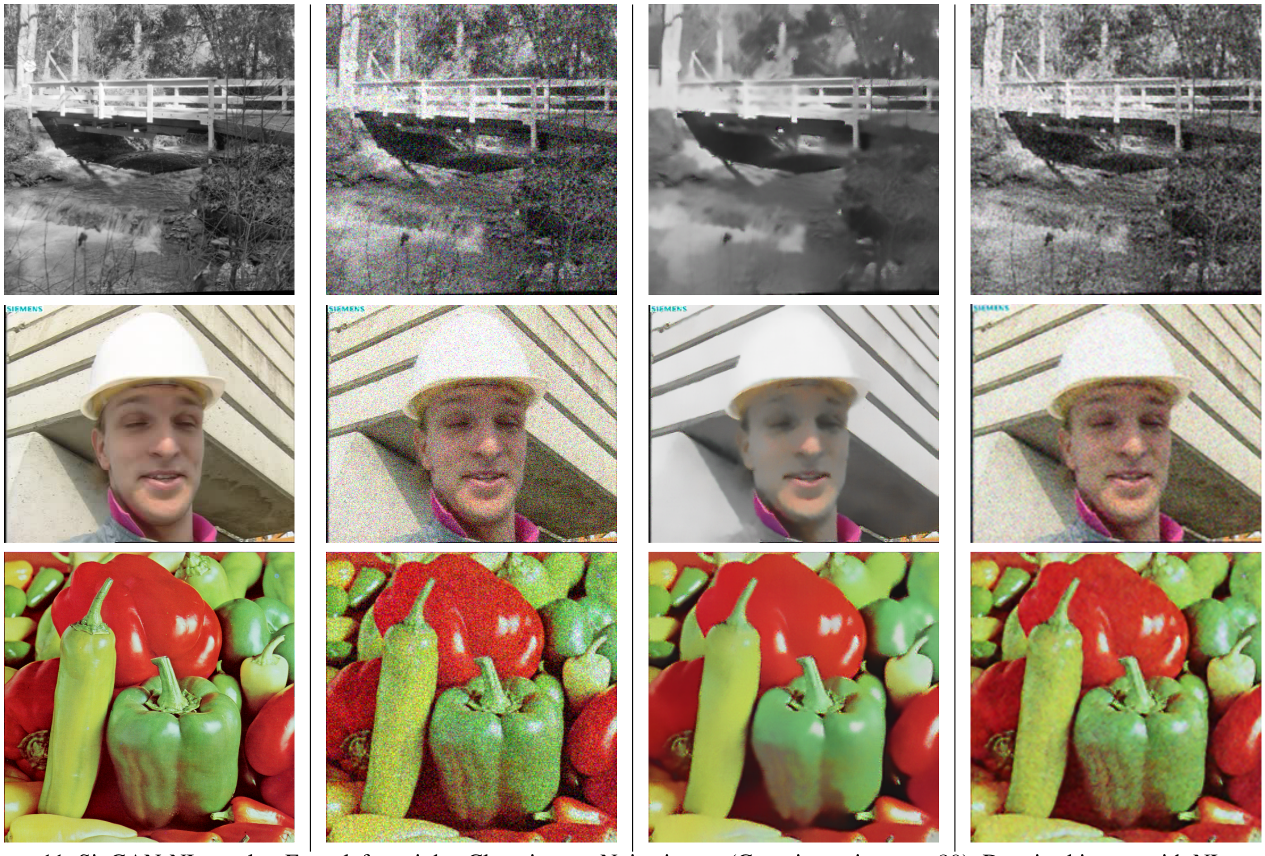

Figure 11. SinGAN-NL results. From left to right: Clean image, Noisy image (Gaussian noise $\sigma = 30$), Denoised image with NL means and Result sinGAN-NL