A neural network super-resolution approach for the reconstruction of coastal sea states

J. Kuehn[1], S. Abadie[1], V. Roeber[1]

[1]Université de Pau et des Pays de l'Adour, E2S-UPPA, chair HPC-Waves, SIAME, Anglet, France

Key Points:

- Deep learning super-resolution methods can be used to reconstruct coastal sea states.
- The method can substitute high-resolution results from low-resolution computations.
- Super-resolution can provide local high-resolution wave forecasts without additional computations in operational models.

Corresponding author: Jannik Kuehn, `jannik.kuehn@outlook.de`

Abstract

In this paper, a neural-network-based super-resolution technique is applied to the reconstruction of significant wave height and other sea state variables calculated over coarse meshes by a spectral wave model. The potential of the technique is demonstrated in a case study and the efficiency of the training process as well as the requirements with respect to data quality are analyzed. In this particular example, reasonable accuracy is achieved using only one year of training data with the help of traditional Machine Learning methods like Transfer Learning and Data Augmentation. The presented method leads to up to 50-times lower computation time in comparison to an equivalent traditional direct modeling approach at fine resolution. Overall, incorporation of the presented method into major wave forecasting systems has the potential to allow for the creation of "zoomed-in" areas of interest without the requirement for supplementary calculations at higher resolution.

Plain Language Summary

Accurate wave height forecasts on a daily basis are essential for many coastal communities worldwide. Though multiple operational wave models provide access to global wave forecasts, local high-resolution output is often not available. In this paper, we propose an approach based on neural networks to convert low-resolution computations into higher resolution. In our case study, this method is reasonably accurate, it enhances the resolution up to 16 times, and can be more than 50 times faster than what is required for the actual high-resolution calculation. After a one-time training process of the neural network, it has the potential to be incorporated into major forecasting systems, allowing to "zoom" into specific regions of interest in real-time.

1 Introduction

Many coastal communities rely on daily wave height forecasts for the purpose of safety and hazard mitigation. Over the last decades, improvements in numerical methods have lead to more accurate predictions of sea states, which had a considerable influence on marine transport, fisheries, and ocean engineering. Global efforts of producing ocean observation networks paired with national ocean services (e.g., the National Oceanic and Atmospheric Administration), national and international buoy networks (e.g., the CANDHIS network in France - `https://candhis.cerema.fr`), and global ocean wave models like SWAN (Booij et al., 1999) and WAVEWATCH III (Tolman, 2009) provide critical information - often in real-time. However, high-resolution data, particularly in the coastal zone, is often missing due to computational constraints associated with large computational domains in combination with fine meshes. Nevertheless, the need for local high-resolution data still exists at this scale, since local forecasts for wave-driven processes are increasingly based on refined computations.

The recent rise in interest in Machine Learning and its ongoing integration into natural sciences can be largely attributed to its ability to perform various computational tasks faster and with similar accuracy after an initial training phase has been completed. In this article we present a super-resolution approach to decrease computation time for forecasts of up to 50 times compared to traditional direct modeling of an equivalent domain at fine resolution. The idea is based on training a neural network that converts low-resolution (LR) results to a higher resolution for a specific study area. The main advantage of this approach is that the computation of a low-resolution forecast and the subsequent conversion are considerably faster than a direct high-resolution (HR) computation. Furthermore, once trained, this model has the potential to be directly linked to the output of global ocean wave models, thus providing local high-resolution results on the fly without the need for costly direct calculations.

Super-resolution has been an active field of research in computer vision for almost a decade. Recently, it also started to be applied to fluid mechanics - especially in the field of turbulence (Kim et al., 2020; Gao et al., 2021). In ocean sciences, deep-learning-based super-resolution was already successfully applied to remote sensing data of sea-surface temperature (Ducournau & Fablet, 2016; Su et al., 2021). Furthermore, treating gridded bathymetric data as digital images, Sonogashira et al. (2020) enhanced the resolution of coarse bathymetric charts by outperforming naive interpolation. This showcased that super-resolution might considerably reduce the amount of measurements needed. Even though previous papers addressed the need for improving local forecasts with neural networks (Londhe et al., 2016; James et al., 2018), this is the first time, to the best of our knowledge, that super-resolution techniques are applied to coastal wave modeling.

In the present paper, we propose to apply the Downsampled Skip-Connection Multi-Scale (DSC/MS) neural network introduced by Fukami et al. (2019) to convert low-resolution SWAN computations to an up to 16-times higher resolution. In the first part of this paper, the workflow is explained with a focus on data processing and potential data enhancement methods. We then present the results from an application of the model to a study case for the nearshore area at Biarritz (SW France). Lastly, we discuss our results, the presently existing limitations of this approach, and the potential for future research.

2 Materials and Methods

As a first step to construct a framework for neural-network-based super-resolution it is necessary to obtain a training and test data set from a coastal wave model .

