Machine Learning (2023) 112:1107–1130
https://doi.org/10.1007/s10994-021-05977-w

Satellite derived bathymetry using deep learning

Mahmoud Al Najar[1] · Grégoire Thoumyre[2] · Erwin W. J. Bergsma[3] · Rafael Almar[2] · Rachid Benshila[1] · Dennis G. Wilson[4]

Received: 14 April 2020 / Revised: 13 January 2021 / Accepted: 5 April 2021 /
Published online: 22 July 2021

Abstract
Coastal development and urban planning are facing different issues including natural disasters and extreme storm events. The ability to track and forecast the evolution of the physical characteristics of coastal areas over time is an important factor in coastal development, risk mitigation and overall coastal zone management. Traditional bathymetry measurements are obtained using echo-sounding techniques which are considered expensive and not always possible due to various complexities. Remote sensing tools such as satellite imagery can be used to estimate bathymetry using incident wave signatures and inversion models such as physical models of waves. In this work, we present two novel approaches to bathymetry estimation using deep learning and we compare the two proposed methods in terms of accuracy, computational costs, and applicability to real data. We show that deep learning is capable of accurately estimating ocean depth in a variety of simulated cases which offers a new approach for bathymetry estimation and a novel application for deep learning.

Keywords Satellite-derived bathymetry · Earth observation · Machine learning · Deep learning · Regression

1 Introduction

Coastal areas are currently facing environmental and resource problems aggravated by anthropogenic pressure and over-exploitation. The environmental context of extreme events (eg, floods and coastal erosion) combined with anthropogenic pressure is a limiting factor for coastal development and flood-risk exposure. The state and evolution of the nearshore

M. Al Najar was partly funded by the IRD (Institut de Recherche pour le Développement) and the STAE Foundation (Sciences et Technologies pour l'Aéronautique et l'Espace) and is currently supported by the CNRS (Centre National de la Recherche Scientifique). E. Bergsma was partly funded by the IRD (Institut de Recherche pour le Développement) and was part of the CNES (Centre National d'Études Spatiales) post-doctoral fellowship-program.

✉ Mahmoud Al Najar
mahmoud.al.najar@legos.obs-mip.fr

Extended author information available on the last page of the article

bathymetry over time is an important element in determining coastal risk exposure, that must be considered for coastal development and planning, but the acquisition of in-situ measured data is time-consuming and expensive, and not always possible due to the energetic nature of breaking waves in the coastal zone, and hence, performing bathymetric measurements is a challenge. Reliable, fast, and inexpensive estimates of ocean depth are increasingly necessary for coastal areas.

Recently, remote sensing tools have emerged as sources of inexpensive data that can be used to estimate and forecast coastal dynamics and features. These tools range from shore-based (Almar et al. 2009; Holman et al. 2013) or drone-mounted video (Bergsma et al. 2019a) cameras, to space-borne satellite constellations (Almar et al. 2019; Bergsma et al. 2019b). Shore-based solutions are limited to the spatial concentration and temporal sparsity of data, two issues which prohibit the estimation of coastal dynamics on a large scale. On the other hand, satellite missions such as the Sentinel-2 Constellation provide us with high-resolution images having global coverage and at a high revisit frequency (every 5 days with Sentinel-2) (Drusch et al. 2012), covering the coastal zone (Bergsma and Almar 2020). The process of estimating water depth from satellite imagery is referred to as Satellite Derived Bathymetry (SDB). SDB is an ongoing topic of research due to its massive potential impact on our ability to forecast coastal morphodynamics and related marine geological hazards on a large scale.

Deep Learning (DL) is a field of machine learning algorithms which have undergone massive development over the last decade and have drawn attention by demonstrating powerful capabilities in a variety of different fields (Goodfellow et al. 2016). One of the most successful applications of DL to date has been computer vision, demonstrated through applications in image classification, processing, and generation (Simonyan and Zisserman 2014). A natural extension to this application is the use of DL in remote sensing through the automatic processing of satellite images. A growing body of recent work has applied DL to satellite image classification or segmentation for a variety of applications (Liu et al. 2017; Iglovikov et al. 2017). These approaches often use deep learning to identify features in satellite images, such as the detection of vehicles.

In this work, we present what we believe to be the first application of deep learning to the problem of bathymetry estimation using wave physics. We create a synthetic dataset with a wide range of bathymetries and wave conditions, to be used in order to test the feasibility of the two proposed deep learning-based SDB methods, showcase their capabilities and understand their limitations, along with the different constraints that should be taken into account when dealing with real bathymetric measurements and satellite data.

The layout of this article is as follows. In Sect. 2 we discuss the problem of bathymetry estimation, the state-of-the-art for estimation methods, and machine learning approaches to bathymetry estimation and similar problems. We present two deep learning approaches for SDB in Sects. 4 and 5. The first method, Deep Large-Scale Reconstruction of Bathymetry (DLSRB), uses an encoder-decoder architecture to reconstruct the bathymetry over an entire 2D area. The second method, Deep Single-Point Estimation of Bathymetry (DSPEB), uses a convolutional neural network to estimate the average depth of individual local areas, which we then iterate to reconstruct the full bathymetry. The accuracy of these methods is analyzed on simulated bathymetry cases in Sect. 6. A general discussion and future works are presented in Sect. 7.

2 Background

In this section, we first describe the problem of bathymetry estimation, and the state of the art for physics-based approaches to the problem of satellite derived bathymetry. We then present other machine learning and deep learning-based literature related to our work.

2.1 Satellite derived bathymetry

Coastal bathymetry is paramount in understanding and forecasting coastal vulnerability (Benveniste et al. 2019). However, accurate bathymetries are often sparse in space and time due to the difficult and time-intensive nature of in-situ echo-sounding measurements. Remotely sensed bathymetry, through e.g. depth inversion and in particular using space-borne sensors, enables frequent and spatially dense bathymetries. Satellite-derived bathymetry from optical space-borne sensors are often derived using two approaches: that link water depth to (1) the radiative transfer of light in water as a link to water depth, and (2) wave kinematics through the dispersive relationships. Each approach has its strengths, limitations, and scope of applications (Salameh et al. 2019; Melet et al. 2020). In radiative transfer techniques, bathymetry is calculated based on the attenuation of radiance as a function of depth and wavelength in the water column using analytical or empirical models (Lee et al. 1999). Analytical imaging models are typically based on radiative transfer and the optical properties of water that impact how light is attenuated or deflected. Empirical imaging models rely on the statistical relationship between pixel-intensities and depth. For radiative transfer approaches, it generally holds that strong correlations between estimated and measured water depths are found in clear, calm waters. These models are strongly dependent on and largely bounded by the maximum penetration depth of sunlight which varies according to different seasons, locations and bottom reflectance. Alternatively, depth inversion approaches based on wave kinematics have been applied to a variety of data-types, collected from different remote sensing tools, ranging from modelled analysis (Bergsma and Almar 2018), laboratory experiments (Catalán and Haller 2006) to local shore-based analysis using time-series data (Stockdon and Holman 2000; Plant et al. 2008; Almar et al. 2009; Holman et al. 2013), airborne video data (Bergsma et al. 2019a; Brodie et al. 2019) and ultimately space-borne sensors including the IKONOS satellite (Abileah 2006), SPOT5 (de Michele et al. 2012; Poupardin et al. 2014, 2016), Sentinel-2 (Bergsma et al. 2019b) and Pleiades (Almar et al. 2019). Wave kinematic approaches exploit temporal information embedded in (satellite) images, in order to extract wave characteristics such as wavenumber and wave celerity that are then used to invert water depths. For the wave kinematics approach, wave signatures have to be visible in the image; while this can be a challenge in very clear waters with short waves, this approach has strong potential to estimate water depths in a large range of coastal waters (Bergsma and Almar 2020) such as different sandy or rocky bed types, turbid waters, with a theoretically deeper water limit compared to the previously mentioned methods based on the radiative transfer of light in water. In this study, we therefore focus on the wave kinematics approach.

2.2 Related works

Building on the existing literature of machine learning applications to wave modelling and bathymetry estimation, this paper presents the first application, to our knowledge, of deep learning to derive satellite-derived bathymetry based on wave kinematics. Machine learning has been used to replace classical wave models utilising time-series data from ocean buoys in Savitha et al. (2017) and Kumar et al. (2018), which use ensembles of feed-forward neural networks to predict wave height. Vojinovic et al. (2013) uses Support Vector Machines to perform water depth estimation based on a transformed ratio between the different bands (blue and green) of NASA EO-1's multi-spectral satellite images, demonstrating skill to predict depths to 15 m water depth. In Sagawa et al. (2019), Random Forests are used to perform a three-step analysis of several multi-band Landsat 8 Surface Reflectance products to create a shallow-water bathymetry map (0–20 m depth) for a specific site. In these works, the input data is transformed to render it accessible for the machine learning models; the goal of our work, however, is to offer models which can be applied to satellite products with minimal preprocessing procedures.

