Unsupervised Downscaling of sea surface height with Deep Image Prior

Théo Archambault, Arthur Filoche, Anastase Alexandre Charantonis, Dominique Béréziat

UNSUPERVISED DOWNSCALING OF SEA SURFACE HEIGHT WITH DEEP IMAGE PRIOR

Théo Archambault[*1,2], **Arthur Filoche**[*1] , **Anastase Charantonis**[2], **Dominique Béréziat**[1]
Sorbonne Université, CNRS, LIP6, Paris [1]
Sorbonne Université, LOCEAN, Paris [2]
theo.archambault@lip6.fr

ABSTRACT

Oceanographic observations exist with different spatio-temporal resolutions and can be assimilated at various precision. The availability of numerous numerical simulations like ocean re-analysis make supervised machine learning appealing to deal with scale-related inverse problems. However data assimilation at finest resolutions using detailed oceanographic models is computationally intensive and building an exhaustive database may not be practical. Here we investigate the deep image prior method to downscale sea surface height observation and characterize estimation uncertainty in a fully-unsupervised manner. To do so, we set up a twin experiment using high resolution simulation from the NEMO Ocean engine and downscale degraded data with multiple ratios. Finally we give further perspectives of the method and make the link with data assimilation.

1 INTRODUCTION

Monitoring and modeling the Ocean is a constant scientific preoccupation whether for global climate understanding or numerical weather prediction. To do so, information from various sensors are combined with physics-based dynamical models in a data assimilation scheme. However, available data and known physics came with numerous spatio-temporal resolutions which leads to scale-matching problems as described in Bolton & Zanna (2019).

For instance, Sea Surface Height (SSH) can be retrieved at a resolution around 25 km from different altimeters which is coarse compared to other measured fields. SSH being a crucial information to derive oceanic currents, downscaling it is a topic of interest. From an image processing point of view, it can be seen as a Super-Resolution (SR) problem which is well documented for natural images.

Since the first super-resolution convolutional neural network Dong et al. (2014), literature aiming to solve such problems has been largely dominated by deep supervised learning approaches and still improves rapidly as shown in Wang et al. (2021). However, most of these methods leverage large dataset, which is not available in the context of very high resolution SSH recovery. An original unsupervised approach, called deep image prior introduced by Ulyanov et al. (2017). The method is proven to be successful in solving numerous imaging inverse problems such as denoising, image inpainting, or SR.

In this work, we investigate this unsupervised method to increase the resolution of a coarse SSH. We set a twin experiment using the NATL60 high resolution Ocean simulation proposed in Ajayi et al. (2019) with initial conditions from MERCATOR Lellouche et al. (2018) as ground truth data to evaluate the method. We first expose the used methodology and precise the case study. Finally we interpret results and discuss some perspectives.

*Equal contribution

2 METHODOLOGY

2.1 SUPER-RESOLUTION FRAMEWORK

The single image super-resolution task consists in recovering a particular high resolution image denoted $\boldsymbol{X}_{hr}$ from a low-resolution observation $\boldsymbol{X}_{lr}$, modeled as the output of a decimation operator denoted d, such that $\boldsymbol{X}_{lr} = d(\boldsymbol{X}_{hr})$. Estimating super-resolution from an observation can be seen as an optimization problem minimizing an energy function of the general form $\|d(\boldsymbol{X}_{sr}) - \boldsymbol{X}_{lr})\|$, where $\boldsymbol{X}_{sr}$ is the produced estimation. Naively choosing the least square cost function, we face severely ill-posed problems as d is usually non-injective. The solution can then be to handcraft a suited regularization or directly learn it leveraging a database. The deep image prior offers an unsupervised compromise, fitting a neural network to a single image using least square optimization.

2.2 DEEP IMAGE PRIOR

The idea behind deep image prior is that deep convolutional architectures are well-suited to generate image data and that imposing the solution of the variational problem to be generated by such network is a form regularization. From a practical standpoint, a generator network denoted $g_{\boldsymbol{\theta}}$ outputs the solution from a latent variable $\mathbf{z}$ such that $g_{\boldsymbol{\theta}}(\mathbf{z}) = \mathbf{X}_{sr}$. We choose here to impose that $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$ and describe the optimization problem in equation 1.