2.1 SWAN and Data Pre-processing

For the creation of a data set with matching LR and HR images, we compute various quantities including significant wave height H_S, peak wave period T_P and mean wave direction θ with the third-generation spectral wave model SWAN (Simulating WAves Nearshore) (Booij et al., 1999) for a part of the coastal area near Biarritz (Fig. 1).

The area of interest is nested inside a coarser grid that is forced by homogeneous spectral boundary conditions taken from the HOMERE hindcast database (Edwige et al., 2013), at the location of the Donostia buoy (i.e., around 35km from Biarritz in a water depth of about 450 m). The wave data set, which is considered for this work covers a two-year time period spanning from January 01, 2018 to December 31, 2019. For the HR nested grid, we chose a grid with 160 x 160 quadratic cells (8 km in x- and y-direction with Δx = Δy = 50 m), which is of sufficient resolution to capture small variations in the wave regime, while enabling down-scaling by multiples of two. Consequently, the corresponding LR grids are of size 40 x 40, 20 x 20, and 10 x 10.

The bathymetry data was obtained from the publicly-available 0.001 ° precision Digital Terrain Model "MNT bathymétrique de façade Atlantique" provided by the French Service Hydrographique et Océanographique de la Marine (SHOM) (SHOM, 2015) and interpolated with Octave (Eaton et al., 2020) to the various grid sizes.

With the objective to demonstrate the feasibility of the application of neural-network-based super-resolution to ocean wave modeling, we use basic, but still realistic settings for the SWAN computations. In particular, wind forcing, wind growth, and white-capping are not taken into account. The water level was set constant and equal to the mean wa-

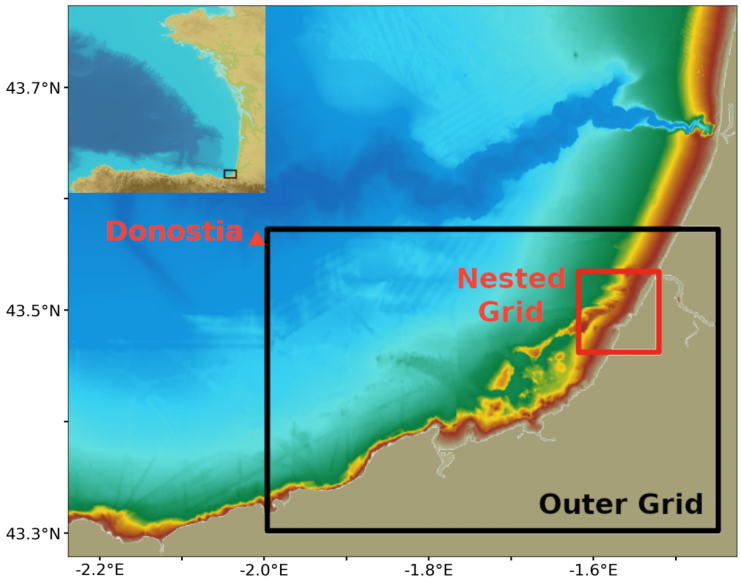

Figure 1. Bathymetry map of the study area and the locations of the outer and the nested grid. (Top left panel : location of the study site on the French Atlantic coast.)

ter level 2.25 m. Depth-induced wave breaking was modeled with constant values $\alpha = 1$ and $\gamma = 0.73$ adapted from Battjes and Stive (1985). Bottom friction is based on Madsen et al. (1988) with a constant coefficient of 0.085.

We note that the SWAN model returns NANs for cells that fall across the coastline. Since Neural Networks cannot work with NANs, they have to be addressed separately as discussed in the workflow in section 2.6.

2.2 Neural-Network architecture

For our neural network architecture, we used the hybrid Downsampled Skip-Connection Multi-Scale (DSC/MS) model presented in Fukami et al. (2019), given their success in reconstructing turbulent flows and its easier implementation and much lower training time as compared to, for example, Generative Adversarial Networks (GAN) (Stengel et al., 2020). Since quality literature exists on the topic of Deep Learning, Neural Networks, and more specifically Convolutional Neural Networks (CNN) and their application to image processing, we refer the interested reader to one of the following sources for an introduction to the topic (Guo et al., 2016; Rawat & Wang, 2017; Aloysius & Geetha, 2018). As for the model employed, it is based on a CNN, but is modified to improve the reconstruction of both large and small-scale patterns. The modifications include data compression, which makes the network more robust to rotations and translations (Ngiam et al., 2010) and skipped connections, that reduce difficulties concerning the convergence of the weights often seen in deep CNNs (He et al., 2016). This is paired with the multi-scale model by Du et al. (2018), which comprises multiple CNN filters of varying length to capture a range of scales. For more detailed information, we refer to the original article of the author.