Earth Observation and remote sensing have recently seen a wide adoption of deep learning methods, largely due to the capacity of Convolutional Neural Networks (CNNs) to perform both satellite image processing and feature analysis. A review of deep learning and its application to Earth Observation is given in Zhu et al. (2017), and a more recent review focused on the different remote sensing applications is found in Ma et al. (2019). There is a strong trend towards image segmentation and classification tasks, such as object detection, which reflects the general trend in deep learning; a review on the application of deep learning methods to object detection and image segmentation tasks in Earth Observation is given in Hoeser and Kuenzer (2020). On the other hand, regression tasks, which require the estimation of continuous values, are less common; examples include Ham et al. (2019), where CNNs were used for El Niño/Southern Oscillation (ENSO) forecasting based on global ocean temperature distribution observations, and Wang et al. (2016), which uses CNNs to estimate ice concentration from synthetic aperture radar (SAR) scenes.

Early applications of neural networks to the problem of bathymetry estimation made use of a multi-layered perceptron to relate the spectral radiance in satellite image products to the depths of the seabeds underlying the imaged areas (Sandidge and Holyer 1998). More recent work made use of deep neural networks to perform bathymetry inversion based on the relation between the radiative transfer of light in water and the underlying water depth (Dickens and Armstrong 2019). As previously mentioned, color-based SDB methods are limited to clear and calm waters and are dependent on multiple location-specific parameters. Our work focuses on the wave-based method and demonstrates that deep learning models can learn to perform water depth estimation based on wave kinematics.

A common problem in deep learning is the magnitude of labeled data required to train neural networks; while satellite images are readily available, the labels required for training are often difficult to obtain. For the ocean temperature forecasting model in Ham et al. (2019), this problem is mitigated by first training their model on simulated data before applying transfer learning techniques to enable the model to handle real observations. Using this approach, Ham et al. (2019) report a positive improvement in performance on real data as a result of using transfer learning. The problem of model transfer from synthetic to real data in deep learning is well-studied (Zhuang et al. 2020) and it has been demonstrated that synthetic priors lead to better performance when compared to non-transfer models which learn fully on real data (Bird et al. 2020). In this preliminary work, we use a synthetic dataset generated to resemble Sentinel-2 images and allow studying extreme bathymetry features, as we discuss in the next sections. We believe that these first experiments in applying deep learning to bathymetry estimation will allow for the transfer of models trained on synthetic data to real data and will also provide insight on this application in terms of problem design, CNN architecture, and training algorithm choice.

3 Data generation

Deep Neural Networks (DNN) require vast amounts of data for their training, on the order of tens or hundreds of thousands of labelled examples. The goal of this work is to enable the use of satellite imagery to estimate bathymetry; we therefore require both satellite images and the corresponding bathymetric measurements of the imaged areas. Using real satellite imagery and bathymetry surveys requires handling uncertainty in the ground truth which might obfuscate fundamental behaviour of the deep learning models. In order to rapidly test and have a clear indicator of the feasibility and performance of our different approaches, we generate an exhaustive synthetic dataset to train and test our deep learning models as a first step (proof-of-concept), with the eventual goal of creating a supervised dataset from real data to which we can apply our models. We also use this synthetic dataset to perform hyperparameter tuning such as architecture choice which we present in this article and which will inform the application to real satellite images.

The process of synthetic data generation is split into two main steps: creating random synthetic bathymetries and then applying a wave model to those bathymetries in order to observe and extract the resulting wave patterns. In this section, we first describe our method of generating synthetic bathymetry profiles, then detail the process of creating synthetic satellite images that replicate those captured by the Sentinel-2 satellite constellation. Finally, we describe our process of transforming the simulated data into training, validation and test sets.

3.1 Bathymetry generation

We simulate coastal areas of 4 km^2 at a resolution of 10 m. It is presumed that an area of 4 km^2 is large enough to realistically incorporate multiple morphological features within a single profile, covering the different possible features that can occur naturally in the real world. To provide a more reliable indicator of the networks' ability to perform bathymetry reconstruction, we raise the complexity of the problem by increasing the level of randomness in our synthetic bathymetry profiles.

We use a Python script to generate random bathymetries by assigning a random slope from a predefined range such that the resulting maximum depth (at 2 km's offshore) falls between 40 and 100 m. We then use the chosen slope to create a flat 2D surface, to which random bumps, sandbars and canyons with random physical dimensions are applied according to a uniform distribution. The value ranges of these physical features are detailed in Table 1. A variety of natural features or man-made artifacts are not simulated in this experimental setup, including islands, lagoons, bays, jettys and wave breakers.

Figure 1 shows two examples of bathymetries, P1 and P2, which demonstrate a diversity of physical features and which will be used throughout this article to demonstrate our methods. For training, we normalize all depth values to range between 0 and 10, by dividing them by 10. This decision is based on an experiment on using different normalization schemes which is discussed in "Appendix B".

3.2 Wave simulations

The generated bathymetries are used as input to FUNWAVE-TVD (Shi et al. 2016), a fully non-linear Boussinesq wave model, which is initialized using random wave conditions for each profile. The three main random parameters that are altered are the significant wave

Table 1 Synthetic bathymetric feature value ranges

Feature	Min.	Max.
Count	2	30
Amplitude	– 20 m	20 m
Cross-shore wavelength	50 m	750 m
Alongshore wavelength	100 m	1500 m

These ranges are used to apply random modifications to the raw bathymetry slope according to a uniform distribution. *Count corresponds to the number of features to be created **A negative amplitude corresponds to a canyon

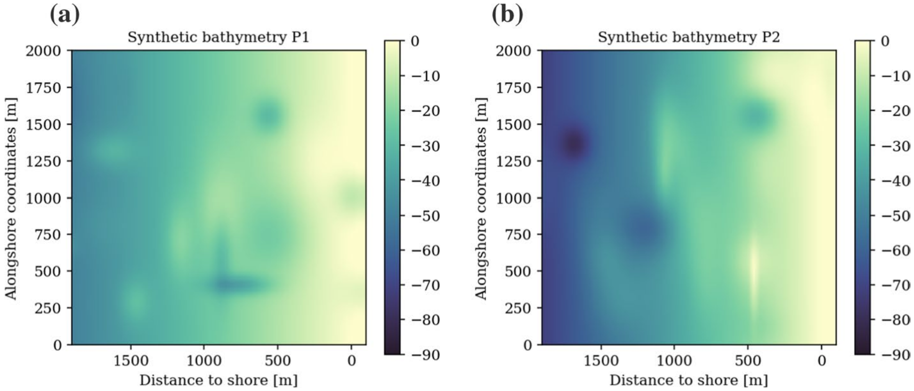

Fig. 1 Synthetic bathymetry profiles P1 and P2 generated using our previously-described bathymetry generation method. A number of Gaussian blobs with random positions, heights, and standard deviations are applied to each bathymetry profile representing bumps, sandbars or canyons in the synthetic seafloor. *The color-scale indicates depth

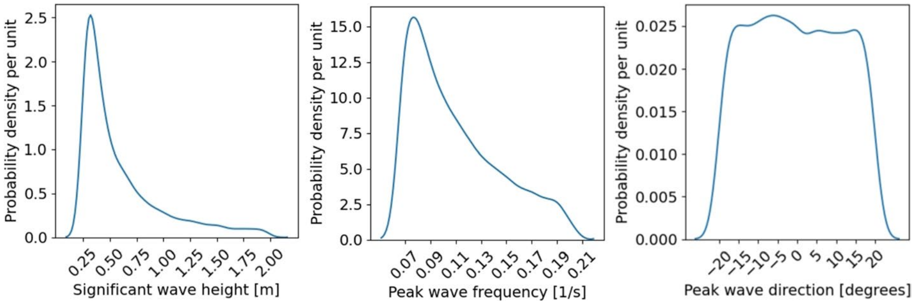

Fig. 2 Distribution and value range of each of the three main wave simulation parameters. (Left) Significant wave height [m]; (Middle) Peak wave frequency [1/s]; (Right) Peak wave direction [degrees]

height, the peak wave period and the peak direction. Figure 2 shows the range and distribution of values for each of these simulated parameters.

The simulated wave pattern images are extracted at different points of the simulation to create image-bursts (stacks), each resembling a single Sentinel-2 image composed of 4 bands with 10 m resolution, taking into account the time lag observed between the different Sentinel-2 bands (Drusch et al. 2012).

The output of the wave simulator is a stack of wave elevation matrices, each representing a single simulation step, or a single band in our synthetic satellite images. The wave simulator we use does not account for sun angles, wind speeds, white caps, seafloor reflection or other factors that can affect our ability to observe wave patterns in real Sentinel-2 imagery. However, we mitigate this difference by further processing our synthetic images, as described in the next subsection.

3.3 Image and dataset generation

Real satellite imagery are composed of pixel intensity values representing the reflection of photons on the surface of the earth. On the ocean, this translates to the gradient of the ocean surface, rather than the surface itself. As discussed in Sect. 3.2, the output of the wave simulator is a stack of wave elevation matrices representing the ocean surface at each time step. Therefore, in order to maximize the similarity between the simulated and real-world satellite images, the wave elevation values of a single band are transformed to a two dimensional gradient map (ΔM) using Eq. 1, where M is the 2D map of wave elevation, δx represents a matrix of slopes along the cross-shore axis, and δy represents the alongshore slopes.

