Coastline extraction from ALOS-2 satellite SAR images

Petr Hurtik and Marek Vajgl

Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, Ostrava, Czech Republic

ABSTRACT
The continuous monitoring of a shore plays an essential role in designing strategies for shore protection against erosion. To avoid the effect of clouds and sunlight, satellite-based imagery with synthetic aperture radar is used to provide the required data. In contrast to standard model-driven methods, we present a deep-learning-based approach to detect coastlines in such data. We split the process into data preprocessing, model training, inference, ensembling, and postprocessing and describe the best techniques for each of the parts. To deal with a small training dataset, we propose a novel multi-sample mosaicing augmentation that helps the deep neural network models to reduce overfitting during training. Our solution has been validated against the real Global Positioning System location of coastlines during a worldwide competition organized by Signate and Japan Aerospace Exploration Agency, where it was runner-up among 109 teams from the whole world.

1. Introduction and problem formulation

Japan is a country surrounded by oceans and a large length of the seaside; thus, it is very sensitive to its changes, as every degradation is irreversible and affects the future. Because the coast is under the continuous influence of water, it is affected by erosion. As mentioned in (Mori et al. 2018), the erosion is accelerated by climate change, but also due to rapid national development. Such changes significantly affect the nearby population and property. It is necessary to know where the store starts degrading to prevent coastal erosion. To avoid artefacts caused by clouds and sunlight, satellite-based synthetic aperture radar (SAR) is used for capturing data. However, the provided SAR data quality is affected by selecting proper polarization, band, incident angle, and their processing demands using more advanced methods. A solution to the problem connects several scientific areas. Knowledge from the electronic engineering area is used to construct a spaceship and satellite itself. Machine vision area contributes to synthetic aperture radar, and on the basis of machine intelligence, systems controlling the image capturing (angle of incidence, swath, etc.) are designed. Finally, machine learning in image processing helps to interpret the captured data.
Figure 1. The pipeline for coastline extraction, split into the training and prediction part.

To tackle the problem of computer vision for SAR imaginary and to exhaustively benchmark the real state-of-the-art (SOTA) approaches in the area, the detection of coastlines was the subject of an open, worldwide competition organized by the Signate company. In this paper, we describe our pipeline and the ideas used in the competition (signate.jp/competitions/284), where our solution achieved 2\textsuperscript{nd} place. The competition was sponsored by the Japanese Ministry of Economy, Trade, and Industry in cooperation with Japan Aerospace Exploration Agency. In total, 803 people registered for the competition, 147 of them submitted at least one solution. Every such submission was immediately evaluated on a partial test set to provide a preliminary ranking during the competition. Every person/team selected one final submission from those submissions, which (after the competition deadline) provided the final ranking for the entire test set. In total, 2783 solutions were submitted. The competition lasted for three months, ending in November 2020.

The evaluation computed the average Euclidean distance between the ground truth and the predicted coastline at predefined (unknown for the test set) evaluation points. If the algorithm missed the prediction, it was penalized. Our solution achieved the final score of 11.23; the first place achieved 11.18, and the third place reached 11.76. For a detailed explanation of how the score was computed, see signate.jp/competitions/284#evaluation.

Our solution includes existing augmentations as well as one novel, efficient architecture with modern SOTA backbones and powerful postprocessing. As our solution is based on a data-driven approach, it can be applied to other remote sensing problems where image segmentation is needed. All source codes are freely available online (gitlab.com/irafm-ai/signate_4th_tellus_competition). The illustrative scheme of our pipeline is shown in Figure 1. Our solution firstly preprocesses the data, augments them, and trains models using the data. The trained models are then taken and used to make predictions on a test image. The predictions are ensembled and post-processed to fix minor issues. A detailed description is the subject of this study.