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|d \circ g_{\boldsymbol{\theta}}(\mathbf{z}) - \mathbf{X}_{lr}\|^2 \tag{1}$$

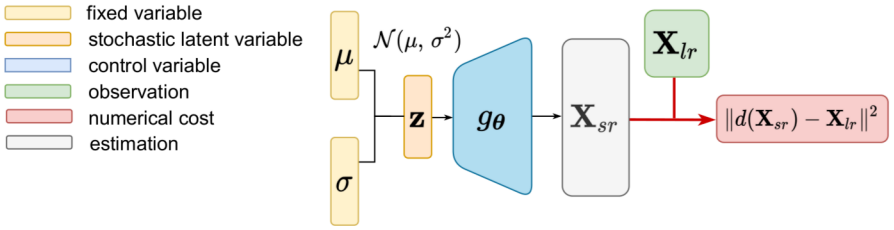

Figure 1: schematic view of the forward operation optimized in deep image prior

3 CASE STUDY

3.1 NATL60 HIGH-RESOLUTION DATA

We have at our disposal 4 months of high-resolution SSH daily images delivered by the Ocean physics-based model NATL60 Ajayi et al. (2019), based on NEMO 3.6 Madec et al. (2017) and using initial conditions from MERCATOR Lellouche et al. (2018). The highest resolution available, denoted R01, corresponds to a resolution of 1/60° at the Equator. The considered months are March, June, September and December 2008 and constitute 122 samples.

3.2 TWIN EXPERIMENT

The NATL60 simulation is here treated as ground truth at a resolution of 3/60° at the Equator (R03) and we purposely degrade it to evaluate the proposed super-resolution method on downscaling ratios, coherent with difference of scale differences between satellite observations of SSH and resolution of numerical simulation. To do so, the decimation operator d is assumed known, up to an additive white noise, and to be a 2-dimensional average pooling convolution. More precisely, by recursively using d, we generate the resolutions R03, R06, R12, R24, R48 and R96 corresponding to pixel size of 4.5, 9, 18, 36, 72 and $144\,\mathrm{km}$ respectively. An example of simulated observations is displayed in Figure 2. The goal is then to estimate a R03 image from each of these resolutions, corresponding

to downscaling ratios of $\times 1$, $\times 2$, $\times 4$, $\times 8$, $\times 16$ and $\times 32$, respectively. To quantify the quality of the estimated R03 resolution, we us the root mean square error (RMSE) and the structural similarity (SSIM).

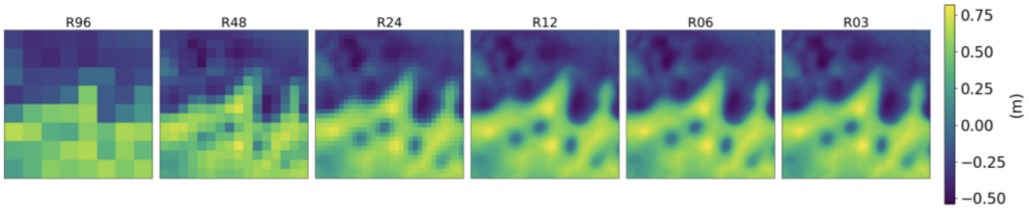

Figure 2: simulated observations of sea surface height at different resolutions

4 RESULTS

4.1 ESTIMATION EVOLUTION REGARDING DOWNSCALING RATIO

The main results of the twin experiment and a comparison with bicubic interpolation is plotted in Figure 3. The first interesting result to note is that even when given the exact R03 image without noise, deep image prior is not able to overfit precisely the high resolution image while bicubic interpolation obviously returns the exact result. This has to do with the chosen architecture, detailed in appendix. However, for similar task with noisy data, deep image prior is much more robust and also performs better for most difficult tasks. For instance if we zoom in the results at the particular task of downscaling $16\times$, going from R48 to R03, we see in Figure 4 that deep image prior systematically outperforms bicubic interpolation. In Figure 4 we follow a particular example along curves in the non-noisy setup. Associated error maps are displayed in appendix, see Figure 7.