2.3 Data Augmentation

Improvements of the predictions made by the neural network can be achieved though an artificial increase of the size of the data set by modifying the labeled data in a real-

istic way. This is a common procedure, especially in the area of image recognition (Perez & Wang, 2017). As an example, a picture of a recognizable object can be mirrored horizontally or the brightness and contrast of the image can be changed, but the object in the picture would still be recognizable. These artificial modifications help the neural network to generalize better and consequently enhance the results (Shorten & Khoshgoftaar, 2019).

In this study, we perform data augmentation on the significant wave height data set by adding a random uniformly distributed offset between zero and five meters to each data instance, to artificially account for wave heights in different regimes. The distribution parameters are arbitrary but realistic and worked well in our case. Excessively small or large offsets deteriorate the performance of the data augmentation. It should be noted that the same offset has to be applied to the input and reference data instances.

2.4 Transfer Learning

Another commonly used technique when working with neural networks is transfer learning (Pan & Yang, 2010), where some or all of the weights of another already trained neural network are reused. Indeed, the lower layers of a network tend to learn small-scale features that might contain useful information for similar tasks.

Here, we first create artificial LR input by downsampling the reference HR samples with average pooling. This operation computes the average over a so-called pooling window that slides with a given stride s over the whole image. In our case the given HR grid size is 160x160 and the pooling window is 4x4 (resp. 8x8, or 16x16) with a stride of $s = 4$ (resp. 8, or 16). The output is a LR image with a 40x40 (resp. 20x20, or 10x10) grid, where each pixel corresponds to an average of a 4x4 (resp. 8x8, or 16x16) part of the original data instance.

We then train the neural network with the average pooled LR images as an input and the corresponding HR images as the references. Models trained from average pooled data usually predict well, given that comparatively much of the information is retained after the downsampling. The weights obtained by training on the averaged pool data are taken to initialize the weights of the actual neural network, which is trained on the directly modeled data. This leads to faster convergence and commonly improves the mean square error (MSE). However, the pre-training on the averaged pool model can take about as long as training the actual execution of the model, which makes transfer learning computationally expensive. Nonetheless, training is a one-time cost, which is typically compensated by the considerably quicker run time after training.

2.5 Bicubic Interpolation

It is also possible to upsample the LR image to the reference grid size with bicubic interpolation to further minimize the prediction error. This is a standard technique in image super-resolution (Dong et al., 2016) and it helped to improve the model results in our study for certain configurations.

2.6 Workflow

Our workflow is summarized in Fig. 2. First, the data set is obtained by computing a coastal wave model over two grids - one with high and the other one with low resolution. The data is then split into "snapshots", i.e. into files containing a particular distribution of a variable (e.g., H_s) across the numerical domain at one time step. In our case, the sampling interval is one hour in both the high and low resolution data set. As a next step, it is customary to set aside a certain fraction of the whole data set for testing purposes. Then, the rest of the data is divided into a training and validation set. The

former is used by the model to adjust the weights, whereas the latter is for hyper-parameter tuning and performance assessment of the model after each training epoch. All of the low-resolution data, including the test data, is upsampled by a simple nearest-neighbor scheme to enable passing the input to the neural network. This means that if the low-resolution grid has 4-times less cells, each pixel gets copied 4 times to fill the high-resolution grid size (Sonogashira et al., 2020). Alternatively, this step is skipped and replaced by upsampling with a bicubic interpolation method. In all cases, the NaNs resulting from the coastline of the coastal wave model are replaced by zero, as the neural network handles only numbers. If necessary, the training set is enhanced artificially by performing data augmentation. Alternatively, or additionally, it is also possible to train a pre-model, on average-pooled data, of which the weights are then used for transfer learning. After the optional data treatment the model is trained with ADAM optimization (Kingma & Ba, 2014) and early stopping (Prechelt, 1998). It is then evaluated with the test set to obtain a realistic estimation of the performance of the neural network. If required, certain parameters are adjusted and the model is retrained. Lastly, after a satisfactory result is reached, the model is trained with all the data available, including the test data to then deploy the neural network.

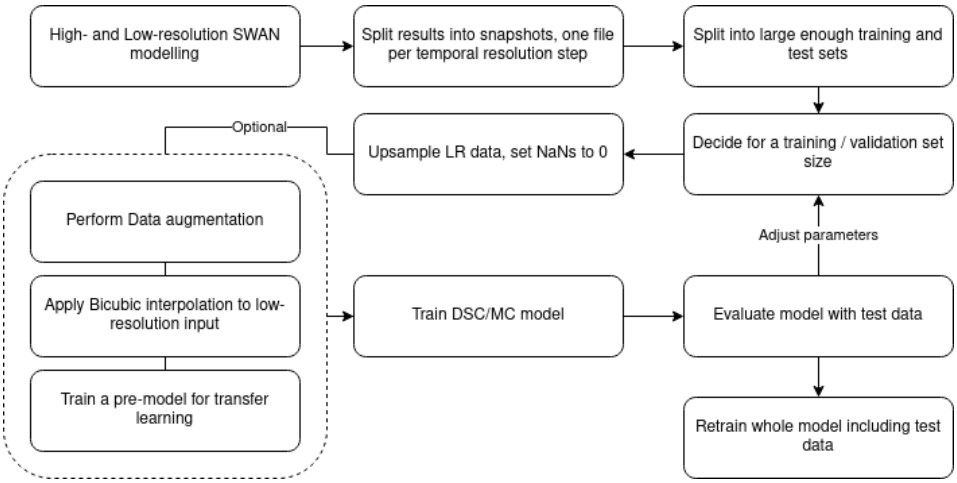

Figure 2. Workflow for training a DSC/MS model from data of a coastal wave model.

