Downscaling Oceanographic Satellite Data with Convolutional Neural Networks

Bertrand Delorme

1 Introduction

The ocean circulation is a central player in regulating Earth's climate and supporting marine life by transporting heat, carbon, oxygen, and nutrients throughout the world's ocean. Yet, our understanding of the processes governing these fluxes is still limited because of the relatively small spatial scales involved, which are difficult to observe with current remote sensing techniques. Yet, small-scale perturbations in the ocean have been shown to have a tremendous consequences in the global circulation (Ferrari & Rudnick, 2000; Thomas et al., 2008; McWilliams, 2016).

One technique commonly used to enrich the wealth of information contained in satellite data is downscaling. It consists in reconstructing a high-resolution (HR) observation from a low-resolution (LR) one's. For 2D fields, this problem is similar to image super-resolution in the computer vision and pattern recognition community (Yang et al., 2014), which has shown tremendous progress in recent years thanks to deep learning (Timofte et al., 2016). This approach uses convolutional neural networks (CNNs) instead of traditional interpolation techniques to learn the downscaling relationship between LR and HR image, and has emerged as the state of the art approach in recent years (Dong et al., 2015, 2016; Kim et al., 2015, 2016; Shi et al., 2016; Wang et al., 2016; Ledig et al., 2017).

Given the availability of large datasets for ocean remote sensing, it appears very tempting to investigate the potential of deep learning models to downscale satellite-derived observations. One question who naturally arises is: are deep learning models efficient to downscale ocean remote sensing datasets? This question is non trivial as the spatio-temporal scales involved in oceanographic field might be difficult to reconstruct for CNN. A previous study (Ducournau & Fablet, 2017) has already worked on this question by rescaling Sea Surface Temperature (SST) field by a factor of 3. Using a very simple CNN architecture, they got very promising results.

In this work, we will try to extend this early study by rescaling both SST with a factor of 5 and Sea Surface Height (SSH) with a factor of 3, and by comparing different CNN models taken from the super-resolution literature.

2 Dataset

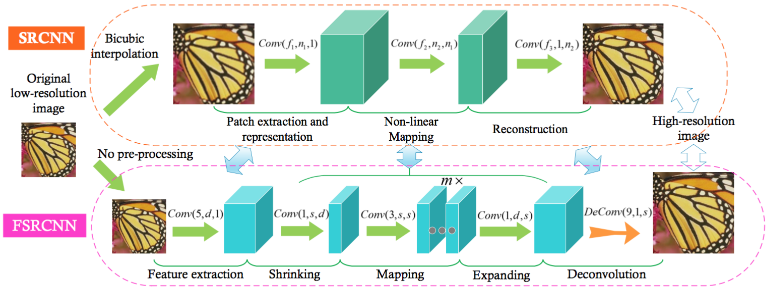
In this paper, 4 datasets are considered, where data are taken from year 2009 at a daily resolution in the highly-dynamic Pacific Equatorial region (crucial for the global climate through the wonderful phenomenon of El Nino):

- The OSTIA 1/20° SST (Donlon et al., 2012), assimilating satellite data from microwave and infrared sensors with in situ data from drifting and moored buoys into an ocean model. It will provide a representative ground-truth dataset to mimic the reconstruction of a HR SST field from a LR observation.
- The NOAA-OI 1/4° SST, combining observations from 2 satellites: AVHRR and AMSR-E (in-situ data also used to perform the interpolation). It will provide the pure satellite SST data distribution.
- The Mercator-GOAFS 1/12° SSH, consisting in altimeter data assimilated into the NEMO ocean model (Madec et al., 2008). It will provide a representative ground-truth dataset to mimic the reconstruction of a HR SSH field from a LR observation.
- The Mercator-GOMASG 1/4° SSH, merging data from all altimeter missions: Jason-3, Sentinel-3A, HY-2A, Saral/AltiKa, Cryosat-2, Jason-2, Jason-1, T/P, ENVISAT, GFO and ERS1/2 to investigate the performance of our models on pure satellite SSH data.