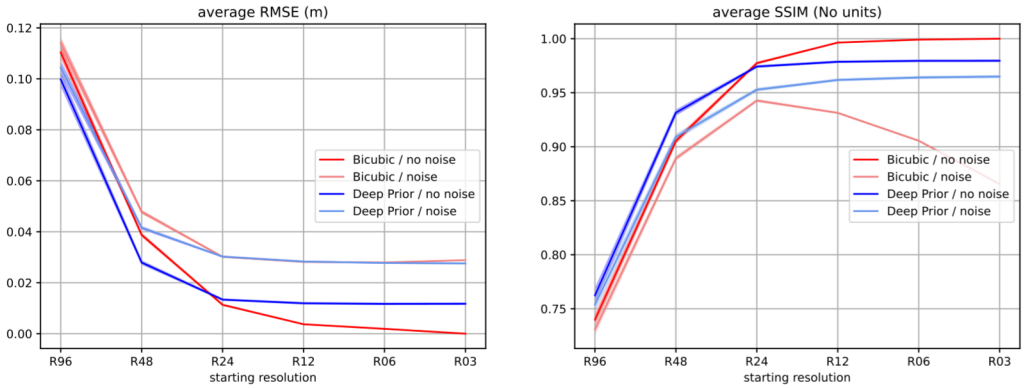

Figure 3: evolution of average scores regarding downscaling ratios $\times 32$, $\times 16$, $\times 8$, $\times 4$, $\times 2$ and $\times 1$

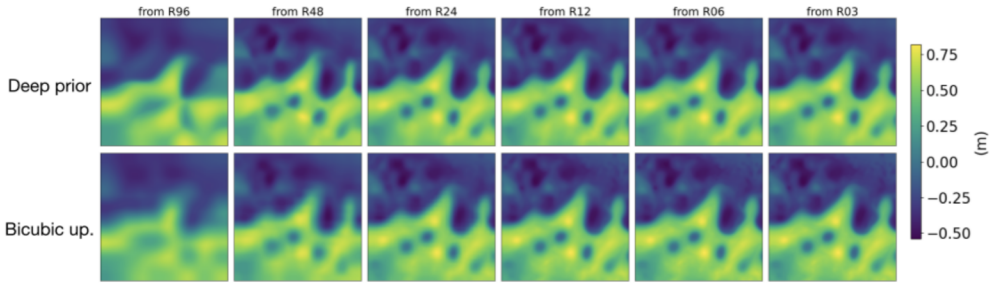

Figure 4: estimated high-resolution sea surface heights starting from different resolutions

4.2 Ensemble of deep image prior

We explicitly introduced stochasticity in the optimization scheme enforcing the latent space to behave like a multi-variate Gaussian variable. But another source of randomness inherent to deep learning architecture is at play when initializing weights of the neural network. It is now well documented that ensemble of neural network can produced significantly better results Allen-Zhu et al. (2020). Even though our approach is unsupervised we observed similar behavior. Training multiple networks with different weights initialization and averaging their outputs, we obtain better estimation than the best one performed by an individual network.

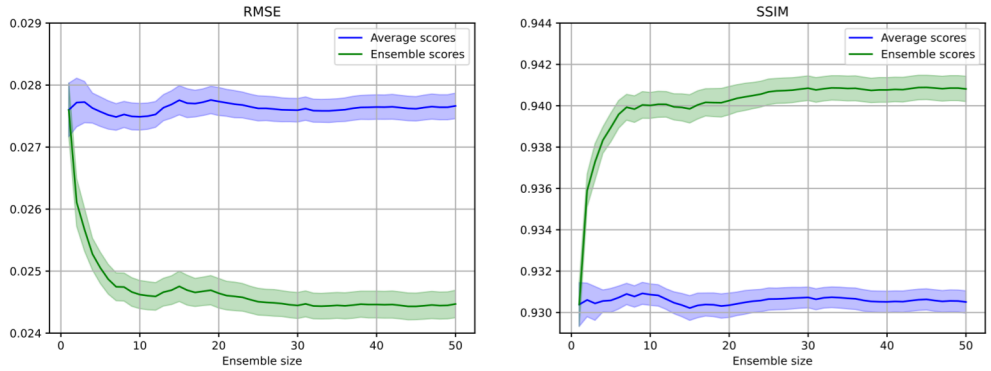

Figure 5: Evolution of deep prior ensembles scores with size, realized on 10 images, observing R48

In Figure 5 we focus on the $\times 16$ resolution augmentation task starting from R48 and consider a set of 10 different images. For each image, we optimize 50 deep prior networks and note that scores of averaged estimation are superior to the average score. We conclude that ensemble of deep image prior enhances performances. Similar results starting from finer resolution are available in appendix in Figure 9.

4.3 Uncertainty quantification

Being able to quantify uncertainty about state estimation may be critical for operational use of such methods. For instance in a data assimilation set up, information about uncertainty is used to assess confidence in data points accordingly. As we have some control on stochastic components of the methods, we can leverage them to quantify uncertainty in the estimations in two different way. Multiple image estimation can be produced either by sampling the latent space $\mathbf{z}$ or sampling members of an ensemble. It is then possible to estimate variance between samples in both case.