Given that we model a range of possible wave conditions rather than a single specific real-world case with a single condition, a direct comparison between the post-processed synthetic images and real satellite images is difficult. For the purpose of this study, we presume that the use of the 2D gradient map allows us to obtain a surface reflectance matrix that is similar to the real products offered by Sentinel-2 after applying a pass-band filter in the range of ocean-specific wavelengths (periods of 5 s–25 s).

$$\Delta M = \sqrt{\delta x^2 + \delta y^2} \tag{1}$$

Each synthetic surface reflectance band is then normalized to a range of 0–1 using minmax normalization, as shown in Fig. 3.

For this work, we generated 1500 different random bathymetries of size 4 km^2 (200*200 px). For each bathymetry, we created 16 randomly initialized wave simulations, each lasting for approximately 15 min of simulation. Presuming that 16 randomly-initialized simulations include enough variation in wave parameters to be representative of the different real-world conditions over each synthetic bathymetry profile, our choice of the number of simulations and the duration of each, was made purely because it suited our computational setup and allowed us to run a large number of simulations in a short period of time. We take a snapshot every 20 s, where a single snapshot captures 4 bands. In total, this results in roughly 1.2 million 200*200*4 px images. The final processing step differs for our two methods, as we describe in detail in the following sections. For our first method, Deep Large-Scale Reconstruction of Bathymetry, each complete 200*200*4 px image is treated as a single input to our network and its corresponding bathymetry as the output. For our second method, Deep Single-Point Estimation of Bathymetry, we reduce the complete area and include offshore samples only, falling between the wave breaking point and 2 km's offshore. We use these areas to perform sub-sampling over sliding windows of size 40*40*4 px to extract the input images, with the average corresponding depth values as outputs. This is done due to the fact that FUNWAVE-TVD does not simulate wave breaking in its output. Although this enlarges the difference between our synthetic representation and that

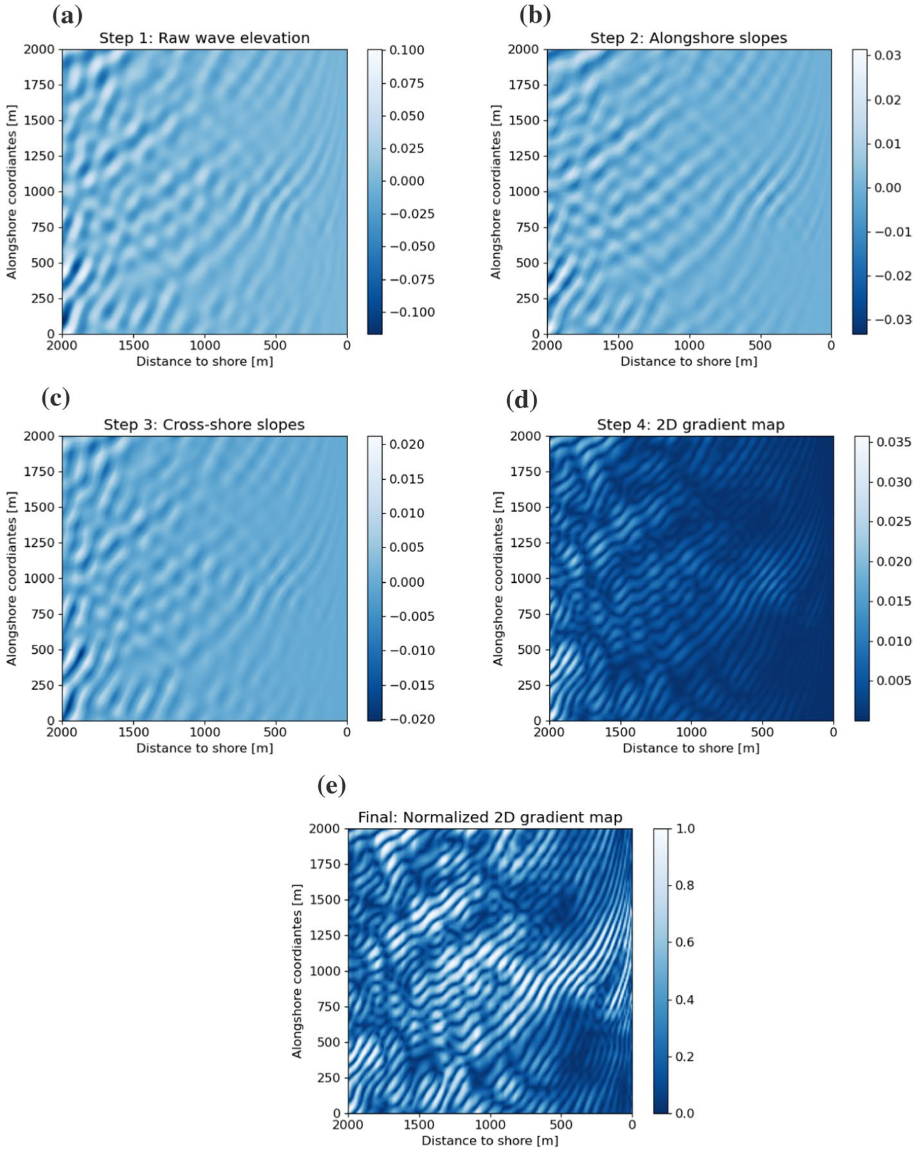

Fig. 3 Pre-processing steps of a single 4 km^2 synthetic satellite band, going from a raw wave elevation matrix (**a**) the alongshore gradient (**b**), the cross-shore gradient (**c**), the 2D gradient map (**d**), and finally the normalized 2D gradient map (**e**) that is used as input to our deep learning models

which can be observed in real satellite imagery, it is sufficient for the purpose of the synthetic dataset.

Our training and validation datasets originate from the same large dataset of 1.2 million full-sized snapshots. These samples are randomly shuffled and split into training and validation sets before being passed to the model in DLSRB. The same is done for DSPEB, subtiles are extracted from the original full-sized snapshots, and are then shuffled and split into training and validation sets. To create our test set, which we use to evaluate our methods throughout this paper, we reran the same data generation scripts to generate 501 different full-sized bathymetry profiles, with 16 wave simulations each. However, we extract only a single snapshot (of 4 bands) from each simulation. This is to further ensure the variety in our test set. Our test set consists of 8016 full-sized (200*200*4 px) snapshots as inputs and their corresponding bathymetry profiles as outputs. In the next sections, we describe the two proposed methods and their performance using these datasets.

4 Deep large-scale reconstruction of bathymetry

Our first method, Deep Large-Scale Reconstruction of Bathymetry(DLSRB), consists of reconstructing complete 4 km^2 bathymetry profiles with a resolution of 10 m, using satellite images of the same size (4 km^2 and 10 m resolution) as input. The main benefit of this approach is the full reconstruction of a bathymetry in one processing step, minimizing the computational cost when applied on a large-scale to reconstruct large coastal bathymetries. This approach also permits the neural network to learn correlations between physically distant wave patterns, which could lead to more accurate bathymetry predictions. In this section, we describe the convolutional network architecture we used then detail our training method and configuration.

4.1 Architecture

For this method, we use an encoder-decoder architecture which allows a neural network to learn a mapping function between inputs and outputs of the same size. This is done by imposing smaller intermediate layers where the information from input and preceding layers are compressed into a dense representation before being mapped back to their original shape. This technique has been used to great success in a wide variety of applications such as image reconstruction, pixel-wise regression, and image segmentation (Yao et al. 2018; Li et al. 2018; Çiçek et al. 2016). In our case, the encoder-decoder architecture forces the neural network to learn the physical relation between the evolution of the wave patterns over time, i.e. the different bands of the input image, and the corresponding underlying water depth in the output data.

The neural architecture used in DLSRB is U-Net, a popular encoder-decoder architecture (Ronneberger et al. 2015). In this work, we make a slight modification to the U-Net architecture, as the original U-Net architecture was designed to work with images of size $572 * 572 * 3$ px images. We created a modified architecture that conforms with our dataset of $200 * 200 * 4$ px images by decreasing the network's input dimensions and moving from crop-and-copy to copy-only connections, as shown in Fig. 4.

4.2 Training

Due to the nature of our goal and our target values, the way we evaluate an estimated bathymetry profile during training can have a substantial effect on the network's ability to learn and achieve our desired goal. We evaluated the effect of different loss functions on network training and present those results in "Appendix A". The conclusion of these experiments is that training with a Mean Squared Logarithmic Error (MSLE) achieves the best overall estimation, but a loss function based on the maximum error in a training batch

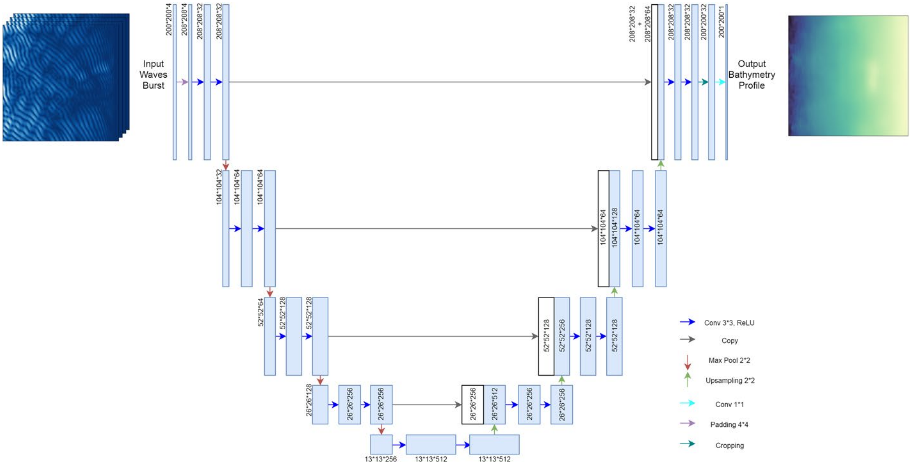

Fig. 4 Deep Large-Scale Reconstruction of Bathymetry using a deep convolutional encoder-decoder network. The neural architecture used in the DLSRB part of the work is a modified version of U-Net (Ronneberger et al. 2015)

may be useful for reducing error in estimating local bathymetry features such as canyons, bumps and sandbars.