2.7 Application

In the present study, we run a pair of high- and low-resolution SWAN computations for the two year period between January 01, 2018 and December 31, 2019. For each model, we use the first year for training (80 %) and validation (20 %), and the second year solely for testing to get a realistic estimation of the performance of the model over various sea states. We perform data augmentation as described in section 2.3 on 2000 data instances randomly sampled from the training set. Finally, the three models are trained to convert modeling results of significant wave height to a grid size of 160x160 from one of 40x40, 20x20, and 10x10, respectively.

3 Results

In Fig. 3 the low-resolution inputs along with the corresponding predictions and their MSE are shown. For comparison, the high-resolution reference snapshot is displayed as well. The data instance was chosen as being close to the average wave regime.

In the three cases, the high-resolution computations are well approximated by the reconstructions. The prediction not only correctly captures most of the wave features, but also reconstructs the original coastline in a sense that the neural network predicts negligible values (< 0.01) for values on land. Setting values lower than a small threshold to NaN results in a near-perfect reconstruction of the original coastline. This turns out remarkably well in particular for the third model with an original grid size of 10x10 where the low-resolution input only provides very coarse information regarding wave height patterns and coastline.

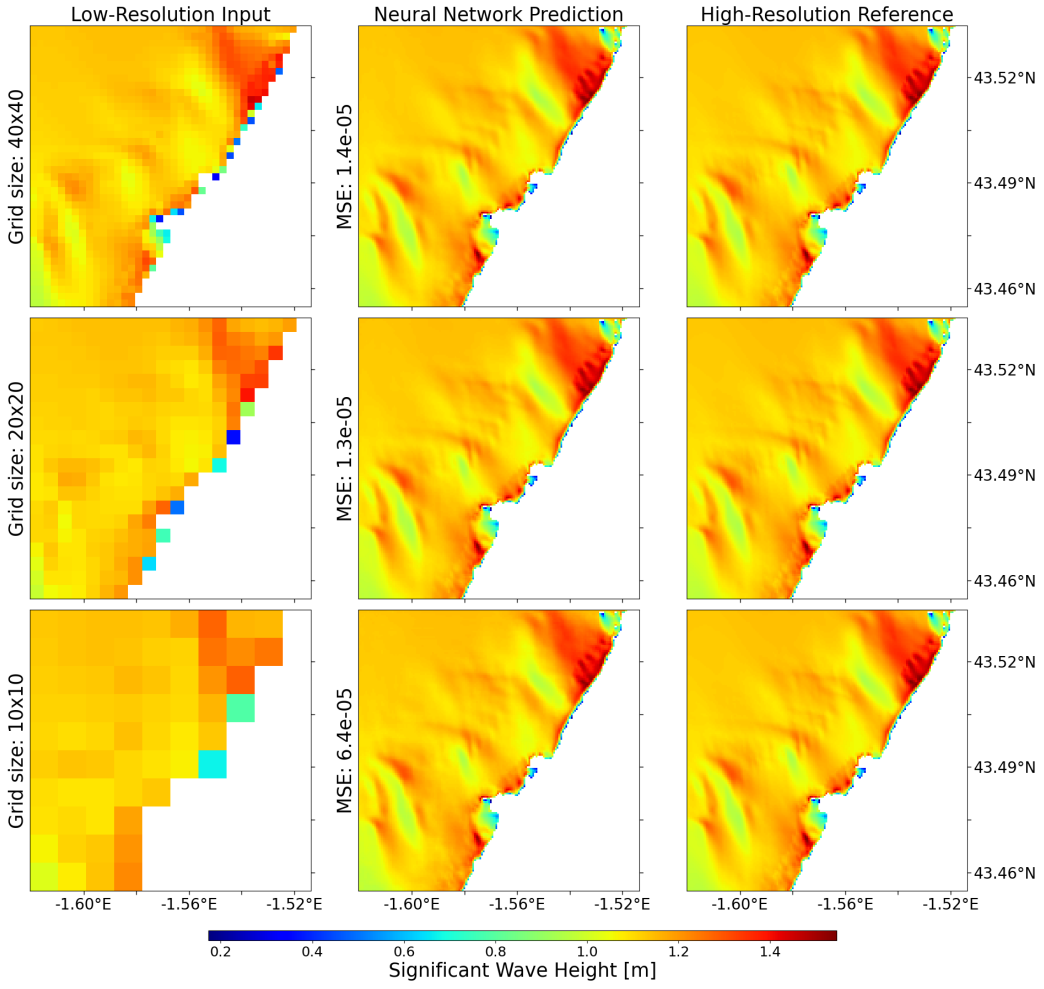

Figure 3. Reconstructions of direct high-resolution computations from different lower resolution calculations. The MSE is shown for the predictions. The data instance is from April 21, 2019 at midnight.