The major contribution and the paper’s goal is to create a ‘cookbook’ of good practices and solutions for challenges in remote sensing, image segmentation, and coastline extraction. The proposed good practices utilize the best techniques known in deep neural networks. The novelty is in the multi-sample mosaicing augmentation, which increases the model’s generalization ability and leads to higher accuracy.
2. Current state

2.1. Model-driven approach

These methods usually involve methods known from standard image processing, such as kernel-based filtering, morphology, or gradient operators. Such a typical example is (Acar et al. 2012) that filters out the image by histogram equalization and mathematical morphology until Sobel’s operator can extract coastline. The process is improved by iterative gap closing and flood filling in (Wiehle and Lehner 2015). In (Mirsane et al. 2018), authors use a wavelet algorithm followed by iso-data extraction, filled by a watershed algorithm, and take the coastline as a border from the largest segment in the image. The obtained shape is smoothened by mathematical morphology. Study (Ferrentino et al. 2020) proposes a model-driven pipeline consisting of noise filtering, morphological filtering, and edge detection. It shows that noise filtering plays an important role when such a model-driven approach is used and affects the results significantly. A similar pipeline is also used in (Modava and Akbarizadeh 2017) where the active contour method is used instead of a ‘simple’ edge detection technique. The same authors use in (Modava, Akbarizadeh, and Soroosh 2019) pipeline with noise filtering as well but extract the coastline based on spectral-texture segmentation. As a segmentation-based approach can also be considered (Liu et al. 2016) which realizes K-mean clustering as a pre-segmentation step and makes the final segmentation by coarse and fine object merging. Study (Demir, Kaynarca, and Oy 2016) realizes fuzzy-based classification of pixels to land/sea classes and extracts the coastline from it as same as (Ferrentino, Nunziata, and Migliaccio 2017) where the authors compute Freeman–Durden decomposition, which is used as a feed to a simple trivial classifier.

We can conclude the literature review by the claim that the model-driven methods work well for a single scenario but might fail when conditions change, i.e., they have low generalization ability. Such changes can be in noise, intensity level, scale, or the presence of artificial objects (ships). That is because the model-driven approaches usually have fixed spatial and intensity dependencies by a bunch of hand-tuned parameters (Demir, Kaynarca, and Oy 2016). They also do not have semantic knowledge, so they cannot distinguish between the sea coastline and the coastline of the inner lakes. That makes the model-driven methods hardly transferable to other landscapes or satellites. Finally, they may have a large computation time (Maraş, Canıberk, and Maraş 2016).

2.2. Data-driven approach

Here, we distinguish between the models based on Deep Neural Networks (DNN) and others. The others involve optimization algorithms, for example, particle swarm optimization (Reis et al. 2018). These optimization methods do not prove their usefulness and, therefore, are used only rarely.

Regarding DNN-based solutions, (Shamsolmoali et al. 2019) propose to use a residual U-Net (Zhang, Liu, and Wang 2018) architecture to segment an input image into sea/land regions. From that, the coastline can be extracted in the way described in model-driven approaches. In (Kesikoglu et al. 2017), authors work with an obsolete fully-connected architecture but propose a pipeline that can directly produce changes in the coastline for
images with various time stamps. Finally, (Dang et al. 2020) proposes a light architecture for coastline classification consisting of five layers, where the most trainable parameters are in the two last fully connected layers. The lightweight architecture allows the usage of relatively high input resolution.

The approaches based on DNNs suffered mainly from big training datasets’ requirements. In this study, we show that with a proper pipeline where pre/postprocessing is carefully designed, it is possible to use even small training datasets, 25 images in our case.

3. Used data, their handling, and augmentations

Our solution is based on deep neural networks. For training such models, it is necessary to prepare the data in a suitable format. Here, we first describe how the data are loaded and transformed, and then how the data are augmented, i.e., distorted to increase the model's generalization ability.

3.1. Provided data

The data were captured as L-band SAR (Synthetic-aperture radar) images by ALOS-2/JAXA satellite with horizontal-horizontal-polarization, i.e., the satellite transmits horizontally polarized waves and receives horizontally polarized waves. The resolution is 3 m × 3 m, an angle of incidence $8° – 70°$, and a swath of 50 km × 50 km. The competition’s data were provided in the form of 32bit tiff images. Unlike the ‘standard’ coastline extraction, the ground truth was prepared using a manual recording of Global Positioning System (GPS) locations during various dates and times instead of the visual observation of the captured images. That means the coastline can visually lie off the border observable from the images because such a border depends on the tide/ebb. It made the problem much more difficult. The training set included 25 high-res images with total $461 \cdot 10^6$ pixels, and the testing set included 30 images with $777 \cdot 10^6$ pixels in total. The usage of external data was forbidden.