The dataset used to train and validate this model consisted of 80k and 20k randomly selected full-sized (200*200*4 px) examples, respectively. To test our model, we reconstructed our complete test set of 8016 full-sized snapshots, as described in Sect. 3. The final model was trained using stochastic gradient descent (SGD) with a learning rate of 0.01 without momentum, and with MSLE as the loss function. It is worth noting that we tested training using the Adam optimizer (Kingma and Ba 2015) however it did not seem to perform well without further tuning. Figure 5 demonstrates that the model trained using SGD reached well-performing results within ≃ 20 epochs. Training was stopped after 30 epochs since no further major improvements in performance on the validation set were

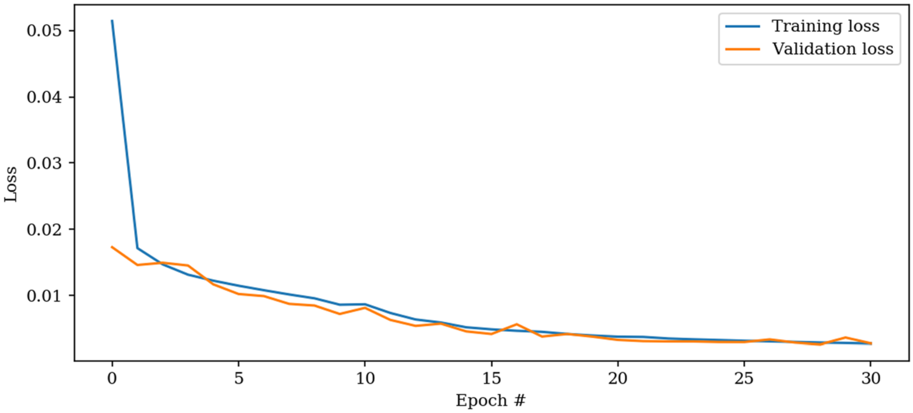

Fig. 5 Training of the DLSRB deep learning model, showing the MSLE training and validation losses over 30 epochs

observed. After training, the model reached a test RMSE of 2.96 m over values up-to 70 m of depth.

5 Deep single-point estimation of bathymetry

In our second method, Deep Single-Point Estimation of Bathymetry (DSPEB), we train a neural network to estimate the average local depth beneath a small sub-tile (0.16 km^2) of a satellite image, instead of reconstructing the complete 4 km^2 area at once as in DLSRB. The advantage of this method is that it can be applied to a wider range of areas and trained using real bathymetric measurements where a large area of measurements is not available. While we compare the two methods in this work on the same simulated profiles, the DSPEB method has a greater range of possible application than the DLSRB method.

5.1 Architecture

The neural network architecture used in this approach is ResNet56, a popular contemporary deep learning architecture (He et al. 2015). This network uses shortcut connections between network layers, allowing for individual functions to be learned by specific network subsections, and resulting in a fewer total trainable parameters when compared with other deep neural networks. In our method, the output of the network is a single value corresponding to the estimated average depth of the area depicted in the input image subtile. We therefore use a single neuron with a rectified linear unit (ReLU) activation as the final layer of the architecture. We evaluated other architectures as detailed in "Appendix C", including very small networks which improve on ResNet56 in computational time, but found that ResNet56 performed best in terms of error and therefore chose it for this method. In the application of this method, we believe that the smaller architectures could be used when faster estimation is beneficial.

The input to this method is a $40 * 40 * 4$ px image representing an area of 0.16 km^2. This is created by sampling over sliding windows of the full-sized satellite images (Fig. 6). The average depth of each window is used as the target value for estimation. As mentioned in 3.3, we only use off-shore subtiles, before the wave breaking point, due to the difference in image representation of wave breaking between the simulation and real data, which is the eventual goal for this model. We also randomly rotate our input images during training, in order to represent a wide range of possible wave propagation directions.

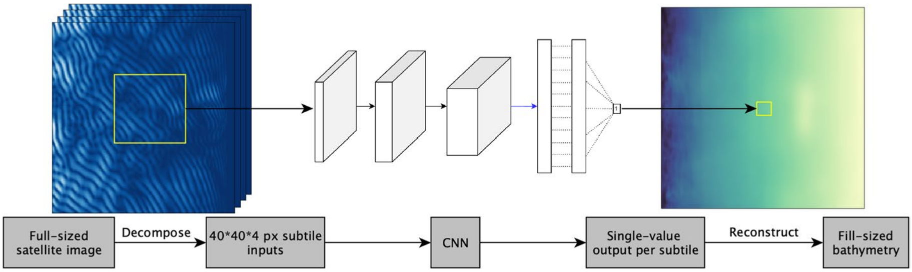
Full-sized satellite image
Decompose
40*40*4 px subtile inputs
CNN
Single-value output per subtile
Reconstruct
Fill-sized bathymetry

Fig. 6 Deep Single-Point Estimation of Bathymetry using a deep convolutional neural network. The neural architecture used in the DSPEB part of the work is a modified version of ResNet56 (He et al. 2015)

5.2 Training

The dataset we use to train and validate this model consists of 75k and 15k randomly selected subtiles (40*40*4 px), respectively. To test our model, we use a sliding window technique to reconstruct the complete test set of 8016 full-sized snapshots. We train our DSPEB model using the Adam optimizer (Kingma and Ba 2015), with mean squared error (MSE) loss and a batch size of 128. We performed a grid search to optimize Adam's hyperparameters; the best training values found were $\epsilon = 1 \times 10^{-8}$, $\beta_1 = 0.99$, and $\beta_2 = 0.999$. A learning rate of 1×10^{-3} was used and was decreased to 1×10^{-4} at 40 epochs. Training was stopped after 50 epochs as there were no substantial improvements on the validation set, as seen in Fig. 7. During training, we keep record of the best-performing model on the validation set. After training, our early checkpoint model with the minimum validation loss (epoch 44) achieved a test RMSE of 3.06 m over values up-to 70 m of depth.

These two methods, DLSRB and DSPEB, present different benefits. We emphasize the applicability advantage of DSPEB since it can be used in a variety of cases which the DLSRB approach can not, due to the small area over which the DSPEB method operates.

6 Results

In this section, we present the resulting bathymetry estimations obtained using our proposed deep learning methods DLSRB and DSPEB. We compare the two approaches based on overall accuracy and performance in Sect. 6.1, we then compare their computational performance in Sect. 6.2.

6.1 Estimation comparison

To compare our deep learning methods DLSRB and DSPEB, we created a test set of 8016 example synthetic cases to be used as our base of reference. The quality of an estimation is based on its point-wise error over the full reconstructed bathymetry grid (2D RMSE), as well as the RMSE of the average cross-shore profile of the estimation compared to the

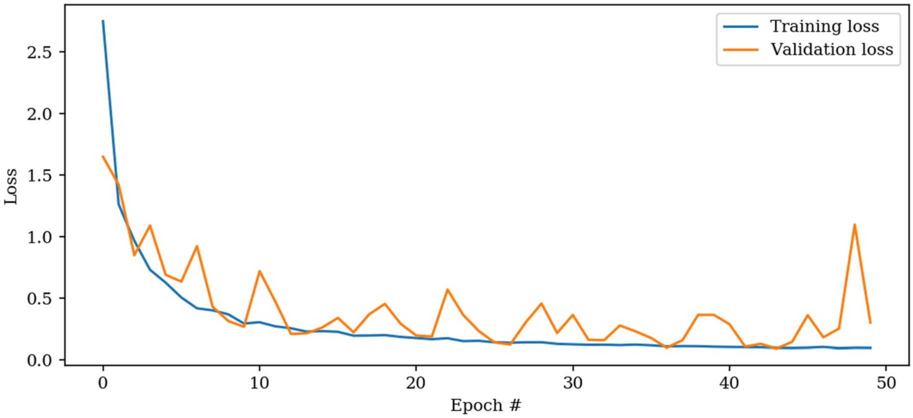

Fig. 7 Training of the DSPEB deep learning model, showing the MSE training and losses over 50 epochs

Table 2 Performance of the two methods on 8016 full-size synthetic test cases

Method	μ 2D RMSE	σ 2D RMSE	μ 1D RMSE	σ 1D RMSE
DLSRB	2.96	1.05	1.78	0.95
DSPEB	3.06	0.62	2.06	0.46