First, we notice that variance estimated by sampling the latent space is smaller than the one introduced by the ensemble, so the diversity in super-resolved images related to weights initialization is greater. But in both cases we observe strong correlation between errors and standard deviations to

that they can be used as uncertainty measure. It is to be noted that biggest errors seem to be made on sharp edges in the image, and observing in the main experiment that the chosen architecture has a smoothing effect, we may observe different uncertainties with different generative architectures.

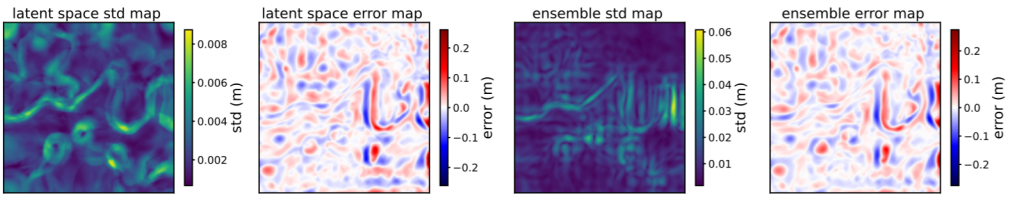

Figure 6: standard deviation and error maps for the latent space sample group (10 samples) and the ensemble group (10 members)

5 Perspectives

5.1 Knowledge distillation

From a purely practical-oriented standpoint, one may be interested in having a tool producing the performances of an ensemble of deep image prior estimation. However, training multiple deep networks for each available image is computationally intensive. It is possible to condense and speed up the process by learning over outputs of the ensembles in a supervised fashion. This process is named *knowledge distillation* and has been introduced in Allen-Zhu et al. (2020).

5.2 Multi-channel output - sea surface temperature

Satellite data usually provide high-resolution sea surface temperature (SST) and, comparatively, low-resolution SSH. Preliminary experiment in a supervised setting let us think that high-resolution SST can help improving resolution of SSH data. But we do not know if this information gain is only statistical leveraging a large database. If it's purely the case, high-resolution SST won't be informative in a single-image unsupervised setup. We performed a similar experiment still asking the generator to output a high-resolution SSH but simultaneously the high-resolution SST as second channel. The obtained results show that we did not succeed in extracting meaningful information form the SST. However, we cannot conclude on the feasibility because it is not clear that the designed architecture is suited for this task.

5.3 Data Assimilation

Deep image prior can solve various kinds of imaging inverse problem. Adding physics-based constraints it can be used in a data assimilation framework to reconstruct the full trajectory of a physical system. Approaches combining deep neural architectures and differentiable physical layers already exist Beucler et al. (2021); Mosser et al. (2018). We already developed in an other work a deep prior variational assimilation algorithm. Such constraints may force coherence in a multi-channel generated estimation. Moreover, by sampling the latent space it is possible to bridge ensemble and variational data assimilation methods.

6 Conclusion

We adapted the original unsupervised method deep image prior to a geophysical single-image super-resolution problem. More precisely we used neural networks to generate an estimation of high-resolution SSH. To demonstrate its interest, we designed a twin experiment mimicking the potential behavior of low-resolution satellite image data and tested the method for multiple downscaling ratios. We introduced two stochastic components allowing to enhance performances and to quantify uncertainty of the estimation. Finally, we drew perspectives towards practical use of the methods and further developments involving multiple-image output and physics-based constraints.

REFERENCES

A. Ajayi, J. Le Sommer, E. Chassignet, J.-M. Molines, X. Xu, A. Albert, and E. Cosme. Spatial and temporal variability of north atlantic eddy field at scale less than 100 km. *Earth and Space Science Open Archive*, pp. 28, 2019.

Z. Allen-Zhu, Y. Li, and Z. Allen-Zhu. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv preprint, December 2020.

T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Physical Review Letters*, 126(9):1079–7114, Mar 2021.

T. Bolton and L. Zanna. Applications of deep learning to ocean data inference and subgrid parameterization. *Journal of Advances in Modeling Earth Systems*, 11(1):376–399, 2019.

C. Dong, C.C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision (ECCV)*, pp. 184–199, 2014.

A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the carbon emissions of machine learning. arXiv preprint, 2019.