3.2. Handling the data

The provided images are in tiff format with a pixel value range in [0, 65,535]. There are two ways to preprocess such images into the [0, 1] range, which is commonly used in deep learning. The first one is based on a linear mapping, i.e., $f' = f/65535$, where $f$ is the original image and $f'$ is the output one. The second one reflects the physical properties of satellites (noise ratio) and it is given as a non-linear log transformation $f' = 10 \log_{10}(f^2 + c)$, where $c$ is a noise reduction coefficient, namely $c = -83$ in our case. There is a general knowledge that the input of a neural network should be the rawest, and the network will select such data/features itself during the training process. Hence, the linear mapping should be selected, but, in our experiments, the non-linear-input-based models yield slightly better accuracy. Therefore, we used an ensemble of models where we applied various types of preprocessing.

The additional manipulation can involve image rescale/crop according to the model resolution defined in Section 4. There are two main reasons to use cropping and avoid
rescaling: firstly, the training dataset is small; thus, there is a need to use as much information as possible. If the images are rescaled to the resolution of a particular model, it will use less than one percent of the original data. Secondly, all images have the same physical resolution, i.e., one pixel in the image always represents the same size in metres, but they have different resolutions, i.e., they capture a different portion of the landscape. Based on the reasons, the rescaling yields poor performance compared to the original data usage. Regarding the cropping, if we have an image with the resolution of $w \times h$ and patch resolution of $a \times b$, we can create $(w-a)(h-b)$ unique overlapping patches. Considering a model resolution of 512 pixels $\times$ 512 pixels and the resolution of particular images, we can obtain more than 365 million unique patches. Because it is impossible to save such an amount on a hard drive, we did not realize the cropping in advance and postponed it to augmentations performed during the training. For an illustration of the augmented images, see Figure 2.

Regarding the labels, they include values of $\{0, 1\}$: the pixel has value 1 if it is a coastline, 0 otherwise. The classes are highly imbalanced because the total area of the coastline is tiny compared to the rest. To postpone the problem of such imbalance, we applied a smoothing filter to make the coastline thicker and obtained labels in the interval $[0,1]$. We also created a second form of label. We manually annotated them as sea/land/no-data. The no-data class is required because parts of some images have missing information. Such labels are more balanced and include more information. Some models were trained on the original labels (coastline) in our pipeline, and some were trained on the modified version to secure the diversity of the models in the ensemble. For details, see Table 1.

### 3.3. Data augmentation

The role of data augmentation is to create new image samples, which help to reduce the model’s overfitting. It is realized during the training and can be generally divided into intensity and spatial augmentation. Intensity augmentation involves additive, multiplicative, or gamma intensity shifts, additive or multiplicative noise, blurring, or Cropout (Hou

---

**Table 1.** Configuration of the four models used in the final ensemble.

| Backbone | Input | Classes                  | Loss      | Last layer |
|----------|-------|--------------------------|-----------|------------|
| EffNetB4 | Log   | Sea/land/no-data/coast   | Dice+Focal| Softmax    |
| EffNetB4 | Log   | Sea/land/no-data         | Dice+Focal| Softmax    |
| EffNetB3 | Log   | Coast                    | BCE       | Sigmoid    |
| EffNetB3 | Linear| Coast                    | BCE       | Sigmoid    |

**Figure 2.** Example of an image obtained from Tellus satellite. Image credit: Tellus.
et al. 2019). The augmentation parameters should always reflect the manner of the data; it is pointless to create entirely artificial samples that cannot be encountered during the inference. The spatial augmentation includes flips, rotation, rescale, creating random patches, elastic transformations, or mosaicing. We took a patch at a random position in the image in our pipeline with a side size in the interval [1024, 1536] and downscaled it to our model’s side size, 512. Then we applied other spatial augmentations. Finally, we realized our own augmentation, multi-sample mosaicing. It splits the data sample into \( n \) rectangular parts and replaces some of them with a same-sized data area from a different image from the training set. The main advantage is that such a composed image includes multiple characteristics, simulating a bigger batch size and postpones overfitting. An example of the input image and its augmented version are shown in Figure 3.