Error is represented in meters

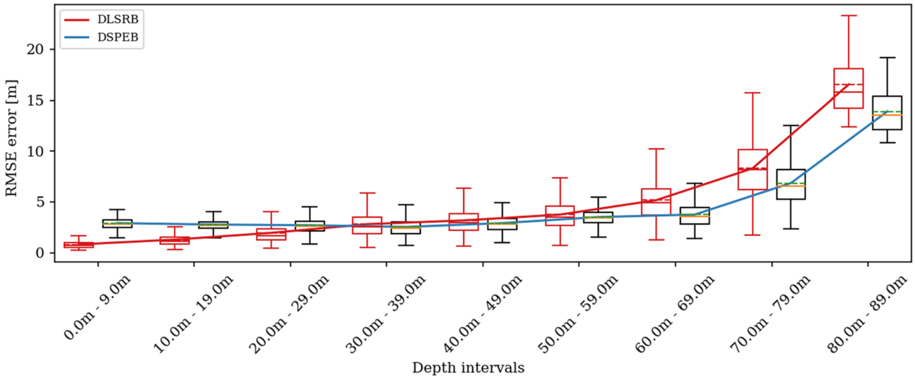

Fig. 8 RMSE at 10 m depth intervals over the full synthetic test set. Both methods demonstrate highly accurate estimation at shallow depths but start to degrade in accuracy in deeper areas

average cross-shore profile of the target bathymetry (1D RMSE). The cross-shore profile is the most common form of bathymetric information, so it is important to consider the cross-shore profile accuracy even though both methods estimate the full 2D bathymetry. Example reconstruction results from the test set are shown in Figs. 9 and 10. We compare the two methods over all test profiles in Table 2. The error of each method at different depth intervals of 10 m over the full test set is then compared in Fig. 8.

In the DSPEB method, we note a consistent decrease in performance near the shoreline compared to DLSRB. This is potentially due to the smaller number of samples used during training at shallow depths. As mentioned in Sect. 3.3, DLSRB was trained to estimate entire bathymetry profiles, but the DSPEB network was trained only on images past the wave breaking point. The under-representation of shallow depths in the training set appears to have resulted in slightly decreased estimation performance near-shore in the DSPEB method. This behaviour can be observed in the example reconstructed bathymetries in Figs. 9 and 10 where DSPEB consistently overestimates the depth in shallow waters. Figure 8 also shows that DLSRB is more accurate than DSPEB in shallow depths. However, Table 2 and Fig. 8 show that DSPEB is more reliable overall, since it has a lower error standard deviation. Furthermore, DSPEB is more consistent down to 70 m, while DLSRB's accuracy starts deteriorating more noticeably after 50 m depth. Finally, as mentioned in Sect. 5, a significant advantage of DSPEB over DLSRB is its applicability to a wider range of cases where DLSRB could be unstable, such as areas with islands or having irregular shorelines.

Both methods demonstrate a high estimation accuracy on synthetic data, which can be competitive with state-of-art methods when applied to real data. In physical inversion models based on wave celerity using satellite imagery such as (Poupardin et al. 2016; Almar et al. 2019), RMSE on real data falls within the same range of error as our methods'

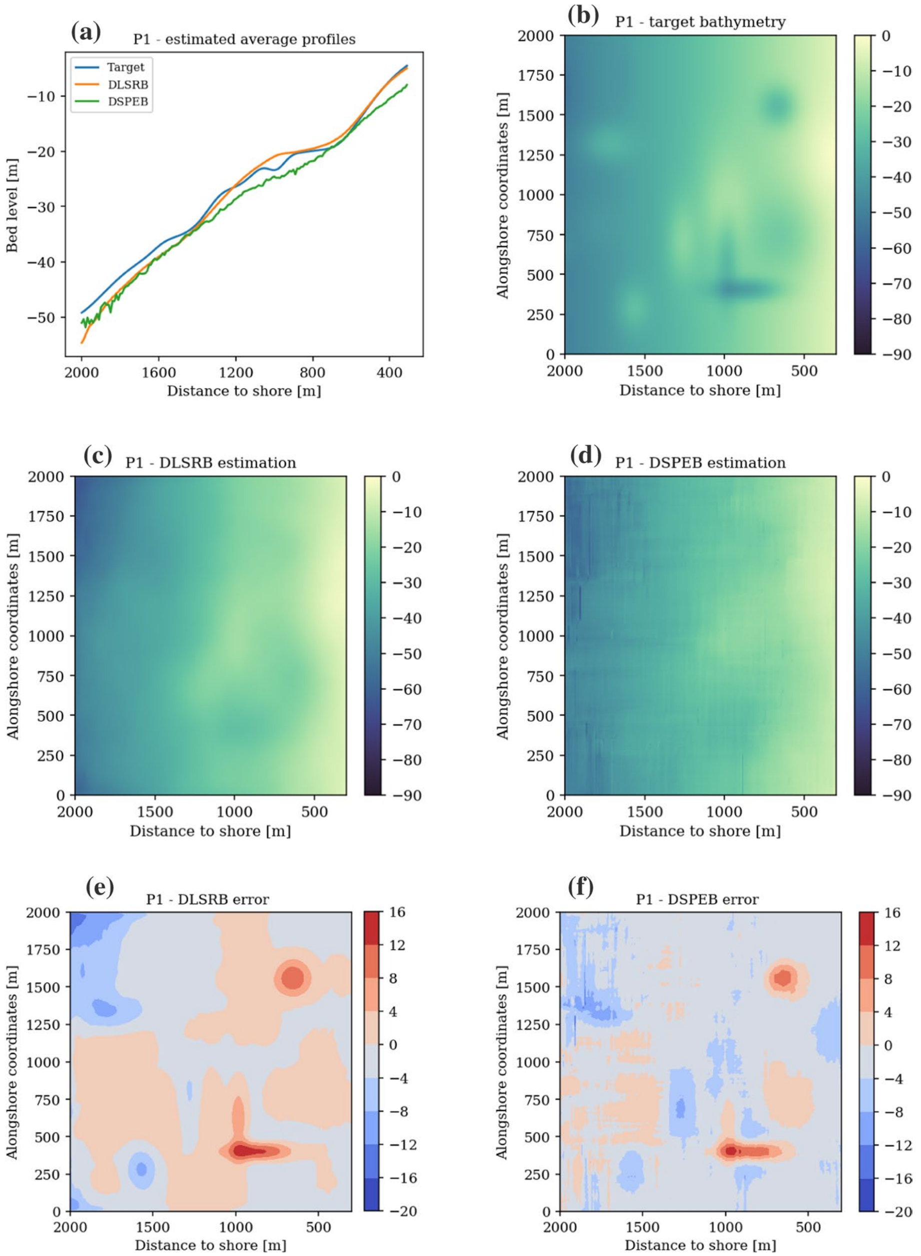

Fig. 9 Reconstruction of bathymetry profile P1. **a** the average profiles of the target and estimated bathymetries; **b** the target bathymetry P1 in 2D; **c**, **d** the reconstructed bathymetry using DLSRB and DSPEB respectively; **e**, **f** the reconstructed bathymetry errors of each method. Cross-shore RMSE: DLSRB = 1.6 m, DSPEB = 2.3 m. 2D RMSE: DLSRB = 3.0 m, DSPEB = 3.3 m