J.-M. Lellouche, E. Greiner, O. Le Galloudec, C. Regnier, M. Benkiran, C.-E. Testut, R. Bourdalle-Badie, M. Drevillon, G. Garric, and Y. Drillet. Mercator ocean global high-resolution monitoring and forecasting system. *New Frontiers in Operational Oceanography*, pp. 563–592, 2018.

G. Madec, R. Bourdallé-Badie, P.-A. Bouttier, C. Bricaud, D. Bruciaferri, D. Calvert, J. Chanut, E. Clementi, A. Coward, D. Delrosso, et al. Nemo ocean engine, 2017.

L. Mosser, O. Dubrule, and M. Blunt. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior. *Mathematical Geoscience*, pp. 53–79, 2018.

A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representation (ICLR)*, 2016.

D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Deep image prior. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Z. Wang, J. Chen, and S. Hoi. Deep learning for image super-resolution: A survey. *Transactions on Pattern Analysis and Machine Intelligence*, pp. 3365–3387, 2021.

A Appendix

A.1 Additional Figures

A.1.1 Estimation evolution regarding downscaling ratio

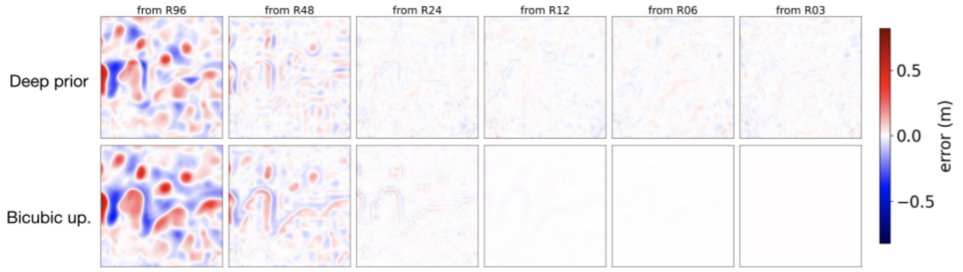

Figure 7: error of estimated high-resolution sea surface height from different resolutions

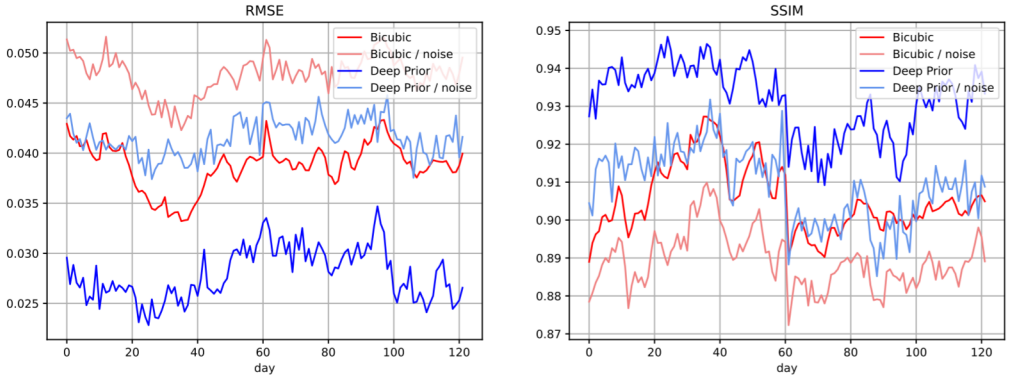

Figure 8: comparison between deep image prior and bicubic interpolation on a $\times 16$ super-resolution task, all 122 samples considered

A.1.2 Ensemble of deep image prior

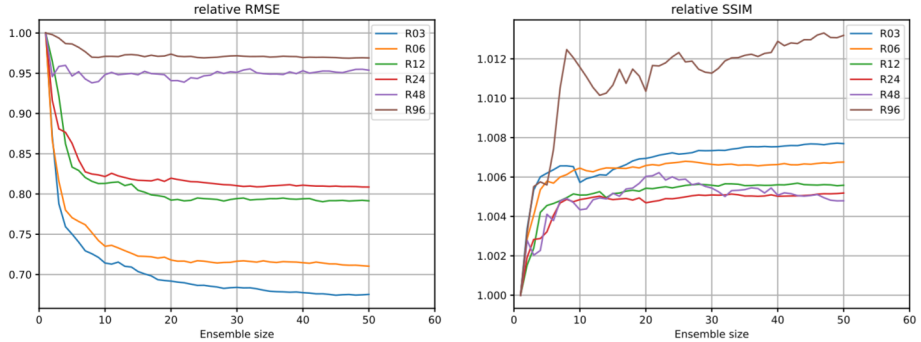

Figure 9: evolution scores of deep prior ensembles scores relatively to the first member, realized on 1 image, observing all resolutions