Note, there is a well-established library called Albumentations (Buslaev et al. 2020) realizing the augmentations. Because we involved our own augmentation, multi-sample mosaicing, we created custom functionality and did not use Albumentations.

4. Architecture and training of the derived model

We describe suitable neural network architectures for image segmentation as well as loss functions and optimizers. The selected combination(s) is then the training subject. During the training, it is searched for the optimal setting of the model’s weights to minimize the loss function value. As the training data, we use the original images which are augmented online. The output of the training process is a model that is used for prediction.

The common segmentation architectures are based on an encoder-decoder scheme, i.e., it encodes the input image into a latent space, which is then decoded back to the original space. The spatial dimension is reduced during the encoding while the feature dimension is increased, and vice versa for the decoding phase. The typical representative is U-Net (Ronneberger, Fischer, and Brox 2015) which has been evolved into a residual form or a nested version (Zhou et al. 2018). The advantage of U-Net is the simplicity of coding and low memory requirements compared with the other architectures. The others are, e.g., LinkNet (Chaurasia and Culurciello 2017) or Feature Pyramid Network (Lin et al. 2017a). We have selected U-Net because it allows us to use a bigger batch size than other

![Figure 3. Example of the augmentations. The original image credit: Tellus.](image)
architectures with the same setting, which may postpone overfitting. Compared to Feature Pyramid Network (FPN), it also yields a lower error measured using the competition’s metric, namely, 16.2 vs. 17.2. Note, the absolute winner of the Signate’s competition reported that he used U-Net as well. Because the U-Net encoder’s ability to extract features is limited, replacing it with some of the SOTA networks is beneficial. These networks are called backbones and can be pre-trained on the ImageNet dataset to converge faster. The most powerful (see the comparison in (Bianco et al. 2018)) are ResNet, ResNeXt, RegNet, or NasNet, to name a few. In our pipeline, we have selected EfficientNet (Tan and Le 2019), or EffNet for short. It is a fully convolutional architecture based on compound scaling that easily controls the trade-off between the network capacity and memory requirements. The detailed configuration is given in Table 1.

Because we planned to create an ensemble of different models, we trained several models based on U-Net with EfficientNet backbone. Regarding the loss function, the commons are based on group characteristics such as intersect over union, Tversky loss, or Sorensen-Dice loss (Wang et al. 2020), or pixel characteristics, like Binary cross-entropy (BCE) or Focal loss (Lin et al. 2017b). Each of them has pros and cons. Sorensen-Dice considers the spatial dimension but generally does not lead to the best performance; Focal loss can partially solve the class imbalance problem but may overfit; BCE is generally a good and universal baseline. In our pipeline, we combined Sorensen-Dice with Focal for the two models and use BCE for the other two models, see Table 1. Regarding the optimizer, the first choice is usually Adam, a combination of AdaGrad and RMSProp, which has been several times marked as one of the best optimizers. On the other hand, there are known datasets (such as CIFAR-10 classification) where it yields sub-optimal performance. Therefore, we used Adam optimizer for the two models and AdaDelta for the next two.

The rest of the training settings are as follows. We use the models’ resolution equal to 512 pixels × 512 pixels and as big a batch size as possible, varying from 3 to 12 according to the used graphics card. The models were trained for 100 epochs with reducing the learning rate on a plateau functionality and with saving the best model according to the validation score. The models with sea/land/no-data labels have in the last layer softmax (a smooth approximation of one-hot argmax); the models with the coastline class only have in the last layer sigmoid (a smooth logistic function). It means that the former creates a decision between the classes, and the latter produces the probability of being a coastline. The training of the four models takes approx three days on an RTX2080Ti graphics card.

5. Inference

In the inference part, we take the trained models, process with them the testing images, and produce the predicted coastline. In detail, each model firstly makes a prediction based on the test time augmentation (TTA) principle. From the predictions, we extract the coastlines. Then, the four coastlines (produced by four models) are aggregated into one. Finally, we apply postprocessing to fill the gaps in the coastline.