In general, the results improved after performing data augmentation or transfer learning. In certain cases, mostly in models converting 40x40 grid sizes, application of bicubic interpolation also reduced the mean-square error. For other grid sizes, however, predictions usually deteriorated. Here, only the results without any prior data "enhancement" are shown to demonstrate the feasibility of the method in its simplest form. Various combinations of the techniques mentioned here, paired with careful hyperparameter tuning, and possible normalization of the data set can significantly improve the predictions. However, a thorough analysis is beyond the scope of this article.

Models trained on the peak wave period or direction performed reasonably well. However, the overall patterns were not reproduced as sharply and accurately as in the case of the significant wave height. This might be due to the more heterogeneous nature of those two variables, as well as larger absolute values. The latter might be mitigated by prior normalization.

For a quantitative overview of the most common reconstruction errors, we compute the prediction for every data instance in the test data set and subtract it from the corresponding reference to obtain the deviation. We then average the LR input, the prediction, the HR references and the errors, which are shown in Fig. 4 for the reconstruction of a 10x10 grid. The mean errors are commonly the most prominent around the visible patterns, with a tendency to underestimate large and overestimate small wave heights. Nonetheless, the average error amounts only to a few centimeters, in consistency with errors of single data instances. We found that the largest errors occurred in the reconstruction of very high and very low sea states. This is to be expected, given that those sea states were the least frequent in our data set and were thus the least represented during training. Data augmentation is able to mitigate the errors by generating artificially more representative sea states, but it is efficient only to a certain extent. This is likely due to a corresponding structural change of the patterns in the extreme sea states, which is not taken into account in the data augmentation process. Possible approaches to minimize the error further are discussed later.

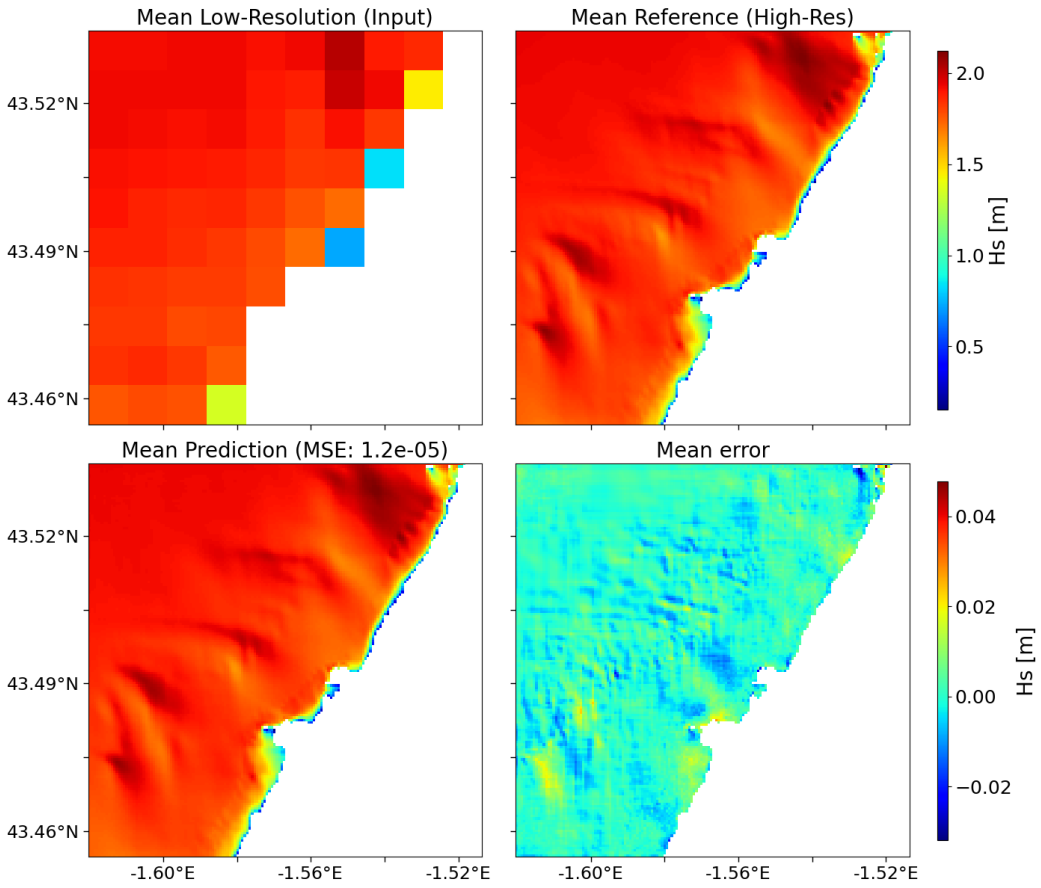

Figure 4. Average reconstruction error over the study area for a 16-times increase in resolution. The mean was taken over the whole test data set.