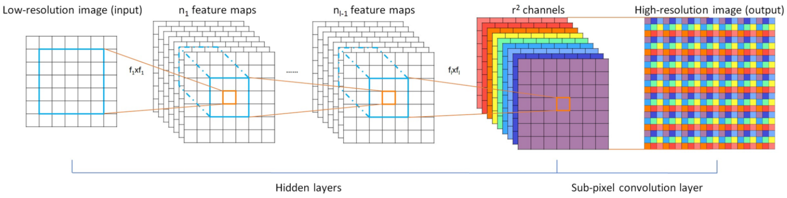
3 Models

In this study, 5 different super-resolution models are investigated: Super-Resolution Convolutional Neural Network (SRCNN; Dong et al., 2015), Fast Super-Resolution Convolutional Neural Networks (FSRCNN; Dong et al., 2016), Efficient Sub-Pixel Convolutional Neural Network (ESPCN; Shi et al., 2016), Very Deep Super-Resolution Convolutional Neural Network (VDSR; Kim et al., 2016), End-to-End Deep and Shallow networks (EEDS, Wang et al., 2016). We have implemented all the models mentioned in this section in Keras with TensorFlow backend[1]. In this section, we will briefly introduce those models and the parameters used, but we refer to the original papers, or our code, for more information.

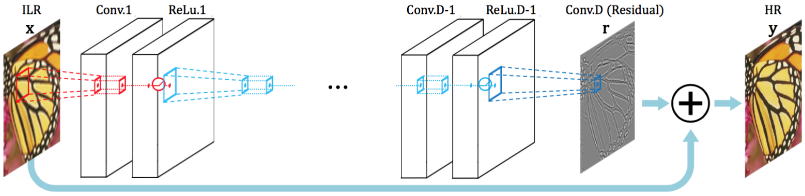
[1] code available at: $https://github.com/bdelorme/HRCNN$



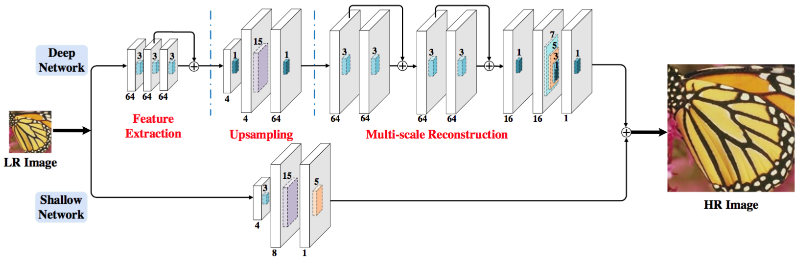
SRCNN: Given a LR image interpolated, the first layer extracts a set of feature maps. The second layer maps these features maps nonlinearly to HR patches representations. The last layer combines the predictions within a spatial neighborhood to produce the final HR image. **FSRCNN**: adopts the original low-resolution image as input without bicubic interpolation but introduced a deconvolution layer at the end of the network to perform upsampling. The non-linear mapping step in SRCNN is then replaced by three steps, namely the shrinking, mapping, and expanding step. Note that FSRCNN adopts smaller filter sizes and a deeper network structure. These improvements have been shown to provide FSRCNN with better performance but lower computational cost than SRCNN. Image taken from Dong et al., 2016. The parameters that we use in our study are $f_1 = 9$, $f_2 = 3$, $f_3 = 5$, $n_1 = 64$, $n_2 = 32$, $d = 56$, $s = 12$, $m = 4$.



ESPCN: two convolution layers for feature maps extraction and a sub-pixel convolution layer that aggregates the feature maps from LR to HR space in a single step. The depth of the network, as well as its computational and memory complexity is much lower than FSRCNN. Image taken from Shi et al., 2016. In this study, $f_1 = 5$, $f_2 = 3$, $f_3 = 3$, $n_1 = 64$, $n_2 = 32$, $n_3 = 1$.