To produce predictions, each model creates its own ensemble. We use the floating window technique, where we create overlapping patches in multi-scale resolution.
Because the inference is significantly less demanding for memory than training, we are able to process hundreds of patches at once, so the calculation is fast. When the predictions are projected back into the original image, the overlapping parts can be aggregated by summation because the process of extracting the coastline described above does not depend on the absolute values. A Gaussian filter smoothes these produced predictions to decrease the impact of noisy outliers.

The process of extracting the coastline differs for models with softmax in the last layer and for models with sigmoid in the last layer and it is following:

**Softmax models:** The models produce a structure $g$ where each pixel $(x, y) \in g$ is a vector of three values with a probability of being a certain class. The encoding is $0 = \text{sea}$, $1 = \text{no-data}$, $2 = \text{land}$. Firstly, we create $g'(x, y) = \text{argmax}(g(x, y))$, so it holds that $g'(x, y) \in \{0, 1, 2\}$. From it, we create the final binary 'coastline' image $g''$ as $g''(x, y) =$

$$\begin{cases} 1, & \Delta_{\text{max}}(x, y) - \Delta_{\text{min}}(x, y) = 2, \\
0, & \text{otherwise}
\end{cases}$$

where $\Delta_{\text{max}}$ and $\Delta_{\text{min}}$ extract the maximum and minimum values of $g'$ in a $3 \times 3$-pixel neighbourhood of $(x, y)$. For $g''$ holds that $g''(x, y) = 1$ marks the presence of coastline and $g''(x, y) = 0$ no coastline. In other words, we say that there is a coastline if some area contains both 'sea' and 'land' classes without taking into account the 'no-data' class.

**Sigmoid models:** Firstly, we initialize $g''(x, y) = 0$ for all $(x, y)$, and then check if the image has a landscape or portrait orientation. For the landscape, we browse all $x$ coordinates and for each of them we set $g''(x, \text{argmax}(f(x, \cdot))) = 1$ whereby $\cdot$ means all $y$ coordinates. In other words, we are searching for the maximum probability of being a coastline in each column for all rows. The process is the same for the landscape, but we search for each column's maximum in a row. The advantage of the process is the absence of a threshold, so we are able to extract even the most uncertain coastlines. The disadvantage is that we can miss coastline points if the coastline slope is stronger than the diagonal or miss a coastline if there are two coastlines in a row/column. We suppress the disadvantages in the postprocessing later.

**Ensemble of coastlines and postprocessing:** The output image functions $g''$ of the four models were taken and processed in the following way. We browsed the images column/row-wise, as the same as when we made predictions for models with sigmoid. If we browse the rows, we find the coastline coordinates in the columns and create the final prediction as the weighted average of the four predictions. The models’ weights have been set according to a particular model evaluation in the public competition’s leaderboard. Because the realized way of extracting coastlines can create holes when the coastline’s slope is big, we apply postprocessing, which searches for such holes and fills them.

6. Overall results and summarization

We have dealt with the problem of coastline extraction from SAR images. We have recalled the current state and justified the importance of the deep-learning-based approach. We have presented a pipeline split into data preprocessing and augmentation, architecture selection, training of a model, and inference with ensembling and postprocessing. All presented pieces of the mosaic are carefully selected and discussed in the
Table 2. The table shows the ablation study where we measured the impact of various aspects on the official competition score.

| Description                                                                 | Improvement in score |
|------------------------------------------------------------------------------|----------------------|
| Input resolution of the model 512 pixels × 512 pixels instead of 256 pixels × 256 pixels | 0.48                 |
| Used architecture U-Net instead of FPN                                       | 1.04                 |
| Model’s backbone EfficientNetB3 instead of SE-ResNeXt50                      | 1.19                 |
| Models’s Backbone EfficientNetB6 instead of EfficientNetB3                  | −1.99                |
| Applied histogram adjustment as preprocessing                                | −3.12                |
| Softmax activation in the last layer instead of Sigmoid in ‘sea/land/no-data’ model | 1.08                 |
| Adding multi-sample mosaicing to data augmentation                           | 0.52                 |
| Curriculum learning of the models                                           | 0.44                 |
| TTA realized by overlapping patches                                          | 0.62                 |
| TTA improved by multiscale patches                                          | 0.04                 |
| Gaussian smoothing before coastline extraction is realized                   | 0.14                 |
| Ensemble of four models                                                      | 1.41                 |
| Postprocessing, filled gaps in coastline                                     | 0.42                 |

Figure 4. Two illustrations of inference made by our model. Green colour: real coastline measured using GPS. Red colour: inference. It is obvious that standard algorithms based on edges or contours will fail because the real coastline lies in shape without a significantly visible gradient. The original image credit: Tellus.