3.1 Comparison of Computation Time

To outline the gains in computation time by this approach, we listed the time needed at all the steps of the method. The SWAN computations were run with 12 parallel threads on an Inter Core i7-9750H processor with 6 cores and 12 threads. For the pre-processing, we used the same processor without parallel threading and for training and prediction of the neural network, a Nvidia GeForce RTX 2070 Max-Q was used.

As a first step, the outer grid of the SWAN model was computed over the time range of 2 years, which took around 7.5 h. Similarly, computing the same time range in the nested, high-resolution 160x160 grid took also 7.5 h. Calculation of the same nested grid in lower resolutions (40x40, 20x20, 10x10) required only 35 min, 14 min, and 8 min, respectively.

Training times of the neural network can change significantly (on the order of 10s of minutes) for the same model, if the algorithm gets stuck in a local minimum. Usually pre-processing and training took 2 h - 4 h, but more specifically for our models, the time required were 2.5 h, 2.5 h, and 3.5 h, respectively. One has to note, however, that this is only a one-time cost, since in theory the model can be re-used for later calculations. In practice, occasional re-training with additional data would be advisable.

The actual prediction by the model is computed very fast, since, once the weights are determined after training, predictions are simple matrix multiplications. A conversion of 1000 data instances, which corresponds to around 40 days of hourly significant wave height, required only 1.9 s.

In summary, converting 2 years of data of 10x10 LR calculations to 160x160 HR, takes only around 8.5 min, including the time to run the LR SWAN computations, compared to the 7.5 h when modeling the domain directly in high-resolution, which is a more than 50-fold increase in computation time.

4 Summary and discussion

The DSC/MS model proposed by Fukami et al. (2019) is able to reconstruct high-resolution features of various sea state variables given a low-resolution image. Overall, this approach has the potential to reduce computation time for forecasts considerably. After a high, but only single computational effort of training the model, predictions can be obtained 50-times faster, compared to a standard high-resolution SWAN computation, with good accuracy. Furthermore, the speed and accuracy presented here are probably lower bound estimates, which can be ameliorated with a more elaborated pre-processing routine, an improved model architecture and careful training data sampling. Also, existing hindcast databases could be used to train the model extensively. Additionally, different loss equations and other neural network architectures like GANs, could improve the results substantially.

Instead of data augmentation a large or well-sampled data set could be used, which covers many possible wave conditions. This would also very likely improve the predictions. Here, we focus on data augmentation to demonstrate that even with relatively small training sample sizes, a robust prediction can be achieved for various wave conditions.

A drawback of our approach is that one model has to be trained for each wave variable, which increases the training time substantially. Nevertheless, it does not affect the prediction time. Training one model to predict all variables is possible, however, it deteriorates the results as commonly reported in literature (Schultz et al., 2021).

Moreover, the trained models are location-specific and, consequently, a model has to be trained for each new location. As discussed in section 2.4, the training time can be reduced with transfer learning. Another recent approach builds on incorporation of physical equations in the training process to produce so-called Physics-Informed Neural Networks (PINN) (Um et al., 2020; Gao et al., 2021). Commonly, this is done by adding particular terms, like the constraint of zero divergence for incompressible fluids in the loss equations (Raissi et al., 2019). This forces the neural network to not only produce more physically plausible results, but also helps it to better generalize. A thorough overview of PINNs can be found in Willard et al. (2020). While this approach appears difficult to implement in a spectral wave model like SWAN, it might be possible for other wave models relying on different governing equations.

Note also, that in comparison to other applications for instance in fluid mechanics, temporal coherence is much easier to achieve in our setting. This is presumably due to the easier structure of our problem. While a reconstruction of turbulent flows varies strongly in space and time, coastal waves do not exhibit the same amount of variation. For example, Xie et al. (2018) developed a sophisticated Generative Adversarial Network to achieve temporal coherence in their super-resolution reconstruction of smoke flow. However, we observed that for our simpler case temporal coherence was already given with the least complex model that we employed. Lastly, we considered only rectangular, uniform grids due to the simplicity of finding and computing high- and low resolution pairs. Given that many studies use more complex grid structures, more research has to be undertaken to generalize this approach to any type of grid.

Despite the current limitations of the super-resolution method applied to coastal wave modeling, it is able to produce high-resolution results and consistently reconstruct the underlying patterns with remarkable accuracy, while being considerably faster than traditional direct computations. Additionally, it has the potential to be more generic and accurate at equal computation time and could be used in the future for locally "zoomed-in" global wave models.