VDSR: an interpolated LR image goes through a cascade of Conv-ReLu layers and transforms into a HR image. The network predicts a residual image and the addition of LR and the residual gives the desired output. Image taken from Kim et al., 2015. In this project, we use 64 filters for each Conv layers, and we have 20 pairs of Conv-ReLu layers.



EEDS: jointly train a deep network and a relative shallow network. The former learn the residual while the latter takes the responsibility of learning the major image components. For the deep part of the network, 3 steps are taking place: 1- feature extraction, 2- upsampling, 3- multi-scale reconstruction. Value of the parameters used in this study as on the picture. From Wang et al., 2016.

4 Training Strategy

During training, each network is fed with small LR image overlapping patches of size 33 × 33 (with a stride of 14 pixels, resulting in approximately 60 000 patches) extracted from the training data, along with their original HR counterpart. The model is optimized with the Adam method minimizing the Mean Squared Error between the HR output produced by the network and the original HR.

Most of the hyper-parameters not mentioned here have been chosen from the original papers which discuss further their implications – this discussion is beyond the scope of this paper but will be investigated in future work.

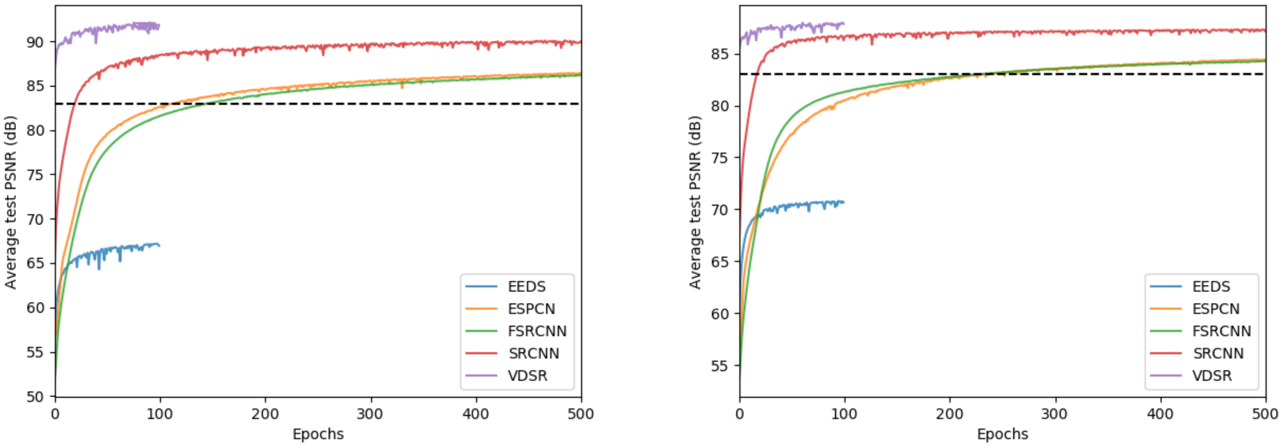
We trained the SRCNN, FSRCNN and ESPCN models for 500 epochs, but because of ressources and time limitation, the VDSR and EEDS were trained for only 100 epochs.

5 Results

5.1 PSNR

We evaluate the performance of a model in terms of the PSNR (Peak Signal to Noise Ratio) between the original HR image and the reconstructed one: $PSNR = 10n\log_{10}(\frac{1}{\sum_{i=1}^{n}|I(i)-\widetilde{I(i)}|^2})$, with I the original HR image and $\widetilde{I}$ the reconstructed one.

In figure 2, we show PSNR on test set as a function of epochs for all network architectures for SSH with upscaling factor 3 and SST with upscaling factor 5. From these figures, it can be seen that the VDSR is performing better, even if trained for only 100 epochs. This is probably due to the fact that it has much more parameters and thus is able to learn deeper features. Yet, the EEDS model has a lot of parameters also but performs poorly in the Test Set. Current work is undertaking to understand why and fix that. All the other models perform better than the traditional bicubic interpolation represented by the horizontal black dotted line in the figure, after ≈ 150 epochs.