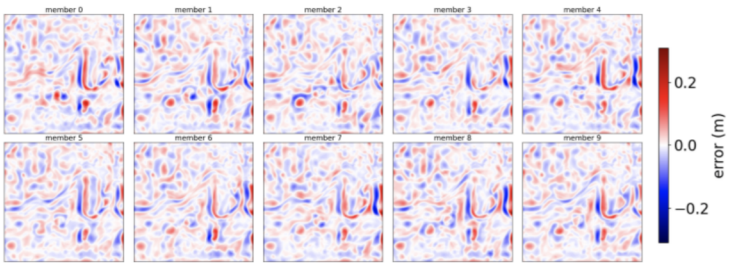

Figure 10: error of estimated high-resolution sea surface height for each member of an ensemble of deep prior, starting from resolution R48

A.2 Deep Image Prior settings

A.2.1 Optimization

The deep prior architecture was implemented in Pytorch and uses the gradient descent algorithm. We used the ADAM optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.999$) during 1500 epochs. The starting learning rate is set to 0.001 and we used a reduce learning rate on plateau scheduler (with $factor = 0.95$, $patience = 10$).

A.2.2 Generator Architecture

In this experiment, we use a neural architecture similar to the generative convolutional network presented in Radford et al. (2016) adpating it to the dimension of our problem, but replacing deconvolution operations to avoid checkerboard artifacts as described in Odena et al. (2016). The exact architecture is given below.

Layer (type)	Output Shape	Param #
ConvTranspose2d-1	[-1, 512, 4, 4]	819,200
BatchNorm2d-2	[-1, 512, 4, 4]	1,024
ReLU-3	[-1, 512, 4, 4]	0
Upsample-4	[-1, 512, 8, 8]	0
ReflectionPad2d-5	[-1, 512, 10, 10]	0
Conv2d-6	[-1, 256, 8, 8]	1,179,904
BatchNorm2d-7	[-1, 256, 8, 8]	512
ReLU-8	[-1, 256, 8, 8]	0
Upsample-9	[-1, 256, 16, 16]	0
ReflectionPad2d-10	[-1, 256, 18, 18]	0
Conv2d-11	[-1, 128, 16, 16]	295,040
BatchNorm2d-12	[-1, 128, 16, 16]	256
ReLU-13	[-1, 128, 16, 16]	0
Upsample-14	[-1, 128, 32, 32]	0
ReflectionPad2d-15	[-1, 128, 34, 34]	0
Conv2d-16	[-1, 64, 32, 32]	73,792
BatchNorm2d-17	[-1, 64, 32, 32]	128
ReLU-18	[-1, 64, 32, 32]	0
Upsample-19	[-1, 64, 64, 64]	0
ReflectionPad2d-20	[-1, 64, 66, 66]	0
Conv2d-21	[-1, 32, 64, 64]	18,464
BatchNorm2d-22	[-1, 32, 64, 64]	64
ReLU-23	[-1, 32, 64, 64]	0
Upsample-24	[-1, 32, 128, 128]	0
ReflectionPad2d-25	[-1, 32, 130, 130]	0
Conv2d-26	[-1, 16, 128, 128]	4,624

```
      BatchNorm2d-27         [-1, 16, 128, 128]              32
             ReLU-28         [-1, 16, 128, 128]               0
         Upsample-29         [-1, 16, 256, 256]               0
  ReflectionPad2d-30         [-1, 16, 258, 258]               0
           Conv2d-31          [-1, 1, 256, 256]             145
================================================================
Total params: 2,393,185
Trainable params: 2,393,185
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 44.59
Params size (MB): 9.13
Estimated Total Size (MB): 53.72
----------------------------------------------------------------
```

A.3 ACKNOWLEDGMENTS - CO2 EMISSION RELATED TO EXPERIMENTS

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.053 kgCO_2eq/kWh. A cumulative of 48 hours of computation was performed on hardware of type RTX A6000 (TDP of 300W).

Total emissions are estimated to be 0.76 kgCO_2eq of which 0 percents were directly offset.

Estimations were conducted using the MachineLearning Impact calculator presented in Lacoste et al. (2019).