5 Open Research

5.1 Data Availability

For the creation of the low- and high-resolution sea state quantities we used the third-generation spectral wave model SWAN (Booij et al., 1999), version 41.31. It was forced by data from the HOMERE hindcast database (Edwige et al., 2013) at the location of the Donostia buoy covering the time range from January 1st 2018 to December 31st 2019. Bathymetry data from the 0.001° Digital Terrain Model "MNT bathy-métrique de façade Atlantique" provided by the French Service Hydrographique et Océano-graphique de la Marine (SHOM) (SHOM, 2015) and interpolated with Octave version 5.2.0 (Eaton et al., 2020). The Neural Network in this paper is based on the code and the implementation of Fukami et al. (2019). Pre-processing and training was done entirely in Python 3.9.7, using mostly the following libraries: Keras 2.4.3 (Chollet & Others, 2015) with Tensorflow 2.4.1 as a backend (Abadi et al., 2015), Pandas 1.3.4 (Reback et al., 2021), and Numpy 1.19.2 (Harris et al., 2020). For the creation of the figures Matplotlib 3.5 (Hunter, 2007) was used. The SWAN, pre-processing and training scripts are available through GitHub under `https://github.com/janfer95/SR_on_SWAN` (Kuehn et al., 2022).

Acknowledgments
This research was carried out under the framework of the joint laboratory KOSTARISK, which is part of the E2S UPPA program managed by the French National Research Agency (ANR-16-IDEX-0002) and supported by the French Government's "Investissements d'Avenir" (PIA). The joint laboratory KOSTARISK is co-funded by E2S UPPA, the AZTI Foun-

dation and the center Rivages Pro Tech of SUEZ. The authors acknowledge financial support from E2S UPPA, the Communauté d'Agglomération Pays Basque (CAPB), and the Communauté Région Nouvelle Aquitaine (CRNA) for the chair HPC-Waves and the European Union Horizon 2020 project ALPHEUS - DLV-883553. All of the code was written in Python. The Neural Network was implemented with the Keras library having Tensorflow as a backend. The code will be accessible through GitHub.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Retrieved from `https://www.tensorflow.org/`

Aloysius, N., & Geetha, M. (2018). A review on deep convolutional neural networks. *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, *2018-Janua*, 588–592. doi: 10.1109/ICCSP.2017.8286426

Battjes, J. A., & Stive, M. J. F. (1985). Calibration and verification of a dissipation model for random breaking waves. *Journal of Geophysical Research*, *90*(C5), 9159. doi: 10.1029/JC090iC05p09159

Booij, N., Ris, R. C., & Holthuijsen, L. H. (1999, apr). A third-generation wave model for coastal regions: 1. Model description and validation. *Journal of Geophysical Research: Oceans*, *104*(C4), 7649–7666. Retrieved from `http://doi.wiley.com/10.1029/98JC02622` doi: 10.1029/98JC02622

Chollet, F., & Others. (2015). *Keras.* `https://keras.io`.

Dong, C., Change, L. C., & Xiaoou, T. (2016). Accelerating the Super-Resolution Convolutional Neural Network. *European Conference on Computer Vision*, 391–407. doi: https://doi.org/10.1007/978-3-319-46475-6_25

Du, X., Qu, X., He, Y., & Guo, D. (2018). Single image super-resolution based on multi-scale competitive convolutional neural network. *Sensors*, *18*(3), 789.

Ducournau, A., & Fablet, R. (2016, dec). Deep learning for ocean remote sensing: an application of convolutional neural networks for super-resolution on satellite-derived SST data. In *2016 9th iapr workshop on pattern recogniton in remote sensing (prrs)* (pp. 1–6). IEEE. Retrieved from `http://ieeexplore.ieee.org/document/7867019/` doi: 10.1109/PRRS.2016.7867019

Eaton, J. W., Bateman, D., Hauberg, S., & Wehbring, R. (2020). GNU Octave version 5.2.0 manual: a high-level interactive language for numerical computations [Computer software manual]. Retrieved from `https://www.gnu.org/software/octave/doc/v5.2.0/`

Edwige, B., Christophe, M., Ardhuin, F., Accensi, M., Pineau-Guillou, L., & Lepesqueur, J. (2013). *A suitable metocean hindcast database for the design of Marine energy converters* (Vol. 3-4) [Article]. FRANCE. Retrieved from `https://marc.ifremer.fr/en/produits/hindcast_sea_states_homere` doi: https://doi.org/10.1016/j.ijome.2013.11.010

Fukami, K., Fukagata, K., & Taira, K. (2019). Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, *870*(Ml), 106–120. doi: 10.1017/jfm.2019.238