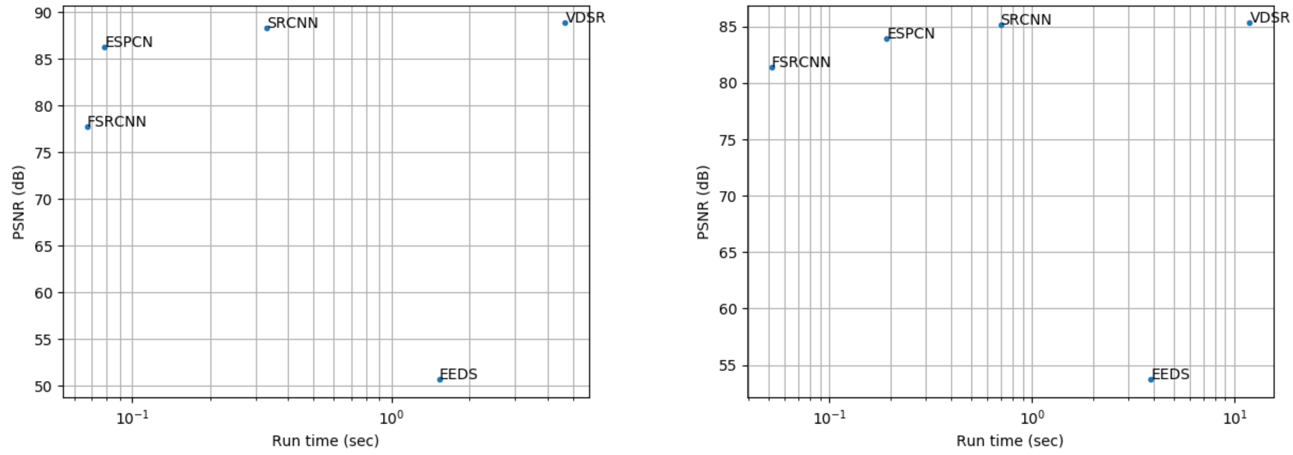
(a) SSH with upscaling factor 3

(b) SST with upscaling factor 5

Figure 1: Average test PSNR vs. epochs for different networks. The horizontal dotted black line denotes the bicubic interpolation score.

5.2 Trade-offs Between PSNR and Run Time

The performance of the networks can also be evaluated by comparing the PSNR of output over test images to the run time. Here run time is defined as the time it takes to make a HR prediction per LR image after the network has been trained. The results are shown in figure 2 for all the network architectures that we tested. Basically, for operational purposes, there is a trade-off between PSNR and run time to find. For example, the FSRCNN enhances considerably the run time of the standard SRCNN but decrease its average PSNR value at the same time. Also, SRCNN has a PSNR score a little bit lower than the VDSR but runs much more rapidly. The EEDS, on the other extreme, cumulate disadvantages as it runs slowly and gives poor performance.