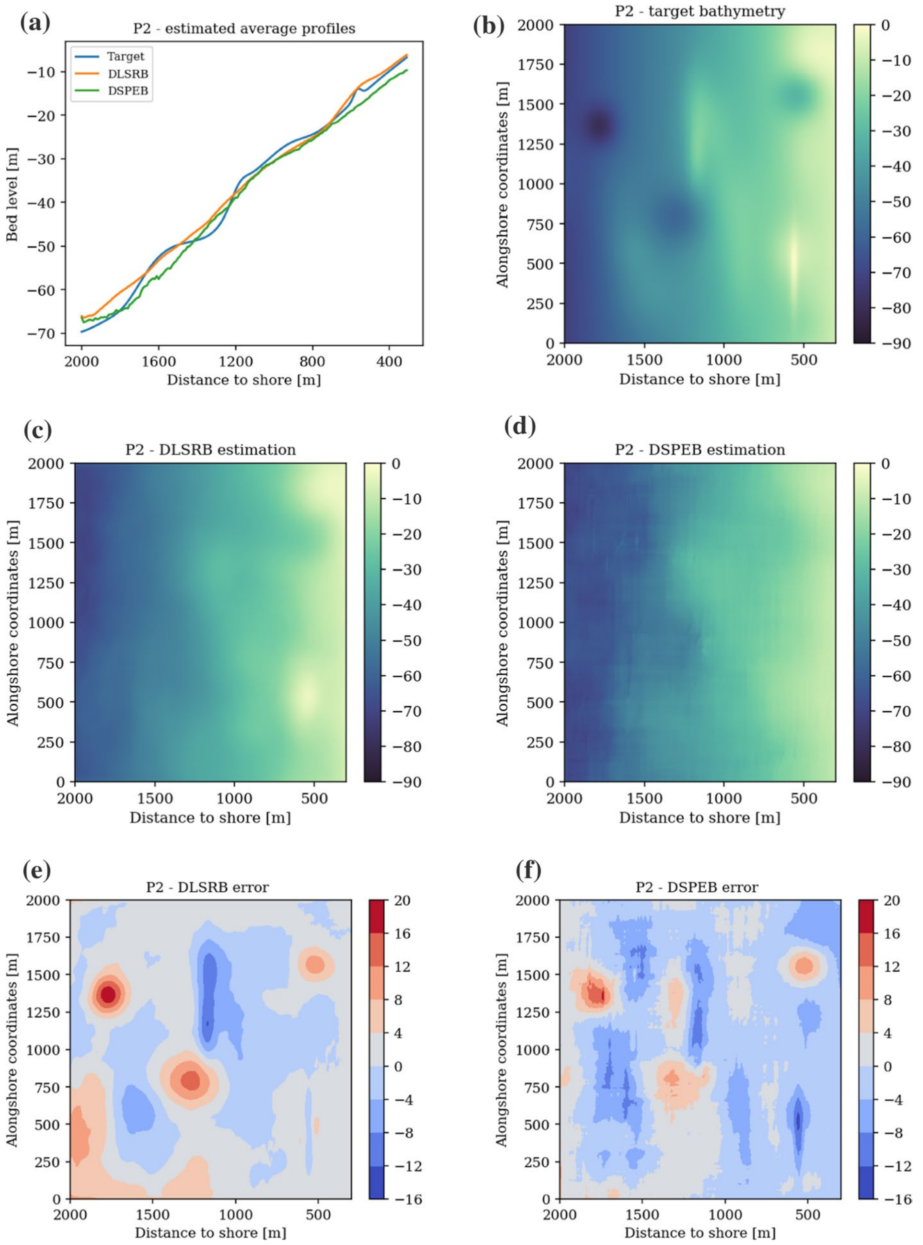

Fig. 10 Reconstruction of bathymetry profile P2. **a** the average profiles of the target and estimated bathymetries; **b**the target bathymetry P2 in 2D; **c**, **d** the reconstructed bathymetry using DLSRB and DSPEB respectively; **e**, **f** the reconstructed bathymetry errors of each method. Cross-shore RMSE: DLSRB = 2.0 m, DSPEB = 2.2 m. 2D RMSE: DLSRB = 3.6 m, DSPEB = 3.6 m

average error range for depths down to 70 m. The preliminary results in this work demonstrate the true potential of deep learning-based methods as alternative methods for performing bathymetry estimation from satellite imagery, especially considering the target depth range they are able to estimate in these synthetic cases. A complete comparison to state-of-the-art methods on satellite data is ongoing, but these first results show the promising value of two different deep learning approaches.

6.2 Computational performance

One of the benefits of deep learning models, apart from their ability to learn complex relations, is that they can be an efficient alternative to perform complicated analytical tasks. While the training of a DNN can be computationally very expensive, once the model is trained, it can be inexpensive to use, especially compared to physics-based methods. We therefore measure the computational time of our proposed methods. Table 3 compares the average time it takes each method to reconstruct a full 4 km^2 bathymetry profile from a satellite image.

DLSRB has a significant advantage over DSPEB in terms of speed, since reconstructing a complete bathymetry profile is done in one forward pass in DLSRB, while DSPEB estimates a single point at a time. In other words, DLSRB performs a single forward pass to produce a 200*200 result matrix, while DSPEB requires 40,000 forward passes to reconstruct a matrix of the same size. This can be optimized by computing these predictions in parallel using a batch, and for this comparison, we optimize DSPEB by treating each cross-shore line as a single batch of input satellite sub-tiles. This test was performed using a single Nvidia GTX 1080 TI GPU on a virtual compute node with 4 CPUs.

Both deep learning methods are competitive in computational time to existing methods, which require computation time on the order of tens of seconds for an estimation of 4 km^2. The DLSRB method, however, is much faster due to the large area it covers, as physical methods also require the estimation of single points. Our future work aims to accurately compare the computational performance in single point estimation between deep learning methods and physics-based methods and to improve the reconstruction time of DSPEB.

7 Conclusion

In this work, we demonstrate a first application of deep learning to satellite derived bathymetry using synthetic cases. We propose two different approaches to the problem of SDB which have different advantages. DLSRB shows a great potential in large-scale reconstruction due to its ability to reconstruct complete 4 km^2 profiles in a very short amount of time. However, due to the difficulty to accurately model the physical characteristics of large coastal areas, we currently consider DLSRB as limited to large areas that have a nearly straight shoreline, and that are clear of man-made or natural artifacts such as ships or islands. DSPEB provides a trade-off between usability and computational efficiency. Since

Table 3 Results comparison using 8016 test cases

Method	Average speed [seconds]
DLSRB	0.03
DSPEB	22.65

it processes a single small sub-tile at a time, it overcomes the limitations of DLSRB which greatly increases its applicability to a large number of coastal areas.

The results presented in this paper showcase deep learning's ability to perform bathymetry inversion using two different techniques that can be adapted to different use cases. Our proposed methods are able to accurately estimate depths down to $\simeq$70 m while maintaining an error rate of $\simeq$3 m. We are actively pursuing further improvements to our methods. This shows that with further work and research, deep learning can become a viable alternative for performing bathymetry inversion from satellite imagery.

For this preliminary application, we used synthetic data meant to replicate images from the Sentinel-2 satellite constellation. During the course of this project, we gained access to a number of real bathymetric measurements that, coupled with Sentinel-2 data, would enable us to create a labelled dataset that is sufficient for training. However, this is non-trivial due to the various complications that can be faced when dealing with real bathymetry data and real satellite imagery and is an ongoing project.

Our next steps in this project include creating a supervised dataset from real data for our DSPEB model, which we will use to experiment with training on real data exclusively, as well as to test the model's performance using transfer learning techniques. We would also like to extend the synthetic data to include different types of shorelines, estuaries, and wave breaking to verify the model's ability in a variety of settings.

Another possible area for improvement is to apply data augmentation to our input data. The technique proposed in Bergsma et al. (2019b) augments the 20 m bands to 10 m, which would give our model access to twice the temporal information it currently has. This could increase the models' ability to give more accurate depth estimations.

Finally, the architecture search performed for DSPEB presented in "Appendix C" is a strong motivation to explore neural architecture search (NAS), which is a method for optimizing neural network structure that has shown great success in many applications (Elsken et al. 2019). In our case, NAS could allow us to find a convolutional architecture able to achieve good estimation accuracy with minimal computational cost for complete profile reconstruction.

Machine learning, specifically deep learning, has recently demonstrated impressive capabilities in image processing and model inversion. In this work, we demonstrate that both can be achieved in the same network; with an appropriate choice of network architecture and loss function, our models were able to perform accurate bathymetry estimation from synthetic satellite images. We believe that this is a promising direction for Satellite-Derived Bathymetry and Earth Observation in general.

Appendices

Appendix A DLSRB loss function

When training a neural network, the loss function determines how estimation error is calculated, which in turn determines how the network parameters are updated. The choice of loss function is an important hyperparameter in DNN training especially for regression problems such as this one.

The loss functions we tested include mean-squared-error (MSE) and mean-squared-logarithmic-error (MSLE). Both of these loss functions give an overall evaluation of the entire bathymetry profile estimated. In addition, we implemented a custom loss function that uses a single-point maximum-error over an entire training batch of examples as the loss value (MAXERROR), which is intended to emphasize a single large local error at a time during training, rather than the complete bathymetry grid at once. This was tested due to the ability of the network with other loss functions to learn the average slope of a bathymetry easily while failing to detect local bathymetric features such as bumps, sandbars and canyons, which cause local inconsistencies in the wave patterns around those features.

We evaluate our results using the two following metrics:

- *2D RMSE* refers to the RMSE of the point-wise errors of the full 2D reconstructed bathymetry grid compared to the original synthetic target.
- *Average profile RMSE* refers to the RMSE of the point-by-point errors between the average cross-shore profiles of the estimation and the original bathymetry.

Table 4 showcases the results of our experiments using different loss functions. We observed that the model was unable to learn using MSE. However, MSLE and MAXERROR both gave good initial results, while varying in the quality due to the nature of each function. MSLE made the model able to accurately predict the general slope of the bathymetry, while MAXERROR enabled it to give more accurate estimations in terms of the local features of the bathymetry. We believe the MAXERROR function could be useful in combination with an MSLE error and will consider this in future work using the DLSRB method.

Appendix B Depth normalization

In deep learning-based regression, scaling the target values can aid in model training due to the initial distribution of neural network parameters which is near 0. As the target depth values are much larger than the range of values possible from a newly initialized neural network, we normalize the target depth values using a linear scale. We divide depth values by two constants, 10 and 100, where we consider 100 to be near the maximum target value possible, and compare on the average cross-shore profile.

In Table 5, we show that using the smaller normalization value of 10 resulted in lower average error for both the full 2D bathymetry and the 1D average cross-shore profile. In future work, we aim to include this hyperparameter as a part of the search described in "Appendix D", as the impact of target value scaling depends on network parameter initialization and optimizer hyperparameters.