The measured impact of various aspects of hyperparameters is in Table 2, where the score expresses the mean Euclidean distance between the ground truth and the predicted coastline. It is computed only at the so-called control points defined by the organizers, and their precise location is unknown to competitors. The purpose of control points is to avoid evaluating near rivers where the exact coastline is not defined. The values in the table were obtained as follows: the whole test set was taken, processed using the proposed algorithm, submitted into the competition’s system, and evaluated using the competition’s metric. The results were available after the competition ended; it means the experiment was blind during the competition.

The described solution forms a general pipeline that can be applied to many segmentation tasks and whose superiority has been confirmed by reaching the second place in the worldwide Signate competition. For the illustration of the visual performance, see Figure 4. The proposed pipeline is modular, which means it can be easily updated by new data, new augmentation types, or more powerful backbones.

Acknowledgments

We want to thank Dr. Kunihiro Watanabe (National Institute for Land and Infrastructure Management, Japan) for his valuable pieces of advice and comments.
Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Petr Hurtik https://orcid.org/0000-0003-4349-9705
Marek Vajgl https://orcid.org/0000-0002-4751-7117

References

Acar, U., B. Bayram, F. B. Sanli, S. Abdikan, F. Sunar, and H. I. Cetin. 2012. “An Algorithm for Coastline Detection Using SAR Images.” ISPAR 39: 457–460.
Bianco, S., R. Cadene, L. Celona, and P. Napoletano. 2018. “Benchmark Analysis of Representative Deep Neural Network Architectures.” IEEE Access 6: 64270–64277. doi:10.1109/ACCESS.2018.2877890.
Buslaev, A., V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. 2020. “Albumentations: Fast and Flexible Image Augmentations.” Information 11 (2): 125. doi:10.3390/info11020125.
Chaurasia, A., and E. Culurciello. 2017. “Linknet: Exploiting Encoder Representations for Efficient Semantic Segmentation.” In 2017 IEEE Visual Communications and Image Processing (VCIP), 1–4. Saint Petersburg, Florida: IEEE.
Dang, K. B., Q. T. Bui, T. P. N. Pham, et al. 2020. “A Convolutional Neural Network for Coastal Classification Based on ALOS and NOAA Satellite Data.” IEEE Access 8: 11824–11839. doi:10.1109/ACCESS.2020.2965231.
Demir, N., M. Kaynarca, and S. Oy. 2016. “Extraction of Coastlines with Fuzzy Approach Using SENTINEL-1 SAR Image.” The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 41: 747. doi:10.5194/isprs-archives-XLI-B7-747-2016.
Ferrentino, E., F. Nunziata, and M. Migliaccio. 2017. “Full-polarimetric SAR Measurements for Coastline Extraction and Coastal Area Classification.” International Journal of Remote Sensing 38 (23): 7405–7421. doi:10.1080/01431161.2017.1376128.
Ferrentino, E., A. Buono, F. Nunziata, A. Marino, and M. Migliaccio. 2020. “On the Use of Multi-polarization Satellite SAR Data for Coastline Extraction in Harsh Coastal Environments: The Case of Solway Firth.” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.
Hou, W., W. Wang, R.-Z. Liu, and L. Tong 2019. “Cropout: A General Mechanism for Reducing Overfitting on Convolutional Neural Networks.” In 2019 International Joint Conference on Neural Networks (IJCNN), 1–8. Budapest, Hungary: IEEE.
Kesikoglu, M. H., S. Y. Cicekli, T. Kaynak, and C. Ozkan. 2017. “The Determination of Coastline Changes Using Artificial Neural Networks in Yamula Dam Lake, Turkey.” In 2017 8th International Conference on Information Technology (ICIT), 737–740. Amman, Jordan: IEEE.
Lin, T.-Y., P. Dollár, R. Girshick, H. Kaiming, B. Hariharan, and S. Belongie. 2017a. “Feature Pyramid Networks for Object Detection.” In Proceedings of the IEEE conference on computer vision and pattern recognition, 2117–2125, Honolulu, Hawaii.
Lin, T.-Y., P. Goyal, R. Girshick, H. Kaiming, and D. Piotr. 2017b. “Focal Loss for Dense Object Detection.” In Proceedings of the IEEE international conference on computer vision, 2980–2988, Honolulu, Hawaii.
Liu, Z., L. Fei, L. Ning, R. Wang, and H. Zhang. 2016. “A Novel Region-merging Approach for Coastline Extraction from sentinel-1A IW Mode SAR Imagery.” IEEE Geoscience and Remote Sensing Letters 13 (3): 324–328.
Maras, E. E., M. Canberk, and H. H. Maras. 2016. “Automatic Coastline Detection Using Image Enhancement and Segmentation Algorithms.”
Mirsane, H., Y. Maghsoudi, R. Emadi, and M. Mostafavi. 2018. “Automatic Coastline Extraction Using Radar and Optical Satellite Imagery and Wavelet-IHS Fusion Method.” International Journal of Coastal and Offshore Engineering 2 (2): 11–20. doi:10.29252/ijcoe.2.2.11.