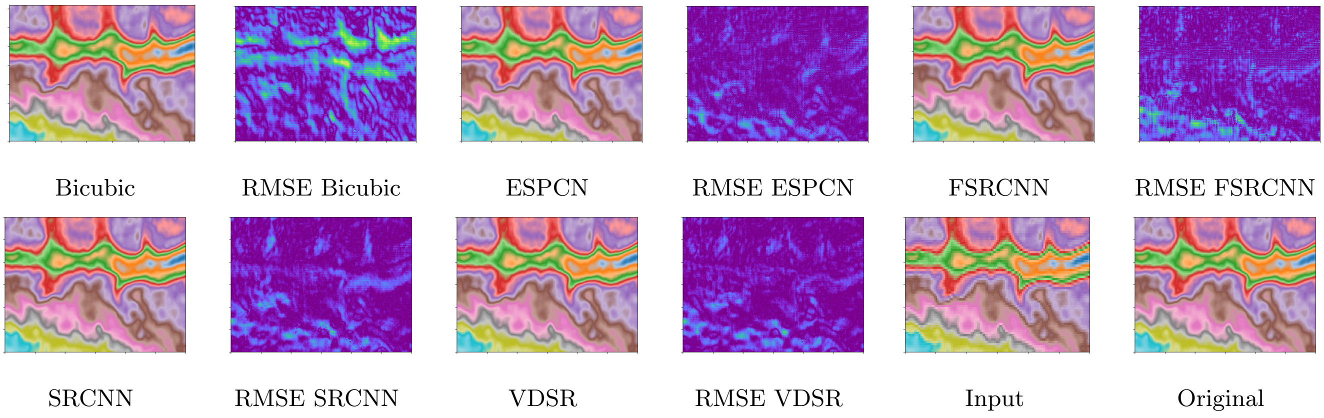
(a) SSH with upscaling factor 3

(b) SST with upscaling factor 5

Figure 2: Trade-off between accuracy (PSNR) and speed (run time) for the different networks considered here.

5.3 Image Examples

Here, we show the resulting visuals for the SST case (results similar for SSH) from each model along with their RMSE with respect to the original image. We don't show outputs from the EEDS model which has been proved to be inconsistent in the previous section. Even if differences might not be striking visually in the reconstructed SST, we can clearly see what model performs best by looking at its RMSE. Moreover, the spatial variability of the RMSE suggests that there might be some coherent patterns in the error made by the models. We investigated this question by looking at the difference between the spatial spectra of the reconstructed image with the original image (figure 4). Interestingly, the bicubic interpolation is providing less power than the original image at all spatial scales whereas the CNNs model are always providing more, suggesting that the level of complexity constructed by such models might be overestimating. In addition, the bicubic method does better for the most energetic wavelength. It is however less successful at longer or shorter wavelength than the CNN models, explaining the difference in PSNR. Further investigations will be needed to explore in more details the spatial dependence of the error.



Bicubic

RMSE Bicubic

ESPCN

RMSE ESPCN

FSRCNN

RMSE FSRCNN

SRCNN

RMSE SRCNN

VDSR

RMSE VDSR

Input

Original

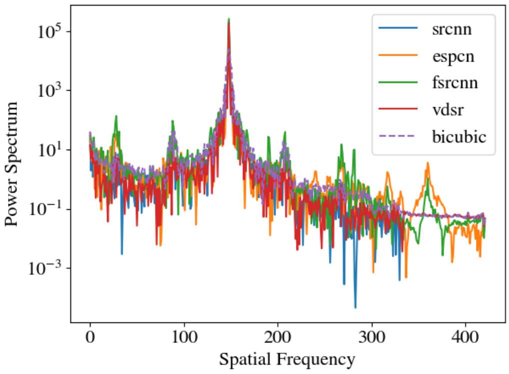
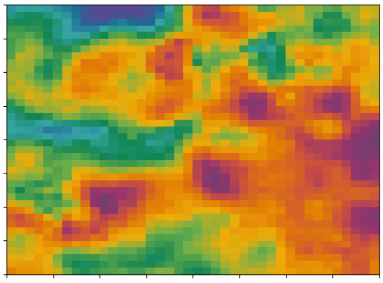


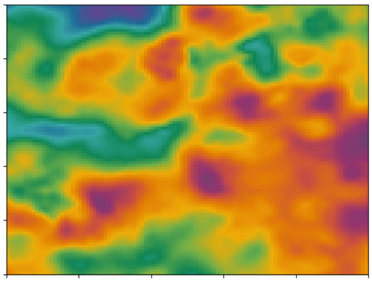
Figure 4: Difference of spatial power spectra between the original image and the reconstructed one for the entire SST field. Dotted lines denotes a positive difference (model underestimated original image) and plain lines a negative difference (model overestimated original image). The bicubic interpolation method is the only one underestimating the spectral power of the original image.