Table 4 Loss functions performance comparison

Loss Function	μ 2D RMSE	σ 2D RMSE	μ 1D RMSE	σ 1D RMSE
MSE	21.46	5.35	21.15	5.38
MAXERROR	10.44	2.13	9.79	2.26
MSLE	3.48	1.33	2.27	1.34

Table 5 Comparison of two target normalization schemes, division by 10 and 100, on the final estimation error of the DLSRB network

Bathymetry normalization	μ 2D RMSE	σ 2D RMSE	μ 1D RMSE	σ 1D RMSE
10	3.48	1.33	2.27	1.34
100	5.41	1.86	4.4	2.06

All other training parameters were fixed

Appendix C DSPEB neural network architecture

The DSPEB model is based on convolutional neural network architectures meant for classification. These networks extract features from input images using convolutional layers, then use the extracted features to perform classification or regression using fully connected layers. Regarding the specific architecture used, we tested the ResNet110 v2, ResNet56 v2 and ResNet20 v2 architectures (He et al. 2015) and we replaced the activation function of the output layer with ReLU activation in order to perform regression. We also implemented a small number of custom convolutional networks with different max pooling and binary normalization layer configurations, which we refer to as DSPEB-nets for the purpose of this study, as specified in Table 6.

From our initial set of seven architectures (3 ResNets, 4 DSPEB-nets), we discarded ResNet110 and DSPEB-2 as they were unable to converge, which could be an optimiser hyper-parameter configuration issue. The progression of the validation loss of the remaining 5 architectures over 30 epochs can be seen in Fig. 11, an analysis of their reconstruction performance on a set of 8016 test cases can be seen in Table 7, and a comparison of their average speed of reconstructing complete 4 km^2 bathymetry profiles using a sliding-window technique can be seen in Table 8.

A pattern can be observed in Fig. 11, where DSPEB-nets' validation losses seem to be more stable as they are able to converge within $\simeq$10 epochs, and maintain stable validation loss patterns during fine tuning while the validation losses of ResNet56 and ResNet20 show large variation during training.

Contrary to what can be observed from the learning curves, Table 7 shows that ResNet56 and ResNet20 have the lowest RMSE scores when comparing the complete 2D reconstructions to the label data, in addition to having the lowest error standard deviation. Comparing the average cross-shore profiles, DSPEB-nets perform better than ResNets, however the error standard deviations suggest that ResNets are more stable.

Table 6 Architectures

DSPEB-1	DSPEB-2	DSPEB-3	DSPEB-4
conv3-4	conv3-4	conv3-4	conv3-4
conv3-16	conv3-16	conv3-16	conv3-16
conv3-32	conv3-32-bn	conv3-32	maxpool
–	–	maxpool	conv3-32-bn
FC-256	FC-256	FC-256	FC-256
FC-256	FC-256	FC-256	FC-256
FC-1	FC-1	FC-1	FC-1

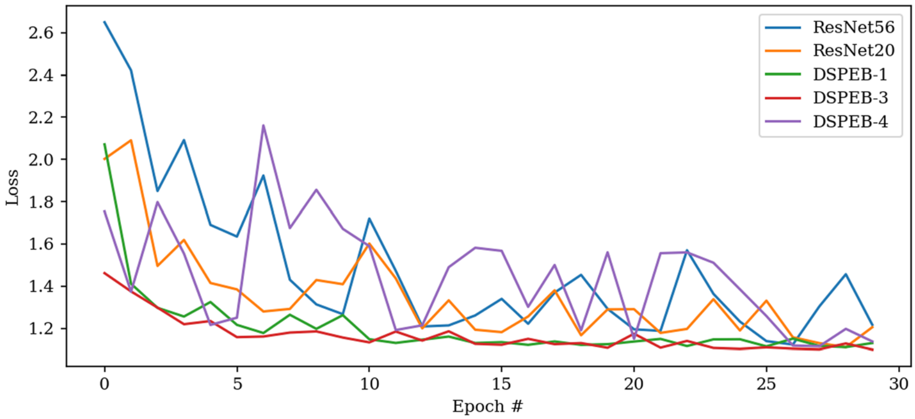

Fig. 11 Training of the DSPEB models, showing the validation MSE over 30 epochs

Table 7 Network performance on a test set of 8016 synthetic cases

Network	avg. 2D RMSE	2D RMSE std.	avg. 1D RMSE	1D RMSE std.
ResNet56	3.06	0.62	2.06	0.46
ResNet20	3.1	0.91	1.97	0.72
DSPEB-1	3.83	1.47	1.93	1.05
DSPEB-3	3.5	1.18	1.84	0.83
DSPEB-4	3.56	0.97	1.88	0.89

All values shown are in meters

Table 8 Average reconstruction speed comparison over 100 synthetic examples

Network	avg. time [seconds]
ResNet56	22.65
ResNet20	13.14
DSPEB-1	8.46
DSPEB-3	8.33
DSPEB-4	7.84

Table 8 shows the average time it takes each model to go through all 0.16 km^2 subtiles of a complete satellite image. In other words, it shows the time it takes each model to perform ≃ 40,000 single-point estimations to reconstruct a 4 km^2 bathymetry profile from a complete satellite image. Tables 7 and 8 suggest that using a simple architecture can be an advantageous choice when applying DSPEB on a large scale. DSPEB-3 for example is able to achieve satisfactory accuracy while having very short reconstruction times relative to more complex architectures such as ResNet56. However, we use ResNet56 as our reference model for DSPEB in this work due to its superior 2D accuracy and low error standard deviation.

Appendix D DSPEB hyperparameter grid-search

Neural network training is controlled by a large number of parameters. This experiment was focused on exploring different Adam hyperparameters in order to optimize the learning process. These parameters were the learning rate (lr), epsilon (ϵ), beta1 (β_1) and beta2 (β_2). The value ranges we tested are as follows:

- lr: $\{1 \times 10^{-3}, 1 \times 10^{-5}\}$
- ϵ: $\{1 \times 10^{-4}, 1 \times 10^{-8}\}$
- β_1: $\{0.5, 0.99\}$
- β_2: $\{0.9, 0.999\}$

For this experiment, a total of 16 models were trained for 50 epochs each, using 20k and 5k examples as the training and validation sets respectively. Figure 12 showcases the validation loss for each of the combinations tested. As mentioned in Sect. 5.2, our final values we used to train our final model were $\epsilon = 1 \times 10^{-8}$, $\beta_1 = 0.99$, and $\beta_2 = 0.999$, with a learning rate of 1×10^{-3} that was decreased to 1×10^{-4} at 40 epochs.

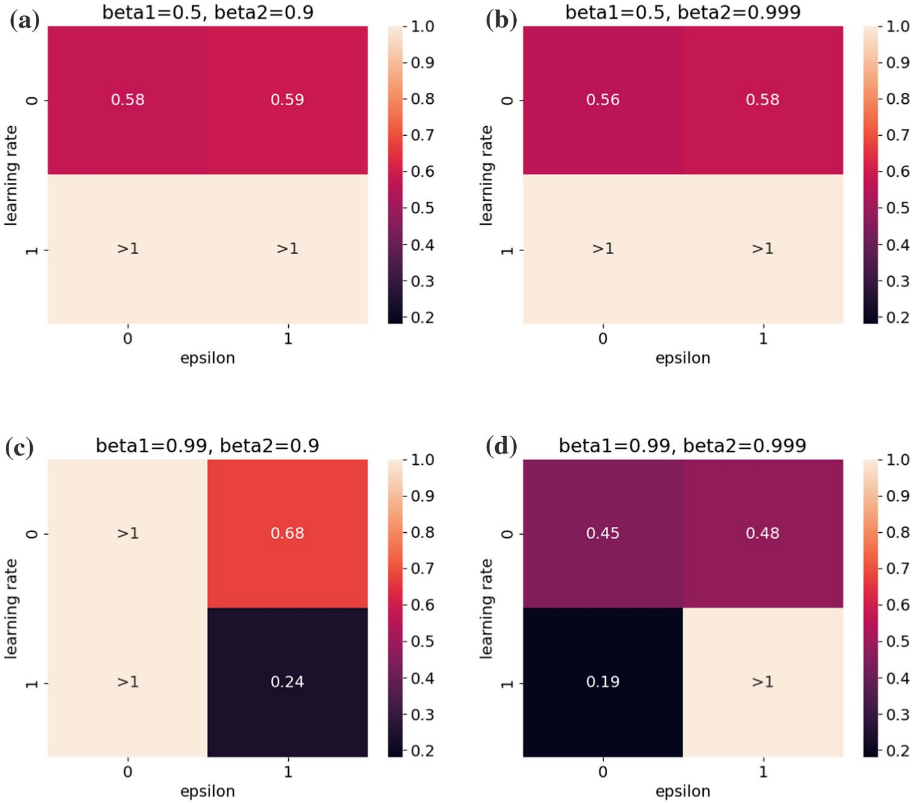

Fig. 12 Hyperparameter grid-search results showing the validation loss using different combinations of Adam's hyperparameters. **a** $\beta_1 = 0.5, \beta_2 = 0.9$; **b**$\beta_1 = 0.5, \beta_2 = 0.999$; **c** $\beta_1 = 0.99, \beta_2 = 0.9$; **d** $\beta_1 = 0.99, \beta_2 = 0.999$

References

Abileah, R. (2006). Mapping shallow water depth from satellite. In: *Proceedings of the ASPRS Annual Conference*, Reno, Nevada pp 1–7

Almar, R., Bonneton, P., Senechal, N., & Roelvink, D. (2009). Wave celerity from video imaging: A new method. In *Proceedings of the 31st international conference coastal engineering* (Vol. 1, pp. 661–673). https://doi.org/10.1142/9789814277426_0056.