Gao, S., Huang, J., Li, Y., Liu, G., Bi, F., & Bai, Z. (2021, jan). A forecasting model for wave heights based on a long short-term memory neural network. *Acta Oceanologica Sinica*, *40*(1), 62–69. Retrieved from `https://link.springer.com/article/10.1007/s13131-020-1680-3` doi: 10.1007/s13131-020-1680-3

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. doi: 10.1016/j.neucom.2015.09.116

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cour-

napeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. Retrieved from https://doi.org/10.1038/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

James, S. C., Zhang, Y., & O'Donncha, F. (2018, jul). A machine learning framework to forecast wave conditions. *Coastal Engineering*, *137*, 1–10. doi: 10.1016/j.coastaleng.2018.03.004

Kim, H., Kim, J., Won, S., & Lee, C. (2020). Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 1–34. doi: 10.1017/jfm.2020.1028

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kuehn, J., Abadie, S., & Roeber, V. (2022). *janfer95/SR_on_SWAN: Initial Release.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.5910948 doi: 10.5281/zenodo.5910948

Londhe, S. N., Shah, S., Dixit, P. R., Nair, T. M., Sirisha, P., & Jain, R. (2016). A Coupled Numerical and Artificial Neural Network Model for Improving Location Specific Wave Forecast. *Applied Ocean Research*, *59*, 483–491. doi: 10.1016/j.apor.2016.07.004

Madsen, O. S., Poon, Y.-K., & Graber, H. C. (1988, jan). SPECTRAL WAVE ATTENUATION BY BOTTOM FRICTION: THEORY. *Coastal Engineering Proceedings*, *1*(21), 34. Retrieved from https://journals.tdl.org/icce/index.php/icce/article/view/4241 doi: 10.9753/icce.v21.34

Ngiam, J., Chen, Z., Chia, D., Koh, P., Le, Q., & Ng, A. (2010). Tiled convolutional neural networks. *Advances in neural information processing systems*, *23*.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. doi: 10.1109/TKDE.2009.191

Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning.

Prechelt, L. (1998). Early stopping-but when? In *Neural networks: Tricks of the trade* (pp. 55–69). Springer.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, *378*(October), 686–707. Retrieved from https://doi.org/10.1016/j.jcp.2018.10.045 doi: 10.1016/j.jcp.2018.10.045

Rawat, W., & Wang, Z. (2017, sep). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, *29*(9), 2352–2449. Retrieved from http://arxiv.org/abs/1803.01446https://direct.mit.edu/neco/article/29/9/2352-2449/8292 doi: 10.1162/neco_a_00990

Reback, J., Jbrockmendel, McKinney, W., den Bossche, J. V., Augspurger, T., Cloud, P., ... Seabold, S. (2021). *pandas-dev/pandas: Pandas 1.3.4.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.5574486 doi: 10.5281/zenodo.5574486

Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H., ... Stadtler, S. (2021). Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *379*(2194). doi: 10.1098/rsta.2020.0097

SHOM. (2015). *MNT Bathymétrique de façade Atlantique.* doi: http://dx.doi.org/10.17183/MNT_ATL100m_HOMONIM_WGS84

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1). Retrieved from https://doi.org/10.1186/s40537-019-0197-0 doi: 10.1186/s40537-019-0197-0

Sonogashira, M., Shonai, M., & Iiyama, M. (2020). High-resolution bathymetry by deep-learning-based image superresolution. *PLoS ONE*, *15*(7), 1–19. Retrieved from http://dx.doi.org/10.1371/journal.pone.0235487 doi: 10.1371/journal.pone.0235487

Stengel, K., Glaws, A., Hettinger, D., & King, R. N. (2020). Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences of the United States of America*, *117*(29), 16805–16815. doi: 10.1073/pnas.1918964117

Su, H., Wang, A., Zhang, T., Qin, T., Du, X., & Yan, X.-H. (2021). Super-resolution of subsurface temperature field from remote sensing observations based on machine learning. *International Journal of Applied Earth Observation and Geoinformation*, *102*, 102440. Retrieved from https://doi.org/10.1016/j.jag.2021.102440 doi: 10.1016/j.jag.2021.102440

Tolman, H. L. (2009). User manual and system documentation of WAVEWATCH III TM version 3.14. *Technical note, MMAB Contribution*, *276*, 220.

Um, K., Brand, R., Yun, Fei, Holl, P., & Thuerey, N. (2020). Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers. *34th Conference on Neural Information Processing Systems*, *1*(c), 1–37. Retrieved from http://arxiv.org/abs/2007.00016

Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). Integrating Physics-Based Modeling with Machine Learning: A Survey. , *1*(1), 1–34. Retrieved from http://arxiv.org/abs/2003.04919

Xie, Y., Franz, E., Chu, M., & Thuerey, N. (2018). tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Transactions on Graphics*, *37*(4), 1–15. Retrieved from https://dl.acm.org/doi/10.1145/3197517.3201304 doi: 10.1145/3197517.3201304