Modava, M., and G. Akbarizadeh. 2017. “Coastline Extraction from SAR Images Using Spatial Fuzzy Clustering and the Active Contour Method.” International Journal of Remote Sensing 38 (2): 355–370. doi:10.1080/01431161.2016.1266104.

Modava, M., G. Akbarizadeh, and M. Soroosh. 2019. “Hierarchical Coastline Detection in SAR Images Based on Spectral-textural Features and Global–local Information.” IET Radar, Sonar & Navigation 13 (12): 2183–2195. doi:10.1049/iet-rsn.2019.0063.

Mori, N., S. Nakajo, S. Iwamura, and Y. Shibutani. 2018. “Projection of Decrease in Japanese Beaches Due to Climate Change Using a Geographic Database.” Coastal Engineering Journal 60 (2): 239–246. doi:10.1080/21664250.2018.1488513.

Reis, H. C., B. Bayram, S. Bozkurt, A. Ince, N. Demir, and D. Z. Seker. 2018. “An Extended Approach of Particle Swarm Optimization for Shoreline Extraction from RASAT Imagery.” Journal of the Indian Society of Remote Sensing 46 (8): 1223–1232. doi:10.1007/s12524-018-0804-0.

Ronneberger, O., P. Fischer, and T. Brox. 2015. “U-net: Convolutional Networks for Biomedical Image Segmentation.” In International Conference on Medical image computing and computer-assisted intervention, 234–241. Munich, Germany: Springer.

Shamsolmoali, P., M. Zareapoor, R. Wang, H. Zhou, and J. Yang. 2019. “A Novel Deep Structure U-net for Sea-land Segmentation in Remote Sensing Images.” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12 (9): 3219–3232. doi:10.1109/JSTARS.2019.2925841.

Tan, M., and Q. V. Le. 2019. “Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks.” arXiv preprint arXiv:1905.11946.

Wang, L., C. Wang, Z. Sun, and S. Chen. 2020. “An Improved Dice Loss for Pneumothorax Segmentation by Mining the Information of Negative Areas.” IEEE Access 8: 167939–167949. doi:10.1109/ACCESS.2020.3020475.

Wiehle, S., and S. Lehner. 2015. “Automated Waterline Detection in the Wadden Sea Using High-resolution TerraSAR-X Images.” Journal of Sensors 2015. https://www.hindawi.com/journals/js/2015/450857/.

Zhang, Z., Q. Liu, and Y. Wang. 2018. “Road Extraction by Deep Residual U-net.” IEEE Geoscience and Remote Sensing Letters 15 (5): 749–753. doi:10.1109/LGRS.2018.2802944.

Zhou, Z., M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. 2018. “Unet++: A Nested U-net Architecture for Medical Image Segmentation.” In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, 3–11. Granada, Spain: Springer.