Almar, R., Bergsma, E. W. J., Maisongrande, P., & de Almeida, L. P. M. (2019). Wave-derived coastal bathymetry from satellite video imagery: A showcase with pleiades persistent mode. *Remote Sensing of Environment, 231,* 111263. https://doi.org/10.1016/j.rse.2019.111263.

Benveniste, J., Cazenave, A., Vignudelli, S., Fenoglio-Marc, L., Shah, R., Almar, R., et al. (2019). Requirements for a coastal hazards observing system. *Frontiers in Marine Science, 6,* 348. https://doi.org/10.3389/fmars.2019.00348.

Bergsma, E. W. J., & Almar, R. (2018). Video-based depth inversion techniques, a method comparison with synthetic cases. *Coastal Engineering, 138,* 199–209. https://doi.org/10.1016/j.coastaleng.2018.04.025.

Bergsma, E. W. J., & Almar, R. (2020). Coastal coverage of esa' sentinel 2 mission. *Advances in Space Research*. https://doi.org/10.1016/j.asr.2020.03.001.

Bergsma, E. W. J., Almar, R., de Almeida, L. P. M., & Sall, M. (2019a). On the operational use of uavs for video-derived bathymetry. *Coastal Engineering, 152,* 103527. https://doi.org/10.1016/j.coastaleng.2019.103527.

Bergsma, E. W. J., Almar, R., & Maisongrande, P. (2019b). Radon-augmented sentinel- satellite imagery to derive wave-patterns and regional bathymetry. *Remote Sensing, 11,* 1918. https://doi.org/10.3390/rs11161918.

Bird, J.J., Faria, D.R., Ekárt, A., & Ayrosa, P.P. (2020). From simulation to reality: Cnn transfer learning for scene classification. In: *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, IEEE, pp 619–625

Brodie, K. L., Bruder, B. L., Slocum, R. K., & Spore, N. J. (2019). Simultaneous mapping of coastal topography and bathymetry from a lightweight multicamera uas. *IEEE Transactions on Geoscience and Remote Sensing, 57*(9), 6844–6864. https://doi.org/10.1109/TGRS.2019.2909026.

Catalán, P., & Haller, M. C. (2006). Nonlinear phase speeds and depth inversions. In *Coastal Dynamics 2005 - Proceedings of the fifth coastal dynamics international conference*. https://doi.org/10.1061/40855(214)42.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., & Ronneberger, O. (2016). 3d u-net: learning dense volumetric segmentation from sparse annotation. In: *International conference on medical image computing and computer-assisted intervention*, Springer, pp 424–432

Dickens, K., & Armstrong, A. (2019). Application of machine learning in satellite derived bathymetry and coastline detection. SMU Data Science Review: 2, https://scholar.smu.edu/datasciencereview/vol2/iss1/4

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., et al. (2012). Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment, 120,* 25–36. https://doi.org/10.1016/j.rse.2011.11.026.

Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research, 20*(55), 1–21.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, MA, USA: MIT Press.

Ham, Y. G., Kim, J. H., & Luo, J. J. (2019). Deep learning for multi-year ENSO forecasts. *Nature, 573*(7775), 568–572.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. arXiv e-prints p arXiv:1512.03385

Hoeser, T., & Kuenzer, C. (2020). Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. *Remote Sensing, 12*(10), 1667.

Holman, R. A., Plant, N., & Holland, T. (2013). Cbathy: A robust algorithm for estimating nearshore bathymetry. *Journal of Geophysical Research: Oceans, 118,* 2595–2609.

Iglovikov, V., Mushinskiy, S., & Osin, V. (2017). Satellite imagery feature detection using deep convolutional neural network: A kaggle competition. arXiv preprint arXiv:170606169

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv:1412.6980.

Kumar, N. K., Savitha, R., & Al Mamun, A. (2018). Ocean wave height prediction using ensemble of extreme learning machine. *Neurocomputing, 277,* 12–20.

Lee, Z., Carder, K. L., Mobley, C. D., Steward, R. G., & Patch, J. S. (1999). Hyperspectral remote sensing for shallow waters: 2. deriving bottom depths and water properties by optimization. *Applied Optics, 38,* 3831–3843.

Li, X., Chen, H., Qi, X., Dou, Q., Fu, C. W., & Heng, P. A. (2018). H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. *IEEE Transactions on Medical Imaging, 37*(12), 2663–2674.

Liu, Q., Hang, R., Song, H., & Li, Z. (2017). Learning multiscale deep features for high-resolution satellite image scene classification. *IEEE Transactions on Geoscience and Remote Sensing, 56*(1), 117–126.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing, 152,* 166–177.

Melet, A., Teatini, P., Le Cozannet, G., Jamet, C., Conversi, A., Benveniste, J., & Almar, R. (2020). Earth observations for monitoring marine coastal hazards and their drivers. *Surveys in Geophysics*. https://doi.org/10.1007/s10712-020-09594-5.

de Michele, M., Leprince, S., Thiebot, J., Raucoules, D., & Binet, R. (2012). Direct measurement of ocean waves velocity field from a single spot-5 dataset. *Remote Sensing of Environment, 119,* 266–271.

Plant, N. G., Holland, K. T., & Haller, M. C. (2008). Ocean wavenumber estimation from wave-resolving time series imagery. *IEEE Transactions on Geosciences and Remote Sensing, 46,* 2644–2658.

Poupardin, A., de Michele, M., Raucoules, D., & Idier, D. (2014). Water depth inversion from satellite dataset. In: *2014 IEEE Geoscience and Remote Sensing Symposium*, pp 2277–2280

Poupardin, A., Idier, D., de Michele, M., & Raucoules, D. (2016). Water depth inversion from a single spot-5 dataset. *IEEE Transactions on Geoscience and Remote Sensing, 119*(4), 2329–2342.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, pp 234–241

Sagawa, T., Yamashita, Y., Okumura, T., & Yamanokuchi, T. (2019). Satellite derived bathymetry using machine learning and multi-temporal satellite images. *Remote Sensing, 11,* 1155. https://doi.org/10.3390/rs11101155.

Salameh, E., Frappart, F., Almar, R., Baptista, P., Heygster, G., Lubac, B., et al. (2019). Monitoring beach topography and nearshore bathymetry using spaceborne remote sensing: A review. *Remote Sensing*. https://doi.org/10.3390/rs11192212.

Sandidge, J. C., & Holyer, R. J. (1998). Coastal bathymetry from hyperspectral observations of water radiance. *Remote Sensing of Environment, 65*(3), 341–352.

Savitha, R., Al Mamun, A., et al. (2017). Regional ocean wave height prediction using sequential learning neural networks. *Ocean Engineering, 129,* 605–612.

Shi, F., Kirby, J., Tehranirad, B., Harris, J., Choi, Y.K., & Malej, M. (2016) Funwave-tvd fully nonlinear boussinesq wave model with tvd solver - documentation and user's manual (version 3.0)

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Stockdon, H. F., & Holman, R. A. (2000). Estimation of wave phase speed and nearshore bathymetry from video imagery. *Journal of Geophysical Research, 105,* 22015–22033.

Vojinovic, Z., Abebe, Y., Ranasinghe, R., Vacher, A., Martens, P., Mandl, D., et al. (2013). A machine learning approach for estimation of shallow water depths from optical satellite images and sonar measurements. *Journal of Hydroinformatics, 15*(4), 1408–1424.

Wang, L., Scott, K. A., Xu, L., & Clausi, D. A. (2016). Sea ice concentration estimation during melt from dual-pol sar scenes using deep convolutional neural networks: A case study. *IEEE Transactions on Geoscience and Remote Sensing, 54*(8), 4524–4533.

Yao, W., Zeng, Z., Lian, C., & Tang, H. (2018). Pixel-wise regression using u-net and its application on pansharpening. *Neurocomputing, 312,* 364–371. https://doi.org/10.1016/j.neucom.2018.05.103.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine, 5*(4), 8–36.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., et al. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109,* 43–76.

Authors and Affiliations

Mahmoud Al Najar[1] · **Grégoire Thoumyre**[2] · **Erwin W. J. Bergsma**[3] · **Rafael Almar**[2] · **Rachid Benshila**[1] · **Dennis G. Wilson**[4]

Grégoire Thoumyre
gregoire.thoumyre@legos.obs-mip.fr

Erwin W. J. Bergsma
erwin.bergsma@cnes.fr

Rafael Almar
rafael.almar@ird.fr

Rachid Benshila
rachid.benshila@legos.obs-mip.fr

Dennis G. Wilson
dennis.wilson@isae-supaero.fr

[1] CNRS, LEGOS, Toulouse, France

[2] IRD, LEGOS, Toulouse, France

[3] CNES, LEGOS, Toulouse, France

[4] ISAE-SUPAERO, Toulouse, France