5.4 Application to actual satellite images

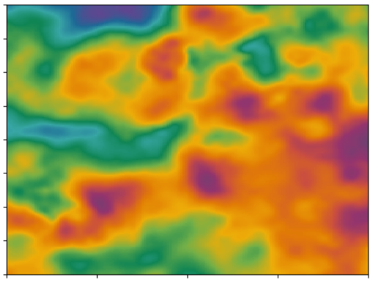
Because of the discrepancy between the distribution of data used for the training/testing process (mixed data from model, in-situ, satellite) and the actual data distribution intended for our application (satellite data), we need to make sure that our model works well with the targeted distribution. Hence, you will find below the result of the construction for the SST case (here again, analogy with SSH is analog). Here, the comparison is purely qualitative, because of the lack of referent dataset, but provide nice insights.



Input SST



Bicubic SST



Output SST with VDSR

6 Conclusion

This study addresses the application of deep learning to ocean remote sensing data. While deep learning has rapidly become the state-of-the-art framework for a variety of problems in computer vision, its applicability to remote sensing data, and more specifically ocean remote sensing, remains to be investigated. As case-study, we consider the reconstruction of high-resolution (HR) SST and SSH fields from low-resolution (LR) observations and investigate different architectures of convolutional neural network. Our experiments clearly points out the relevance of CNNs for the considered dataset with clear improvement over the bicubic interpolation for geophysical fields downscaling. In front of the success of this project, further work are planned to include the comparison of deep learning techniques with state-of-the-art methods for typical downscaling (e.g. EOF based approach, see He et al., 2015), the exploration of a broader range of hyper-parameters, and the design of new CNNs architectures.

References

[1] C.-y. Yang, C. Ma, and M.-h. Yang, "Single-Image Super-Resolution : A Benchmark," pp. 372–386, 2014.

[2] Y. Wang, L. Wang, H. Wang, and P. Li, "End-to-End Image Super-Resolution via Deep and Shallow Convolutional Networks," pp. 1–10.

[3] E. T. H. Zurich, "Seven ways to improve example-based single image super resolution."

[4] L. N. Thomas, "Submesoscale Processes and Dynamics," pp. 17–38, 2008.

[5] W. Shi, J. Caballero, F. Husz, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network."

[6] J. C. Mcwilliams and J. C. Mcwilliams, "Submesoscale currents in the ocean Subject Areas : Author for correspondence :," 2016.

[7] G. Madec, "NEMO ocean engine," no. 27, 2016.

[8] C. Ledig, L. Theis, F. Husz, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network."

[9] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-Recursive Convolutional Network for Image Super-Resolution."

[10] ——, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks."

[11] L. He, R. Fablet, B. Chapron, and J. Tournadre, "Learning-Based Emulation of Sea Surface Wind Fields From Numerical Model Outputs and SAR Data," vol. 8, no. 10, pp. 4742–4750, 2015.

[12] R. Fablet, "Deep Learning for Ocean Remote Sensing : An Application of Convolutional Neural Networks for Super-Resolution on Satellite-Derived SST Data."

[13] C. J. Donlon, M. Martin, J. Stark, J. Roberts-jones, E. Fiedler, and W. Wimmer, "Remote Sensing of Environment The Operational Sea Surface Temperature and Sea Ice Analysis (OSTIA) system," *Remote Sens. Environ.*, vol. 116, pp. 140–158, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.rse.2010.10.017

[14] C. Dong, C. C. Loy, and X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network."

[15] C. Dong, C. C. Loy, K. He, and I. Ntroduction, "Image Super-Resolution Using Deep Convolutional Networks," pp. 1–14.

[16] R. Ferrari and D. L. Rudnick, "Thermohaline variability in the upper ocean," *J. Geophys. Res.*, vol. 105, no. C7, p. 16857, 2000.